CV Week 1

1. OpenCV 用 BGR order purity brightness
2. RGB HSV (Hue Saturation Value) (human perception) Grayscale
   color type → 100 鲜艳 后到
   绿色在转化时权重最大
3. Top left 读取 矩阵. 先行后列

Week 2 image filters

1. box filter. ① mean filter $\frac{1}{9}$ 全1 $h[m,n]=\sum_{k,l} g(k,l)f[m+k,n+l]$
   可以 smooth, remove sharp features
   ② sharpening $2\cdot$ − $\frac{1}{9}$ 全1 锐化中心

linear filters 性质 (一个 filter 是线性, iff $T[\alpha f_1+\beta f_2]=\alpha T[f_1]+\beta T[f_2]$
① 线性: filter(f_1+f_2) = filter(f_1) + filter(f_2) α,β 是标量常数. f_1, f_2 是任意图像
② 平移无关性: shift invariance filter(shift(f)) = shift(filter(f)) 输入移动像素, 输出也移动像素
任何线性, 平移无关的计算可被卷积表示.
③ 齐次性: 缩放输入等比缩放输出
④ 交换律: $f*h=h*f$ (仅限卷积定义, 互相关不对称)
⑤ 结合律 $(f*h_1)*h_2=f*(h_1*h_2)$
⑥ 分配律: $f*(h_1+h_2)=f*h_1+f*h_2$
⑦ Identity $a*e=a$

2. Gaussian filter $G_\sigma=\frac{1}{2\pi\sigma^2}e^{-\frac{(x^2+y^2)}{2\sigma^2}}$
   $=(\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}})$
   $(\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{y^2}{2\sigma^2}})$
   ① 移除高频, smooth
   ② 和自己卷还是高斯. $\sigma\to\sqrt{2}\sigma$
   ③ separable kernel 可以被分解为两个 1-d 的高斯变量好. 假设输出为 $M\times N$. 一个像素要做 mn 次乘法. $MNmn\to MN(m+n)$

filter 多大. half-width $\geq 3\sigma$ 左是右. 右是左
贴边咋办? ① clip filter(black) 全填黑 ② wrap around ③ copy edge 把边 pixel extend ④ 镜像

3. Sobel filter ⇒ 识别横. 纵边

4. laplacian filter unit-Gaussian ≈ Laplacian of Gaussian 可以提取高频细节

5. 为什么 filtering 重要

① 窗间相关 ② locality 重要信息在小 region ③ Stationarity 平稳性 ④ Hierachical structure 层次结构 ⑤ similarity ⑥ scale invariance

6. Denoising

① Gaussian：大方差可 smooth + 压噪，但伤害画质

② Median filtering 抑制 salt-and-pepper (impulse noise)

select the median intensity : Robustness to outliers

还有 weighted median，clipped mean 等

7. Template matching

a. 直接把 eye patch 当作 filter cons：易被高亮响应，而非对应 patch

b. 把 eye patch 去均值

c. Sum of Square Distance (SSD) $h[m,n]=\sum_{k,l}(f[k,l]-im[m+k,n+l])^2$

$=\sum_{k,l}f[k,l]^2-2\sum_{k,l}im[m+k,n+l]f[k,l]+\sum_{k,l}im[m+k,n+l]^2$

常 linear 平方求和再过一个全1 filter

可进一步拆成更快的 linear filter

cons：对 intensity shift 敏感

d. normalized cross correlation

$$h[m,n]=\frac{\sum_{k,l}(f[k,l]-\bar{f})(im[m+k,n+l]-\overline{im_{m,n}})}{\sqrt{\sum_{k,l}(f[k,l]-\bar{f})^2\sum_{k,l}(im[m+k,n+l]-\overline{im_{m,n}})^2}}$$

Cosine similarity $h[m,n]=\frac{\sum(f-\bar{f})(im-\bar{im})}{\sqrt{\sum(f-\bar{f})^2(im-\bar{im})^2}}$

Comparison： zero-mean filter 最快，但效果一般

SSD：第二快，但对 overall intensity 敏感

Normalized cross-correlation：最慢，和局部均值与对比无关

8. Low-pass filter

图 $\xrightarrow{\text{Gaussian filter}}$ $\xrightarrow{\text{Sample}}$ low-res

Gaussian Pyramind.

Laplacian Pyramid.

构建多分辨率层级

单尺度全局大搜索 → 局部多尺度搜索

为什么先 gaussian？直接 downsample 引发 Aliasing，高频细节 →低频伪影

但 Gaussian Pyramind 的 downsample 是有损压缩

引入 Laplacian Pyramind 存预测偏差 $L_i=G_i-Expand(G_{i+1})$

↑ 上采样 + 高斯平滑

可逆性

I hybrid = Low Freq (A) + High Freq (B)

对A做 Gaussian filter 对B做 Laplacian gaussian filter

或 B-Gaussian(B)

CV Week 3 Edges & Contours 边缘检测与轮廓分析

1. 尝试一：High frequency ≈ Edges?

① Disconnected：高频响应在弱边界处断裂，无法生成闭合环

② Noise Amplification：随机像素波动被高频算子放大，产生大量伪边

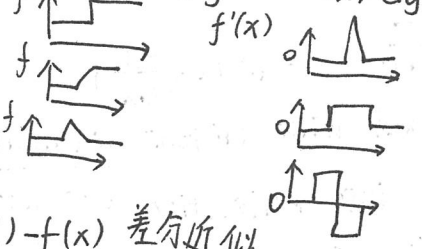
③ Thick Edges：边缘带宽 5~10 pixels，~~丧失~~ not precise 但对noise更sensitive

edge是brightness突然变化

微分放大 noise

2. Gradient-based：$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) = (G_x, G_y)$ 一阶导在边缘处达到峰值，二阶导在 edge 处穿过 0

三种边 ① Step edge ② Ramp edge ③ Roof edge



Gradient Magnitude $|\nabla I| = \sqrt{G_x^2 + G_y^2}$

Gradient Direction $\theta = \arctan\left(\frac{G_y}{G_x}\right)$ arctan2(Gy, Gx)

gradient ⊥ edge

离散：$f'(x) \approx f(x+1) - f(x)$ 差分近似

只用2 pixel，噪音大 没有 smoothing 通用

2.1 Sobel Operator

Direction info：X和y

2.1.1 Sobel X Kernel - 检测纵边 vertical

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$f(x+1) - f(x-1)$

水平变化 ⇔ 纵边

2.1.2 Sobel Y Kernel - 检测横边

$$Sobel_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$f(y+1) - f(y-1)$

这样 3×3 邻域 + Built-in smoothing (center weight = 2) + 对噪声不敏感

2.1.3 Design Insight

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \otimes [-1\ 0\ 1]$$

Smooth vertically ↑ differentiate horizontally

note 用 cv2.CV_64F CV_8U 不能负 0~255

2.1.4 $|\nabla I|$ 大的 $\sqrt{G_x^2 + G_y^2}$ all edge directions combined

2.1.5 Limitations：① Thick edges，非 pixel-level precision

② Still Noise sensitive：比 simple diff 好

③ No thresholding：不能说哪个是边，没有自动评判标准

我们需要用 entire pipeline 把 gradient ⇒ clean, precise 边

DoG 与 LoG 快

$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2\nabla^2 G$

选smooth再拉 ↓高斯差分

LoG ≈ DoG approximation

2.2 其他算子

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

[1, -2, 1] 与 $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$ 相加

2.2.1 Prewitt Operator 有方向

$$Prewitt_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad Prewitt_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

no center emphasis

2.2.2 Laplacian：2nd：找 $\nabla^2 I = 0$ Zero-crossing of 二次导

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ or } \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

比 Sobel 对 noise 还敏感 !!!

