Xavier Initialization: **std=1/sqrt(Din)**. Din=F^2C for Conv layer
这样使 Var(y=wx)=Var(xi)，解决了 covariance shift。如果用一个固定的 var，X 会随着层数加深被挤到中间或者两边。
ReLU 去掉了一半，要乘以 sqrt2 修正。**Std=sqrt（2/Din）**
梯度消失：使用了饱和激活函数 Sigmoid/tanh，导数在两端接近 0，会导致浅层网络无法有效学习。解决：ReLU，Res, BN.
梯度爆炸：初始权重过大，每层放大因子>1，连乘后爆炸，会导致训练损失变成 NaN 或者权重溢出，模型不收敛。解决：梯度裁剪，BN, He/Xavier initialization

SGD 的问题：陷入鞍点(高维常见)，如果 loss 有一个方向陡峭一个方向平坦，GD very slow progress along shallow dimension, jitter along steep direction，梯度来自 minibatch 所以很 noisy。
解决 optimizer 1. Momentum：vt+1=rho*vt+gradx, x-=αvt+1
2. Adam: 给了一套 default 参数，lr 为常数。
Iteration：one batch，a GD step。Epoch：完整走过一遍训练集，每个 Epoch 过后测一下 validation loss。（对较大模型可重新定义 Epoch）
调整 lr 的逻辑是按照 iteration 而非 epoch,常见阶梯式下降、Cosine：α=α0(1-t/T) T 为 epoch 总数。Lr 的线性 warm-up，防止一开始 lr 过大 loss 爆炸
SGD+momentum 可能会比 Adam 更好，但是需要更多调参.

欠拟合：1. 模型 capacity 受限：中间层加宽加深 2. 优化不佳：BN
BN 在 FC 和 Conv 层之后，ReLU 前使用。BN 每层的均值和标准差都可学习，这样 BN 在一方面限制了模型，又给予模型一定的自由度。输入是 NxD 维的，N 为 batchsize，D 为 channel 数，都是在 N 上进行 norm，gamma 和 beta 均为 D 维（保证平移等变性）。BN 分为 train mode 和 eval mode，因为 eval 时逐张测试，不能计算均值和方差，采用 running mean 和 var。running mean=rho×running mean+(1−rho)×μB。是 BN 在训练过程中滑动平均计算的全局均值。Y[ij]=gamma[j]X_hat[ij]+beta[j]
BN 让网络更容易训练，可以允许更大的 lr，对初始化更鲁棒，而且 BN 还相当于一种正则化，可以同时缓解欠拟合和过拟合，BN 逐渐取代了 Dropout。
最后一层不能加 BN，因为最后输出要用于预测，强制归一化会破坏输出的实际含义。BN smoothens the loss landscape, but may not reduce the interval covariate shift. 层与层之间γ和β不同，分布不同，不能说 BN 控制 covariate shift 不变。He init+BN 实现了从起始到 update 后对每一层的分布控制。
问题：BN 在 training 和 testing 时表现有差异，会成为 bug 来源。BN 在 batchsize 较小时问题显著，因为 batch 之间差异较大（选 group norm）。
变长序列任务（如 NLP）不适用 BN，替代：LayerNorm, InstanceNorm, GroupN. 它们都不关于 batch 维算 mean，则 batch 中有多少 data 和它无关，在训练和测试中表现相同。LayerNorm 在 NLP 中常用，但 CV 很少用，因为对 channel 做了平均，但不同 channel 代表不同 feature，不能平均。
过拟合：早停，数据增强，正则化，Dropout、BN。平衡模型表达力和 datavari。数据增强做完一定要人为 check，防止核心信息丢失。正则化项系数目的是防止正则化喧宾夺主，一开始先不加确保 loss 能下降，等出现过拟合再逐渐加。Dropout 也分 train-mode 和 eval-mode，使训练对特征更鲁棒，防止不同特征 co-adapt，只在大 FC 层使用。DA 和 BN 总是很好用。