用于 detects all edges equally (isotropic)

LoG (with 高斯) smoothing

对noise没那么敏感 isotrophic 各向同性

LoG：Gaussian + Laplacian

Smooth first → clean edges

3. Canny Edge Detection 40年历久弥新. 先定义optimal. 再设计

3.1 3 Goals

① low error rates : 真边✓ 假 X
② precise localization : 精确定位
③ Single Response : 一条边 → 一个 pixel

Gaussian Smoothing + dual threshold
NMS
NMS Non-Maximum Suppression

3.2 5 步

① Gaussian Smoothing : 去噪. 保结构 $\frac{d}{dx}(f*n) = \frac{df}{dx} * n$ 白噪声均值为0, 但方差被放大

② Gradrent : Sobel $M = \sqrt{G_x^2 + G_y^2}$ $\theta = \arctan(\frac{G_y}{G_x})$

③ NMS : keep only ridge (pixel with largest gradient) 1 pixel
沿梯度方向比较(和边垂直)
将连续 gradient 方向 quantized into 4 directions : 0°, 45°, 90°, 135° 量化到最近角度
对每个 pixel, 沿该方向比较前后两个相邻pixel的 gradient magnitude
若 current ≥ both → keep ; else : suppress 舍弃
→ one pixel

④ Dual Threshold 单个 threshold { T大 : noise 抑制但不连续 / T小 : 有noise但连续 → 双 threshold

M ≥ T high : 强边 (一定是真边 "种子"
T low ≤ M ≤ T high : 弱边 (noise / 边的延伸)
M < T low : 舍弃

强度不是唯一, connectivity decides

T high : T low 2:1 to 3:1
① 够宽才行. (T_L, T_H) too narrow, 真 ion tour weak segments mostly fall below T_L and being discarded outright. 后面无法重连
⇒ 强弱边
② 太宽不行. ⇒ T_L 很低. all noise pixels 进入 dense noise regions 在 step 5 仍可能成边

⑤ Edge Tracking by Hysteresis
①从强边出发, 沿 8 邻域搜弱边 可用队列 queue 维护
②连到的弱边 promote 成强边 ↙能在此上搜8邻域
③孤立弱边为噪声

cv2.Canny(img, threshold1 = 50, threshold2 = 150) built-in 用固定 5x5 G Blur

3.3 去① 很多 noise → 升 false edges ②去③ thick bands ③单threshold 去边/noise
去⑤ weaker edges 可能是真边被去了

3.4 选什么. ① general : Canny
② 需要 gradient 方向 : Sobel
③ Isotropic : LoG
④ Real time : Sobel (single-step)

调参 平滑尺度 σ, T_L, T_H

4. Edge → Contour

4.1 区别：边是 pixel labels，2D image，不连续，不能数object，不能看形状
轮是 connected curves，点序列，连续，可以数，能看形状

4.2 pipeline Original —Canny→ Edge Image —findContour→ Contour List → Object Boundary

4.3 findContours
① 扫图找第一个edge起点
② 沿edge逐像素追踪，直到返回原点，存为一个contours
③ 继续扫，提取所有
可以只提最外层 | 所有(list) | 所有(Tree)　RETR_EXTERNAL　RETR_LIST　RETR_TREE　无层次，在乎嵌套
压缩：appox_none不压．approx_simple 只保端点　4个点

4.4 性质．S.C．矩．Centroid　空间矩 $M_{ij}=\sum_{x,y} x^i y^j I(x,y)$　$S=M_{00}$；质心 $\bar{x},\bar{y}=(\frac{M_{10}}{M_{00}},\frac{M_{01}}{M_{00}})$
bounding box
$M_{pq}=\sum_{(x,y)\in contour} x^p y^q$ p,q次方
M_{00} zeroth moment = S 像素总数
M_{10} 一阶矩 x方向．所有像素x坐标之和
M_{01} y方向　y坐标之和
轮廓contour内所有点

5. 分析工具

5.1 Circularity：多圆 $=\frac{4\pi \times Area}{Perimeter^2}$　圆是1，越小越不圆
周长

aspect ratio：$\frac{w}{h}$
x,y,w,h = cv2.boundingRect(c)　extent = $\frac{Contour\ Area}{Bounding\ box\ Area}$　闭合长方形　0.785 circle $\frac{\pi r^2}{4r^2}$　square = 1.0 方形

面积可用于过滤，把面积低于threshold的丢弃

完整流程　转灰度 ⇒ 高斯模糊 ⇒ canny → 形态学闭运算 → 查找过滤contour

~~convex hull~~ solidity = $\frac{A}{convex\ Hull\ Area}$　凸形状无凹陷=1　星形月牙形<1

morphological closing
= dilation then erosion
expands region, closing small gaps in the edges
shrinks them back to approximately original size
1-3 pixels get bridged　broken curves ⇒ closed loops

CV Week 4 Match

1. 希望找 $(x_i, y_i) \leftrightarrow (x_i', y_i')$

1.1 能用 edge to match？ 平行于边的变化无法察觉（无法区分边上的点）

1.2 好 feature 两个 hard-req

① repeatability 不同视角、光照仍能检测到 or 找不到

② distinctiveness：不易与别的附近区域 mix or wrong match

⇓

Corner 优于 edge，在二维方向有明显变化，完整位置约束

good Features To Track

2. Haris Corner：两边都变为 corner

2.1 Window shift energy：设窗口中心在 (x,y)，shift by(u,v)，Intensity change

$$E(u,v) = \sum_{(i,j)\in W} [I(i+u, j+v) - I(i,j)]^2$$

对所有方向 (u,v) 都小：flat

某些大，某些小：edge。所有大 corner

对小 (u,v) 有 $I(i+u, j+v) \approx I(i,j) + I_x u + I_y v$ $I_x = \frac{\partial I}{\partial x}$ $I_y = \frac{\partial I}{\partial y}$

$$E(u,v) \approx \sum_{(i,j)\in W} (I_x u + I_y v)^2 = [u\ v] \underbrace{\sum_W \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}}_{M} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \begin{bmatrix} \sum_W I_x^2 & \sum_W I_x I_y \\ \sum_W I_x I_y & \sum_W I_y^2 \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

$\sum_W I_x^2$：x 梯能　$\sum_W I_x I_y$：相关性　$\sum_W I_y^2$：y 梯能

M 可以拆成 $M = Q \Lambda Q^T$ $\Lambda = diag(\lambda_1, \lambda_2)$

$$E(u,v) = [u\ v] M \begin{bmatrix} u \\ v \end{bmatrix} = [u\ v] Q \Lambda Q^T \begin{bmatrix} u \\ v \end{bmatrix}$$

$$= \triangleq [u'\ v'] \Lambda \begin{bmatrix} u' \\ v' \end{bmatrix}$$

$$= \lambda_1 (u')^2 + \lambda_2 (v')^2$$

也就是 λ_1 大，x 上梯度大，有纵边

λ_2 大，y 上梯度大，有横边

都大，有 corner

$M \in R^{2\times 2}$ $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$

2.2 怎么求 λ_1, λ_2 $\det(M) = \lambda_1 \lambda_2$

$\det(M - \lambda I) = 0$

$\lambda^2 - (a+c)\lambda + (ac - b^2) = 0$

$\lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$

$\lambda_1 \lambda_2 = ac - b^2 = \det(M)$

$\lambda_1 + \lambda_2 = a + c = \text{trace}(M)$

2.3 定义 Haris Response $R = \det(M) - k(\text{trace}(M))^2 = AB - C^2 - k(A+B)^2$

$k = 0.04 \sim 0.06$

- Corner (λ_1, λ_2 大)：$R \gg 0$
- flat：$R \approx 0$
- Edge ($\lambda_2 = 0$)：$R \approx -k\lambda_1^2 < 0$

3、Haris 后处理：Haris 得到的是 Haris Response Map

3.1 NMS：局部邻域内只保留最大的 ⇒ exactly one pixel → [ ]

3.2 Threshold：$R > R_{max} \cdot quality\ level$

← 整图最大　← 保留点质量比例

3.3 minDistance：按 R 从大到小排序，每选一个，判断其 minDistance 范围内候选点。

⇒ 让点均匀分布

Image $\xrightarrow{Sobel}$ I_x, I_y $\xrightarrow{window\ sum}$ M $\xrightarrow{det\ Trace}$ Rmap $\xrightarrow{NMS}$ local maximum $\xrightarrow{Threshold}$ Candidates $\xrightarrow{minDist}$ final

Haris 三大问题：无尺度、无方向、无descriptor

缺口	具体	SIFT解决
scale	远近变化窗口看不同	DoG scale space. G
orientation	旋转patch对不齐	主方向估计. 描述前旋转对齐
descriptor	只有位置. 无法验证是否同	128-D gradient histogram descriptor

4. SIFT

4.1 scale invariance : DoG

$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$

尺度σ下平滑图 ↑ 标准差为σ高斯核

建 Gaussian Pyramid (blur at ↑σ). 相邻层相减

$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma)$

↑ DoG response

k是相邻尺度间倍数. kσ下一层尺度

$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2\nabla^2 G$ approximates scale-normalized Laplacian of Gaussian

SIFT在DoG体中找三维极值 $D(x,y,\sigma) > D(x',y',\sigma')$ $\forall (x',y',\sigma') \in N_{26}(x,y,\sigma)$

N_{26} 当前尺度同层8邻居 或 $D(x,y,\sigma) < D(x',y',\sigma')$ $\forall (x',y',\sigma') \in N_{26}(x,y,\sigma)$

上下尺度各9邻居. 总计26邻居

一个包含 S 个可用尺度层次 要 S+2 张 DoG ⇒ S+3 张高斯模糊图像

极大极小 → 候选keypoint，并记录其 σ

4.2 Lowe's keypoint sieve：过滤低质量点 ~~以上一步keypoint为中心，按照σ确定~~

上一步 90% junk

① Low Contrast. $|\hat{D}(\hat{x})| < 0.03$ (8-bit image) ⇒ Discard ↙ contrast threshold 对比度.

② Edge Response (Hessian filter)

令 Hessian 为 $\begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$ = 阶导 ⇒ 特征值 α, β 且 $\alpha \geq \beta$ 令 $r = \frac{\alpha}{\beta}$

若 $\frac{tr(H)^2}{det(H)} < \frac{(r+1)^2}{r^2}$ 作为保留条件. ratio太高，则r大 ⇒ 为边，舍弃

和 Harris 很像，不过这里用曲率

4.3 Rotation Invariance 以上一步keypoint为中心 按照σ确定圆形邻域大小

在一个大圆区域 ①为36个方向 36-bin梯度直方图. 取峰值方向为dominant orientation；旋转坐标系对齐

$\theta(x,y) = \text{atan2}(L(x,y+1) - L(x,y-1),\ L(x+1,y) - L(x-1,y))$ $\theta^* = \text{argmax}\, h(b)$

SIFT会返回比extremas多的keypoint $\tilde{x} = R(-\theta^*)(x - x_0)$ 一阶导 若次要峰值达到主峰值 ≥80% ▷ keypoint is duplicate 相同位置与σ，但 secondary orientation

4.4 Lighting Invariance (raw pixel敏感，pixel difference也sensitive，gradient direction robust)

每个keypoint已有 position, scale, orientation ⇒ 需 compact numerical vector encoding local appearance

SIFT descriptor标准构造 以先前keypoint的pos, σ, orientation为基准，将patch规范化到同一坐标系下，算方向直方图，生成descriptor

① 尺度归一，方向对齐后的key point周围取16×16邻域 ②划分为4×4 16个cell ③每个cell统计8-bin梯度方向直方图 ④拼接成 16×8 = 128 R^{128} ⑤ L2归一化 $f \leftarrow \frac{f}{\|f\|_2}$

⑥ clip $f_i \leftarrow \min(f_i, 0.2)$ 再归一化 恢复单位长度 防止单个主导梯度 dominates descriptor

减少梯度饱和突变的影响

②

5. Feature Mapping：从descriptor相似到几何一致

5.1 Brute-force matching

直接找最近 l_2-norm

$D_A = \{d_i^A\}_{i=1}^N$ $D_B = \{d_j^B\}_{j=1}^M$ 对每个 d_i^A 寻找最近邻 $j = \arg\min_j \|d_i^A - d_j^B\|_2$

$O(MN)$ 会带来 false matches. 存在 scalability wall

5.2 kNN + Lowe Ratio Test 对2

找 k=2 最近邻 比较 ratio of d $\frac{d_1}{d_2} < 0.75 \rightarrow$ keep repetitive textures 效果最好 纹理

好 $\begin{cases} \text{if } d_1 \ll d_2 \text{ 好} \\ d_1 \approx d_2 \rightarrow \text{discard} \\ \text{若都很大} \rightarrow \text{discard} \end{cases}$

descriptor 构多 distinctive.

解决局部 descriptor 惟一性问题

但不检查多匹配是否满足同一个几何变换

5.3 RANSAC (Random Sample Consensus)

正确匹配需满足一个全局几何变换

Homography $x' \sim Hx$ $x = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ $x' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ 8个自由度

① 随机采4对非共线匹配点 minimum for a homography

② 求候选 H

③ 对所有匹配求 rejection error

④ error < threshold 为 inliers

⑤ 重复 N 次

⑥ 留 inlier 最多 H

⑦ 用所有 inliers 重新拟合 H

$s\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = H\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$

$N = \frac{\log(1-p)}{\log(1-w^s)}$ 采样点数 $s=4$ 内点率

什么时候 hold $\begin{cases} \text{① scene is planar} \\ \text{② rotates around its optical centre} \end{cases}$

若有显著 depth change 和相机平移

视差会破坏模型

为什么用直方图不是 raw pixsels

① illumination robustness. Uniform brightness change doesn't affect ∇ and its dir

normalization removes absolute magnitude

② Spatial Tolerance: 4×4 cell grids gives tolerance to small shift

③ Compactness: 128 floats ≈ 512 bytes per point. 省空间

cumulative drift via bundle adjustment 多张一起优化 wide-angle disortion via cylindrical projection brightness differences via multi-band blending 多频段

CV Week 5

1. Pre-requisite

1.1 scalar $|a+b| \le |a|+|b|$

1.2 vector $\|x\| = \sqrt{\sum x_i^2}$

1.2 Matrix $C = AB$ $C_{ik} = \sum_j A_{ij} B_{jk}$

$\|a+b\| \le \|a\| + \|b\|$

$\|A\|_{Frob} = \left[\sum_{ij} A_{ij}^2\right]^{\frac{1}{2}}$

$\|A+B\|_{Frob} \le \|A\|_{Frob} + \|B\|_{Frob}$

$\|aA\|_{Frob} = |a| \cdot \|A\|_{Frob}$

小图 ⇒ no room for multi-scale feature hierachies, texture statistics, or spatial pyramid

红特征 { combinational explosion：组合数太多，类内变化大
domain expertise bottleneck：design
brittleness：只能用在指定conditions

Why DL
① 数据
② 算力
③ 架构

2. kNN

2.1 1-NN：训练时不真正学参数，记住全部 data, labels，test时找最近的，复制 label

$i = \arg\min_i d(x, x_i)$ $\hat{y} = y_i^*$ d是 distance function 常用 L_1

2.2 从 1-NN → kNN 1-NN对noise sensitive 若最近点刚好异常，wrong.

$N_K(x)$ = the set of K nearest training samples to x

$\hat{y} = \arg\max_c \sum_{i \in N_K(x)} 1(y_i = c)$ 多数投票

2.3 超参选择，k和 $d(\cdot,\cdot)$ 不能直接选，应 train set 训模型
val set 选超参
test set 最后评一次

但DL用得少
Cross-Validation
训练分多个 fold，
一个 val，剩余 train
然后平均验证结果

2.4 基本不用 { ① pixel distance 无语义信息
② 慢 O(ND) ← 训练样本 ← dimension

3. Linear Classifier

4. SVM

4.1 Loss 要求正确类别比错误类别至少多一个 margin，set to 1

$S = f(x_i; W) \in R^C$ C是类别数

$L_i = \sum_{j \neq y_i} \begin{cases} 0, & \text{if } s_{y_i} > s_j + 1 \\ s_j - s_{y_i} + 1, & \text{otherwise} \end{cases}$

s_j 是模型对 j 的打分

s_{y_i} 是模型对正确类别打分

对每个错误类别，希望 $s_{y_i} > s_j + 1$。

margin

$L = \frac{1}{N} \sum L_i$

4.2 optimal W 并不唯一 $f(x; W) = Wx$ s.t. $L=0$ 可令 $W = nW$ n↑ margin↓ overfit

$n(W_j^T x_i - W_{y_i}^T x_i + \frac{1}{n})$ 等效 margin $1 \to \frac{1}{n}$

引入 regularization $L = \frac{1}{N} \sum_{i=1}^{N} L_i(x_i, W, y_i) + \lambda R(W)$

L_1 $R(W) = \|W\|_1 = \sum_k |W_k|$

L_2 $R(W) = \sum_k W_k^2$

当 小W对应大 margin，泛化更好；限W能让输出平滑

4.3 SVM score → prob：softmax $\frac{e^k}{\sum e^i}$ Cross entropy $L = -\frac{1}{N} \sum_i p_i \log(q_i) = -\frac{1}{N} \sum_i \log \frac{e^{s_{y_i}}}{\sum_j e^{s_j}}$

当 $s_{y_i} \to \infty$ 整个 $\to 1$ $L \to 0$ 当 $s_{y_i} \to 0$ 那 $L \to \infty$ $L_i = -s_{y_i} + \log \sum_j e^{s_j}$

若初始化，所有 score 等 $L_i = -\log \frac{1}{C} = \log C$

4.4 Comparison SVM Loss → score Softmax cross entropy 输出 prob

5. Optimization

5.1 Random Search 低效　上游 $\frac{\partial L}{\partial output}$　局部 $\frac{\partial output}{\partial input}$

5.2 GD　往下游传 upstream x local

5.2.1 SVM Loss的∇ $L_i = \sum_{j \neq y_i} \max(0, w_j^T x_i - w_{y_i}^T x_i + 1)$

对某错误类别 j　$m_j = w_j^T x_i - w_{y_i}^T x_i + 1$

$\frac{\partial L_i}{\partial w_j} = x_i$　对正确类边权重 $\frac{\partial L_i}{\partial w_{y_i}} = -x_i$

若有多类别超出margin, $\frac{\partial L_i}{\partial w_{y_i}}$ 会累加

$\frac{\partial f(x)}{\partial x_r} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

$z = (\langle x, w \rangle - y)^2$　令 $a = \langle x, w \rangle$　$b = a - y$　$z = b^2$

$\frac{\partial z}{\partial b} = 2b$　$\frac{\partial b}{\partial a} = 1$　$\frac{\partial z}{\partial a} = 2b$　$\frac{\partial a}{\partial w} = x$　$\frac{\partial z}{\partial w} = 2bx = 2(a-y)x$

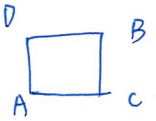
$= 2(\langle x, w \rangle - y)x$

决策边界　$s = w_1 x_1 + w_2 x_2 + b = 0$　i和j的决策边界在分数相等处

$s_i = s_j \Rightarrow (w_i - w_j)x + (b_i - b_j) = 0$

每个区域是凸的，任何凸区域内连线仍在凸区域内

XOR　若有 $w_1 x_1 + w_2 x_2 + b = 0$ 分开



$b \leq 0$ 代入(0,0) 类别为0

$w_1 + w_2 + b \leq 0$　$w_2 + b > 0$　$w_1 + b > 0$

$p_i = \frac{e^{s_i/T}}{\sum_{j=1}^{C} e^{s_j/T}}$

$T < 1$ 低温　指数陡，放大得分差异，分布更尖锐

$T \to 0$　max 独热

$T > 1$ 高温　压缩得分间差异

$|1 - \eta\lambda| < 1 \Rightarrow \eta < \frac{2}{\lambda}$　η超过这个，每一步会增加|θ|-远离最小值

全量 GD　$O(N \times D \times C)$　C: class

N个　输入维度

单样本∇是真实梯度有噪无偏估计

$E_i[\nabla L_i(\theta)] = \frac{1}{N} \sum_{i=1}^{N} \nabla L_i(\theta) = \nabla L(\theta)$

$CE = -\log(P_y) = -\log\left(\frac{e^{z_j}}{\sum_{i=1}^{C} e^{z_i}}\right) = -z_j + \log\left(\sum_{j=1}^{C} e^{z_j}\right) = -z_j + m + \log\left(\sum_{j=1}^{C} e^{z_j - m}\right)$

B是比单样本SGD的梯度估计方差的缩小倍数

令 $m = \max_j z_j$　保证指数输入小于0

CV Week 6 MLP → CNN

Linear Classifier 只能学习线性边界，无法处理 XOR 等非线性

MLP 引入 activation function → 非线性 提升表达能力，但对图像全连接参数量巨大

CNN 利用 image 局部性，平移共享，空间结构，大幅减少参数并学局部视觉特征

$\sigma'(x) \le 0.25$ 每次乘 0.25

1. 激活函数

且总在[0,1]，随深度加大，问题放大，▽消失

Leaky ReLU $\max(0.1x, x)$ ↓ Maxout $\max(w_1^T x + b_1, w_2^T x + b_2)$ $ELU = \begin{cases} x, & x \ge 0 \\ \alpha(e^x - 1), & x < 0 \end{cases}$

问题 ① sigmoid { ① 过大过小 saturated

② 非 0-centered 若神经元输出总为正，局部梯度也为正，所有权重符号受 upstream gradient 的统一控制，要么全增，要么全减，无法针对更新个

$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial z} \cdot a_j$ ↙总正 $z = \sum_j w_j a_j + b$

不适合隐藏层

$\sigma'(x) = \sigma(x)(1-\sigma(x)) \ge 0$

② tanh (-1,1) zero-centered. 比 sigmoid 适合当中间层，但仍会 saturate

$1-\tanh^2(x)$ 最大为1，比 sigmoid 的 0.25 好，但仍会收缩

③ ReLU pros: ① 正区间不 saturate ② easy-compute ③ 实践比 sigmoid/tanh 收敛快 ④ Sparsity 约一半神经元关闭，有计算优势

cons: ① output 非 zero-centered ② 可能出现 dead neuron 输出总负，无法更新

{ ① 大 lr，把 b 推到很大负值，$w^T x + b < 0$ ② 不巧的初始化一开始就死

leaky ReLU ~~$\max(x, ax)$~~ ↑ 可以监控 减小 lr

$\begin{cases} x, & x > 0 \\ ax, & x \le 0 \end{cases}$ 使用小的正偏置 b=0.01

PReLU α可学 a可学 ← 0.01 死了还学

④ ELU $f(x) = \begin{cases} x, & x > 0 \\ \alpha(\exp(x) - 1), & x < 0 \end{cases}$

pros ① 正区间不 saturate ② 负区间接近 zero mean ③ 负区间 saturate 增加对 noise 的 robustness.

cons: exp 难算

对于非常负，趋近于 $-\alpha$，产生接近0 mean的输出，解决 ReLU 和 sigmoid 零中心问题

⑤ Maxout $f(x) = \max(w_1^T x + b_1, w_2^T x + b_2)$ 可以泛化 ReLU 和 Leaky ReLU

pros { ① liner regime ② non-saturating ③ does not die

problem: 一个神经元要训两倍参数

建议 ① hidden 用 ReLU，但小心 lr

② 可以用 Leaky ReLU，Maxout ELU 获得边际提升

③ hidden 不用 sigmoid/tanh，只用来限制 output range

2. CNN

2.1 paras: 一个 filter $H \times W \times C + 1$ ↙ bias

C 要求输入 channel 一致

一层可以 stack k 个 filter 出来 k 层

2.2 输出公式 无 padding

$\text{Output size} = \frac{N-F}{S} + 1$ ↑ filter size ↑ 输入尺寸 ↑ Stride

N=7, F=3 stride 1 就是 $\frac{7-3}{1} + 1 = 5$

输出非整数，不能合法卷积。

Receptive field = $L \times (F-1) + 1$

不pad的话，size 快速减小

有 padding $\frac{N+2P-F}{S} + 1$ 相当于原来尺寸变大

$Y(i,j) = \sum_{c=0}^{C-1} \sum_{m=0}^{F-1} \sum_{n=0}^{F-1} W(c,m,n) X(c, i+m, j+n) + b$ translation equivariance

2.3 Receptive field 当层则和filter size一致

2.4 Pooling layer : ①:减少 representation size
②让特征更 manageable
③对每个 activation map 独立操作
不影响channel数

深比宽好 ① less param
② more nonlinearities
③更易优化

Input → [Conv → ReLU → Pool] × N → flatten → FC → softmax

CV Week 7 Optimization of CNNs

1. Batch Normalization

1.1 Input: $x \in R^{N \times D}$

对每个 feature $\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$ $\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{i,j}-\mu_j)^2$

Xavier $Var(W) = \frac{1}{n_{in}}$

He $Var(W) = \frac{2}{n_{in}}$

$\hat{x}_{i,j} = \frac{x_{i,j}-\mu_j}{\sqrt{\sigma_j^2+\varepsilon}}$ ε 小常数

LN: 对 HxW 归一 GN: 通道组+空间

但 zero mean 和 unit variance 可能过于严格

可引入 learnable scale and shift $\gamma, \beta \in R^D$ $y_{i,j} = \gamma_j \hat{x}_{i,j} + \beta_j$ can perfectly undo

希望恢复 identity mapping 可以学习 $\gamma_j = \sigma_j$, $\beta_j = \mu_j$ $y_{i,j} = \sigma_j \frac{x_{i,j}-\mu_j}{\sigma_j} + \mu_j = x_{i,j}$

1.2 训测时差距 训练时 μ_j, σ_j^2 来自当前 mini-batch never hurts

μ_j^{test} = running mean $(\sigma_j^2)^{test}$ = running variance

pros:
① stablising training
② act as a regulariser
③ allows higher lr
④ Reduce sensitivities to init

1.3 插在 FC 和卷积层后 在 activation 前

$\mu_{running} \leftarrow (1-m)\mu_{running} + m\mu_B$ $\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial \hat{x}_i}\cdot$

$\sigma^2_{running} \leftarrow (1-m)\sigma^2_{running} + m\sigma_B^2$

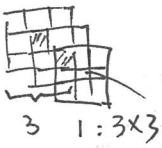
cons:
① 小 batch works poorly
② 不能用于 sequence (leakage)
③ different beha in train & infer

2. 架构

2.1 AlexNet 用较大卷积核，网络深度有限，FC层参数量大，训练依赖工程技巧

2.2 VGG: 用小卷积堆深度.

3个 3x3 stride 1 等价一个 7x7 的 conv



$3\times2+1=7$

$R = 1+3\times(3-1)=7$

每多一层 kxk，receptive field 增加 k-1

$R_k = R_{k-1} + (K_k - 1)$

$R_0 = 1$ 输入层的一个像素

1个 $7\times7 = 49C_{in}C_{out} = 49C^2$ 若等 3个 3x3 $3\times9C^2 = 27C^2$ 减少参数又还能加非线性层数

2.3 Plain deep network: 54层比20层 traing error 和 test error 都更差

不是 overfit 因为54层比20层 train 还差 是 performance degradation

2.4 ResNet $F(x) = H(x) - x$ $y = F(x) + x$

1. Stack Residual Blocks
2. 每个 residual block 两 3x3 conv
3. 周期性 double filters 用 stride 2 downsample spatial size
4. 开头有 stem conv 如 7x7 stride 2
5. 末尾无大 FC layers 只用 global average pooling + FC 1000

HxWxC

$\Downarrow$

$z_c = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} x_{i,j,c} \Rightarrow z \in R^C$ 减少参数，降过拟合

理论支持 variable size input

保留 channel-level semantic representation

3. Optimizer

1. SGD问题
① Poor Conditioning: 一方向快，一方向慢。Steep上来回震荡，flat上进展缓慢
$\kappa = \frac{\lambda_{max}(H)}{\lambda_{min}(H)}$ H: Hessian matrix
最flat上曲率方向 最steep上的曲率方向
κ越大，优化越难
② Local Minimum
二阶导
③ Saddle point: 无更新方向，易stuck
高维更常见
saddle point: 一些方向曲率为正
但一阶导为0 一些方向曲率负
④ Noisy Gradient 随机采样有噪声

2. SGD + Momentum
历史梯度方向累积
$v_{t+1} = \rho v_t - \alpha \nabla f(x_t)$
$x_{t+1} = x_t + v_{t+1}$
在flat direction 累积速度，加快前进 steep上来回震荡，互相抵消，减少jitter
通过惯性帮助越过小local minimal/saddle point

3. Adagrad
$r_t = r_{t-1} + g_t \odot g_t$ $x_{t+1} = x_t - \alpha \frac{g_t}{\sqrt{r_t} + \varepsilon}$
↑ 维
历史梯度平方和
逐元素开方
若某维梯度一直大，对应η↓
一直小，对应η↑
steep → damp flat → accelerated
问题 r_t 单增，step size → 0

4. RMSProp: 修正Adagrad lr衰减问题，指数滑动平均
$r_t = \beta r_{t-1} + (1-\beta) g_t \odot g_t$ $x_{t+1} = x_t - \alpha \frac{g_t}{\sqrt{r_t} + \varepsilon}$

5. Adam 一阶矩 $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$
二阶矩 $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$ $(g_t \odot g_t)$
预热 从10^{-7}开始
前1000-5000步线性增加
对Adam很有用
$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$ 和 $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$ $x_{t+1} = x_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$
初期估计偏小

4. Regularization generalization error = test error - traing error
4.1 Early Stopping 监控 val loss/acc 当 val performance ↓ stop
4.2 Model Ensembles 训多个 $p_k(y|x)$ $p(y|x) = \frac{1}{k}\sum_{k=1}^{k} p_k(y|x)$ $y = \arg\max_y p(y|x)$
性能↑ 开销大
4.3 l-2 weight decay $+\lambda \|W\|_2^2$ 惩罚大的 梯度 $+2\lambda W$ l-1 $+\lambda |W|_1$ ⇒ sparse weight
Elastic net $+\lambda_1 \|W\|_1 + \lambda_2 \|W\|_2^2$ L_1 稀疏 L_2 平滑

4.4 Dropout
4.4.1 训练时每个neuron采样一个Bernoulli mask p是留存prob
dropout后激活 ↓ 原激活
$\tilde{h}_i = m_i h_i$

4.4.2 ensemble 解释 每个binary对应一个network . 2^n种 n为unit数

4.4.3 test. 所有 neuron 都启用

$h_i^{test} = p h_i$ 缩放, 使期望一致

现代实现常用 inverted dropout 训练除以 p . 测试不用缩放

希望 training 引入 randomness. 测试 通过 average out randomness

5. Data Augmentation $(x,y) \mapsto (T(x), y)$ T是 Augmentation

① horizontal flip

② Random crops and scales

③ Color Jitter

6. Transfer Learning

小数据集policy：冻 CNN backbone . 重新初始化分类，重新训

也可以训一些layer

CV Week 8 RNN + Transformers

1. RNN $h_t = f_w(h_{t-1}, x_t)$ $y_t = g_{w_o}(h_t)$ $h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$

同一组参数 W 在所有 timestamp 共享 $y_t = W_{hy}h_t + b_y$

1.1 RNN计算图与任务类型

1.1.1 many → many $L = \sum_{t=1}^{T} L_t(y_t, \hat{y}_t)$ frame-level classification

language modeling

1.1.2 many → one $h_T = f_w(h_{T-1}, x_t)$ $y = g(h_T)$ sentiment classification

video classification

1.1.3 one → many image captioning $h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + W_{ih}v)$

把上一步输出塞进来 v是image feature

1.2 charcter-level LM $o_t = W_{hy}h_t + b_y$ $P(x_{t+1}=k \mid x_{\le t}) = \frac{\exp(o_{t,k})}{\sum_j \exp(o_{t,j})}$

输入 [Start] → 下一词分布 → sample → 喂回去 → 生成 END / 最大长度

2. Seq2Seq : Encoder - Decoder 架构

$[x_{1:T}] \longrightarrow u_t = RNN(x_t, u_{t-1})$ h0是初始hidden state

Decoder $y_t = g_v(y_{t-1}, h_{t-1}, c)$ c是context vector

或 $h_0 = f_w(z)$ z可以是encoder最终状态 / 所有encoder states集合 与输入几乎隔开 可以是 h_0

2.1 Bottleneck: c必须承载整个输入序列所有信息。 encoder出来 h_0 h_0和c要生成后续所有.

但 c 是 fixed-length vector ⇒ 信息瓶颈

3. RNN + Attention

3.1 固定 context vector c → 每个timestamp不同 context

$y_t = g_v(y_{t-1}, h_t, c) \rightarrow y_t = g_v(y_{t-1}, h_t, c_t)$

动态选择输入相关区域

3.2 Captioning 中的attention

① CNN提取 spatial features $z \in R^{H \times W \times D}$ $z_{i,j} \in R^D$

② 对 decoder 第 t 步，算 alignment score → MLP $e_{t,i,j} = f_{att}(h_{t-1}, z_{i,j})$ 当前步与 z_{ij} 契合度

③ 归一化 $a_{t,i,j} = \frac{\exp(e_{t,i,j})}{\sum_{m,n} \exp(e_{t,m,n})}$ 来满足 $0 < a_{t,i,j} < 1$ $\sum_{i}^{H}\sum_{j}^{W} a_{t,i,j} = 1$

④ context vector $c_t = \sum_{i,j} a_{t,i,j} z_{i,j}$

3.3 soft att vs hard attn soft: $\sum_{i,j} a_{t,i,j} z_{i,j}$ 连续可微，可end2end BP

hard att: 离散选某一区域 $i_t \sim Categorical(a_t)$ $c_t = z_{i_t}$ 采样不可微，要RL / policy gradient

4. NLP 中的 Attention：Translation

Input $x = x_{0:T}$　　输出 $y = y_{0:T'}$　长度可以不同

Encoder: $z_i = RNN(x_i, u_{i-1})$　← RNN hidden state

Decoder

$e_{t,i} = f_{att}(h_{t-1}, z_i)$

归一化 $a_{t,i} = \frac{\exp(e_{t,i})}{\sum_{i'=1}^{T}\exp(e_{t,i'})}$

context vector $c_t = \sum_{i=1}^{T} a_{t,i} z_i$

decoder 输出 $y_t = g_v(y_{t-1}, h_{t-1}, c_t)$

不同 target word 对应不同 source word

Attention 自动学 alignment

且无需人工标注对齐关系

5. General Attention Layer

$X = \{x_{1:N}\}$ $x_i \in R^D$　query $h \in R^D$　alignment $e_i = f_{att}(h, x_i)$

attention weights $a_i = \frac{\exp(e_i)}{\sum_{j=1}^{N}\exp(e_j)}$　$c = \sum_{i=1}^{N} a_i x_i$　操作本身 permutation invariant

把 MLP Alignment 换成点积 $e_i = h^T x_i \Rightarrow$ scaled $e_i = \frac{h^T x_i}{\sqrt{D}}$　D 是维度

若 h 和 x_i 各维度近似独立且方差为 1，那么点积方差大约与 D 成正比．维度越大，logits 越大，softmax 越易饱和．梯度变小．除以 $\sqrt{D}$ 控制数值

5.1 多个 query

$q_{1:M}$　$e_{i,j} = \frac{q_j^T x_i}{\sqrt{D}}$　$a_{i,j} = \frac{\exp(e_{i,j})}{\sum \exp(e_{i',j})}$　$y_j = \sum_{i=1}^{N} a_{i,j} x_i$

j 是 query 索引，i 是 input index

6. Transformer

XW_Q　XW_V

$X \in R^{N \times D}$　$Q \in R^{M \times D_K}$　$V \in R^{N \times D_K}$

$D_K = D$

$W_Q \in M \times N$

$Att = softmax(\frac{QK^T}{\sqrt{D_K}})V$

7. PE

$p_j = pos(j)$　$pos(\ ): R \to R^d$

- learned positional embedding 需 model 自学　不能自然泛化到超过 Tmax 位置间关系
  $P \in R^{Tmax \times d}$　$p_t = P[t]$
- Sinusoidal　高维↓低频维捕捉长距离变化

$PE(pos, 2i) = \sin(\frac{pos}{10000^{2i/d_{model}}})$

$PE(pos, 2i+1) = \cos(\frac{pos}{10000^{2i/d_{model}}})$

$MultiHead(Q,K,V) = Concat(head_1, \cdots, head_H)W_o$

CV Lec 9 Detection / Segmentation

1. Semantic Segmentation

Input $I \in R^{H \times W \times C}$ $\Rightarrow Y = \{1, \dots, C\}^{H \times W}$ 总共class数

⇒ 只关心 what class each pixel belongs to. 忽略同类间实例差异

1.1 直觉方法 sliding window：each pixel 周围取 patch，CNN 判中心 pixel.

$P_{i,j} = I[i-r:i+r, j-r:j+r]$ ⇒ 预测 $\hat{y}_{i,j} = \underset{c}{\text{argmax}}\, f_\theta(P_{i,j})_c$ ↑在类别中

用了 context，但未解决 shared computation

1.2 全卷积，对一张图一次提 feature map，对每个空间分类

$F = g_\theta(I) \in R^{D \times H \times W}$ feature map

$S = h_\phi(F) \in R^{C \times H \times W}$ 每pixel出prob

$Y_{i,j} = \underset{c}{\text{argmax}}\, S_{c,i,j}$ 对pixel取最大类

但一直维持 H×W 计算大，CNN的 pooling 会降尺寸 ⇒ 不满足 pixel-wise prediction.

FCN 折中：encoder ⟶ downsample ⟶ (decoder) upsample

2. Upsampling

2.1 非学习，① Nearest Neighbour 直接复制值 不可学

② Bed of Nails Upsampling 插 0 配合后续卷积

③

2.2 Max Unpooling，和 Max pooling 成对出现 下采样记 max，还记 max 所在位置

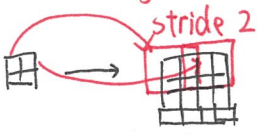
即 pooling switches. 上采样把低 reso 值放回原来最大值处，其余填 0

pros：部分恢复 pooling 前空间结构 cons：只恢复最大激活位置，仍 sparse

2.3 Transposed Convolution：deconvolution：对每个输入元素都"描"出一个乘以该输入值的 filter pattern，多个 pattern 在输出空间处叠加

对 input [a, b] filter 为 [x, y, z] stride = 1

⇒ $a[x,y,z] + b[\_, x, y, z]$ ⇒ $[ax, ay+bx, az+by, bz]$



重叠部分相加 一个值被 3×3 filter 成了 9 个值

stride gives the ratio between movement in output and input

$O_{transposed} = (I-1) \times S - 2P + K$ (S: 步长, P: padding, K: 核大小)

3. U-Net

3.1 FCN 问题：encoder 深层 feature 语义强，但空间细节弱；浅层 feature size 大，但语义弱

3.2 solu：encoder-decoder + skip connection encode 中高 reso 直传 decoder 拼接 copy and crop

$D_l' = \text{Concat}(D_l, E_l)$ $D_{l-1} = \text{Conv}(D_l')$

4. Object Detection:

输出不是像素类别，而是对像集合 $\{(c_i, b_i)\}_{i=1}^{N}$（类别 / bounding box）

$b_i = (x_i, y_i, w_i, h_i)$（box左上或中心 / 长宽）

单目标检测可看成 classification + localization

网络共享一个 feacture vector $z \in R^{4096}$

分类头 $p = \text{softmax}(W_c z + b_c)$　　$\hat{b} = W_b z + b_b$ 实位头.

loss $L = L_{cls} + \lambda L_{Loc} = -\log p_y + \lambda \|\hat{b} - b^*\|_2^2$

⇒ 多目标 detection 困难 { ① 图 object 数不同，输出长度不固定
② CNN分类要输出固定长多维向量，无法直接输出可变长度集合

⇒ 低效法（候选）对图像中大量 crop 分别分类为 object/background

问题：所有位置，尺度，长宽比都枚举，计算量巨大

⇓ Region Proposals: Selective Search 选举 candidate crops

4.1 RCNN

Input image → (selective search) RoIs → (Warp to 224×224) CNN → Feature vector → SVM分类 / bbox regress

每个RoI分别过CNN　　每个RoI一个

预测的是 correction (dx, dy, dw, dh)

因为每个RoI有初始(x,y,w,h)　一个窗口

败笔：每图约2000个RoI，独立过CNN，慢 未共享卷积计算

4.2 Faster RCNN：先对 image 做 backbone 得到 feature map，再把 proposal 映射到 feature map 上做 RoI Pool / Crop + Resize

Input Image → (Backbone CNN，只有一次) Conv Feature Map → (RoI proposals projected onto feature map) RoI pooling / crop-resize

提取固定大小特征 ⇒ 过全连接 → cls / box

→ per-region head → classification + bbox regression

仍依赖外部 proposal 方法，如 selective search，在 CPU 上，慢

4.3 Faster R-CNN：提出 Region Proposal Network (RPN)，RPN与 detector 共享 backbone feature map，直接在 feature map 每个位置预测 anchor 是否包含 object 以及对应 box correction

生成 RoI 几乎无延迟

RPN：一个点有 k different anchor boxes of different size/scale.（预定义，且预测对 anchor corrections）

让 conv 输出 $k \times H \times W$ (Objectiveness)，Score $k \times H \times W$ by "objectness score"（且做边界微调）

用 top ~300 as our proposals → (全连接) 输出 box transforms $4k \times H \times W$

RoI-Pooling 量化取整，舍去小数部分，取子窗口像素最大值，空间精度低，存在"未对准"

4.4 Single-Stage Detector: YOLO | SSD | RetinaNet

NMS { ① 按置信度对所有检测排序 ② 保留置信度最大的 ③ 移除与其 IoU 超过 threshold 的结果

4.4.1 Two-stage: 先生成 proposals 再进行分类回归

Single stage: 直接在 dense grid 上预测 boxes 和 classes

4.4.2 YOLO: 把图划成 S×S 网格，每个网格预测 B 个 bounding boxes 和类别 prob

⇒ S×S×(5B+C)　B: (dx, dy, dh, dw, confidence). C是类别(预测)，含enviroment

confidence = P(object)·IoU($\hat{b}$, b*)　很像 RPN，但具体到类别

⇒ P(c|object) P(object) IoU($\hat{b}$, b*)　每个cell还pred p(c|object)

pros: 快，整张图只forward一次

cons: dense prediction 产生大量候选框，需NMS

且早期 YOLO 对小物体，密集物体不如 two-stage

4.5 DETR: Object Detection with Transformer

不用 anchors, IoU threshold, NMS, 把 object detection 视为 set predictions

输出固定 N 个 object queries 的预测 $\{(\hat{c}_i, \hat{b}_i)\}_{i=1}^{N}$　无 object 的 query 被视为 no-object

图 → CNN, PE → encoder → decoder → FFN → prediction heads

queries (一组可学习向量)

Set Prediction + Hungarian Matching: 给定真实 objects: $\{(c_j, b_j)\}_{j=1}^{M}$

预测出 $\{(\hat{p}_i, \hat{b}_i)\}_{i=1}^{N}$　$N \geq M$.

找最优匹配 $\hat{\sigma} = \arg\min_{\sigma \in G_N} \sum_{j=1}^{M} L_{match}((c_j, b_j), (\hat{p}_{\sigma(j)}, \hat{b}_{\sigma(j)}))$

L_{match} 含分类与 box 代价

找唯一最优一对一匹配，避免重复预测

① 消除很多手工工程组件 (anchors, IoU thresholds, NMS)

② 不用 duplicate prediction 带来的后处理 (NMS等，传统方法会重复预测)

③ 拥有 global reasoning (self-attention 带来，任意两 pixel 直接交互)

④ end2end，可以一起训，不用像传统方法分阶段训 (RPN → 分类头)

5. Instance Segmentation　$\{(c_i, b_i, m_i)\}_{i=1}^{N}$　m_i 是第 i 个 instance 的 binary mask

5.1 Mask-RCNN　在 $m_i \in \{0,1\}^{H \times W}$

在 faster R-CNN 上，除了 RoI feature 上再加一个 mask head → 大 C×28×28 masks　在 object detection 上多了 instance identity

用 RoI Align 解决了空间对齐问题 (用 bilinear interpolation 保留更准确空间对齐，因为 mask prediction 对 pixel level alignment 很敏感)

训练只对 GT 类别对应的 mask 算 binary cross entropy

6. Visualization: 理解 model 到底学了什么

6.1 filter visualization

6.2 Saliency Map: 看类别分数对输入像素的梯度. 给定类别 c 的 unnormalized score $S_c(I)$

$G = \frac{\partial S_c}{\partial I} \in \mathbb{R}^{3 \times H \times W}$　$M_{i,j} = \max_{k \in \{R,G,B\}} \left|\frac{\partial S_c}{\partial I_{k,i,j}}\right|$　取 RGB max 可视化

若某像素轻微变化强烈影响 score.

⇒ 其对类别重要

6.3 Class Activation Mapping (CAM)

依赖特殊结构：卷积层后接 GAP，再接 linear classifier

$f \in R^{H\times W\times k} \xrightarrow{GAP} F \in R^k \xrightarrow{FC} S \in R^C$

$F_k = \frac{1}{HW}\sum_{h,w} f_{h,w,k}$ $\quad S_c = \sum_k W_{k,c} F_k = \frac{1}{HW}\sum_k W_{k,c}\sum_{h,w} f_{h,w,k}$

$= \frac{1}{HW}\sum_{h,w}\sum_k W_{k,c} f_{h,w,k}$

类别 c 的 CAM map 是 $M_c(i,j) = \sum_{k=1}^{k} W_{k,c} f_{h,w,k}$

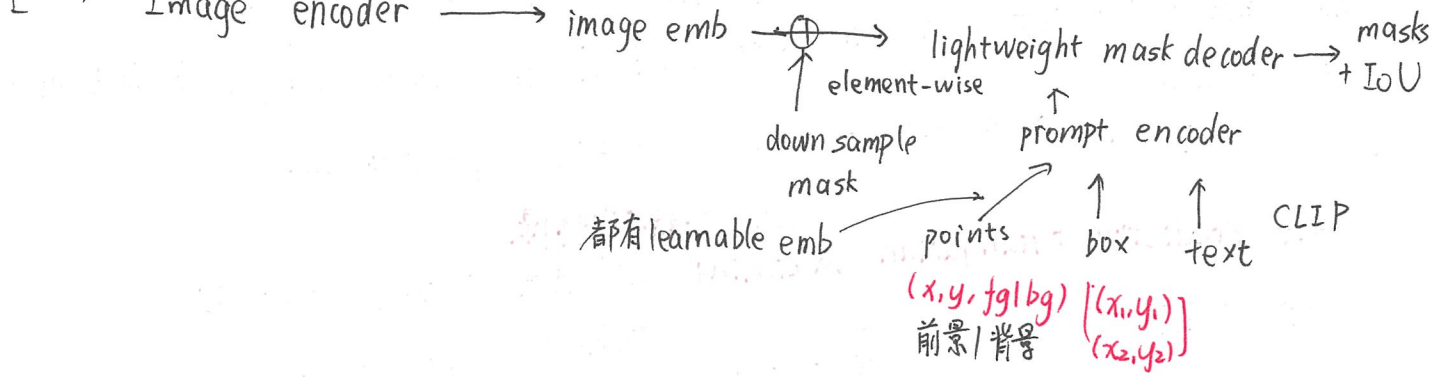
Grad-CAM
① Any layer, with Activation $A \in R^{H\times W\times k}$
② 算 S_c 对 A 梯度 $\frac{\partial S_c}{\partial A} \in R^{H\times W\times k}$
③ GAP $\alpha_k = \frac{1}{HW}\sum_{h,w}\frac{\partial S_c}{\partial A_{h,w,k}}$
④ $M^c_{h,w} = ReLU(\sum_k \alpha_k A_{h,w,k})$ 只保留对 c 有正贡献

梯度：增强 feature channel → score↑?

7. SAM Image + Prompt → Valid Mask

point box text mask

promptable seg
可能 prompt ambiguous，但仍要返回 valid mask



CV Lec 10

1. Autoencoder → VAE 压缩表征 → 生成模型

1.1 Autoencoder：无监督特征学习

input x $\xrightarrow{\text{encoder}}$ latant feature z $\xrightarrow{\text{Decoder}}$ reconstrction $\hat{x}$

$L_{AE} = \|x - \hat{x}\|^2$ → 比如说保留训好的 encoder 来初始化 supervised model 做 pred

降维、特征学习，辅助预训练，非好生成模型：

AE 学到的是离散，不规则，可能有空洞的 latent space

若只生成一样图片那还好，但很难生成类似有差异的图

$x \mapsto z$ 没有保证 z 分布连续光滑可采样

1.2 VAE 假设 AE 输出确定性 latent vector $z = f_\phi(x)$，但 VAE 输出 latent distribution $q_\phi(z|x)$

$q_\phi(z|x) = N(\mu(x), \Sigma(x))$ 或 $q_\phi(z|x) = N(\mu(x), \text{diag}(\sigma^2(x)))$

生成时 ~~$z \sim p_z(x)$~~ $z \sim p(z)$ $\hat{x} = \text{Decoder}_\theta(z)$ 选 $p(z) = N(0, I)$

$L = \|x - \hat{x}\|^2 + D_{KL}(q(z|x) \| p(z))$

reconstruction error：decoder 能从 sampled latent z 恢复原图

D_{KL}：要求 decoder 输出 latent distribution 接近标准正态

会惩罚 encoder 把不同样本压到孤立 cluster

怎么算？ $z = \mu + \sigma \odot \varepsilon$

$$D_{KL}(q(z|x)\|p(z)) = \frac{1}{2}\sum_{i=1}^{n}(\Sigma_i^2 + \mu_i^2 - 1 - \log\Sigma_i^2)$$

$$= \frac{1}{2}\sum_{j=1}^{d}(\sigma_j^2 + \mu_j^2 - 1 - \log\sigma_j^2)$$

$x \xrightarrow{\text{encoder network}} \begin{cases} \mu_{z|x} \\ \Sigma_{z|x} \end{cases} \xrightarrow{\text{Sample}} z \xrightarrow[p_\theta(x|z)]{\text{Decode}} \hat{x}$

$z|x \sim N(\mu_{z|x}, \Sigma_{z|x})$

- VAE 易模糊
  - 分布 multi-modal：$p(z) = N(0, I)$ 不同模式之间被平滑连接，采样可能落在类间
  - pixel-wise reconstruction loss 本身有 averaging effect。若多输出合理，l_2 loss 倾向于预测平均图像 $\hat{x} = \arg\min_{\hat{x}} E[\|x - \hat{x}\|^2]$ 趋向条件均值

VAE: limited diversity

2. Generative Adversarial Networks (GANs)

不要求重建某样本，而是学从随机噪声到真实数据分布的 mapping

$z \sim p_z(z)$ $\hat{x} = G(z)$ 不知对应 ⇒ 不用 reconstruction loss

目标 $G(z) \sim p_{data}(x)$

D：Discrimminator

$$\min_{\theta_g}\max_{\theta_d} V(D,G) = E_{x\sim p_{data}(x)}[\log D_{\theta_d}(x)] + E_{z\sim p_z(z)}[\log(1 - D_{\theta_d}(G_{\theta_g}(z)))]$$

训练：① fix G, train D 对 D，希望 $D(x)\to 1$ $D(G(z))\to 0$

② fix D, train G to fool D 对 G，希望 $D(G(z))\to 1$

GAN 问题
- ① Bad convergence 训练不稳定，生成图像更锐利
- ② Mode collapse
- ③ Discriminator too strong/too weak
- ④ No explicit likelihood

~~DL L8 LLM~~

2.1 Cycle GAN & Style GAN

Cycle GAN：解决 unpaired image-to-image translation

$G: X \to Y$

$F: Y \to X$

D：判断是否属于目标域

Cycle consistency loss

$$L_{cyc}(G,F) = E_{x\sim p_{data}(x)}[\|F(G(x))-x\|_1] + E_{y\sim p_{data}(y)}[\|G(F(y))-y\|_1]$$

$y \to F \to \hat{x} \to G \to \hat{y}$，$D_X$ 判 $\hat{x}$ 与 x；$G(F(y))-y$

$x \to G \to \hat{y} \to F \to \hat{x}$，$D_Y$ 判 $\hat{y}$ 与 y；$F(G(x))-x$

对抗 loss

$L_{GAN}(G, D, X, Y)$

对 G，minimize $E_{x\sim p_{data}(x)}[(D(G(x))-1)^2]$ 希望 D 判 G(x) 为真

对 D，minimize $E_{y\sim p_{data}(y)}[(D(y)-1)^2] + E_{x\sim p_{data}(x)}[D(G(x))^2]$ 希望 D 判 y 为真，判 G(x) 为假

Style GAN：把 latent code 映射到 style space　AdaIN 模块

3. Diffusion

3.1 forward

$$x_t = \alpha_t x_{t-1} + \beta_t \varepsilon \qquad \varepsilon \sim N(0,I)$$

$t \uparrow$　$x_t \to N(0,I)$

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

3.2 Reverse

$x_T \sim N(0,I)$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t,t)\right) + \beta_t \varepsilon$$

3.3 训练　采样

$x_0 \sim q(x_0)$

$t \sim Uniform(\{1,\dots,T\})$

$\varepsilon \sim N(0,I)$

$\nabla_\theta = \|\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t)\|^2$

直到收敛

$x_T \sim N(0,I)$

for $t=T,\dots,1$ do

$z \sim N(0,I)$ if $t>1$, else $z=0$

$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t,t)\right) + \beta_t z$

end for

return x_0

3.4 Stable diffusion

直接在像素空间生成计算大　$H \times W \times 3$

Image x —Autoencoder Encoder→ latent z —Diffusion U-Net（在 latent space）→ $\hat{z}$ —AE decoder→ Generated image

conditioning (Guiding the output): text prompts, semantic maps, reference images

过一个 domain-specific encoder

U-Net 还用了 cross-attention 去看 text prompt

3.4 ControlNet　很难精确描述空间结构

3.4.1 text prompt limitation：① pose ②透视 ③边缘 ④深度.

可输入 Openpose. Canny Edge　Depth Map　Scemantic Map

锁住 pretrained diffusion model

加一个可训练控制分支

$y_c = y + ControlNet(x)$　zero conv 是 1×1 conv，权重 bias 初始化为 0

初始训练时 $ControlNet(x)=0$　$y_c = y$

复制一份 encoder，在旁边加一份可训练副本

用零卷积逐步注入原始模型

希望输入近的逐渐获得权重