非参数模型：KNN：对语义无关的变化过于敏感，且 test 太慢（要遍历数据集）当 KNN 所用的度量是深度神经网络学来的，3d 视觉，image retrieval 有应用
Softmax+CELoss：softmax 相当于二分类中的 Sigmoid，是加了 exp 再归一化(所有维和为 1)。Exp 使输出趋向于 one-hot，但是相比 argmax 更 soft 而方便 BP，前者没有 gradient 无法优化。Logits->unnormalized prob.->prob.
对 prob 和标签（one-hot）求 KL-Divergence，其中第一项即为二者交叉熵，第二项为 label 的熵直接舍去。L=-∑PlogQ，Q 为输出的 prob，P 为标签向量。随机初始化时 L≈logN 类别，但是最差无上界，最小为 0.

增长 receptive field：层层抽取结构，在 flatten 后变为全图（FC 每个 pixel 均为全图），最后层之前变为语义的高维线性空间，可以用 softmax 进行分类。3 个 3x3conv 相当于一个 7x7conv，但是层数更深，更多非线性层，而且参数更少 3*(3^2C^2) vs 7^2C^2; AlexNet 一开始使用 11x11conv，太浅了欠拟合
ResNet 用一个 res block 代替大卷积层，而且多了一个 ReLU 表达能力上升（除了一开始用 7x7conv s=2 快速下采样省内存）。每过几个 block C 翻倍但也用 s=2 来下采样进行收缩。ImageNet 输入是 224x224，最后缩到 7x7，信息由局部变到整体。更深的网络用 bottleneck 丢弃冗余信息，先用 1x1conv 把 C/4,再用 3x3conv, 最后再 1x1conv 把 C 复原。参数为 17C^2/16，对比 ResBlock 为 2*3^2C。两个 bottleneck 感受野与原来相同，但是参数量大大减少。
NAS：把神经网络的结构变化当成强化学习中动作（比如搜出一个又小又快的网络架构布置在边缘端），是一个比较成熟的技术

Semantic Instance Segmentation，grounding，granularity 颗粒度越来越小
语义分割是 Dense prediction，对每个 pixel 都给出预测，输出在空间上与输入同维，但是我们必须降低 resolution 然后升回来，不然 Conv 开销太大。Auto-encoder 可行的原因是信息有冗余可被剔除，最后把冗余加回来。但是要注意不能出现 irreversible 的信息丢失。低维 latent z。
上采样除了 unpooling 还有可学习的转置卷积，stride=2 使 resolution 翻倍。这种 bottleneck 结构用同样的 conv 次数达成大得多的感受野，提供更广的 context，而且内存 cost 更低。FCN 在 bottleneck 中还存有原始 pixel 的位置信息，提升为 **UNet**：在相同 resolution 的两端加 skip link，每一层都不断聚合原始语义，bottleneck 只需提供 global context。 Mean IoU 评估
评估对每个类分别统计：IoU 交集比并集，全类别 40%已经很不错。1-IoU loss

Pinhole camera 有亮度和清晰度的 trade-off：aperture size 模型太简单少用 Lense 近轴折射，当光线远离光轴会有畸变。z'=f+z0 代替 f 仿射变换保直线但平行线可能相交，出现弯折说明有畸变 barrel&pin
投影变换中相纸坐标系有 offset，metric 到 pixel 有单位换算，最终参数是αβCxCy 形成内参矩阵。除以 z 导致近大远小。K 加上θ共 5 个自由度。
外参 RT，R 三个自由度，外内参共 11 个自由度。P=K[R,T]Pw
弱投影视 z 为常数，线性，不用齐次坐标。正交投影只用平行光。
把世界坐标系建在校准架上，检测网格线交叉点可知像坐标。用 6 个点列 12 个方程来求解内外参。实际采用>6，因为可能有矛盾现象，参考 RANSAC。
同样先对 m 加 norm=1 的限制再用 SVD 求解，最后再 unnormalize
需要不在一个平面的数据且非交线，因为近大远小，单张图不能判断深度大小
K 中αβ控制的是相机不同像素射出光在现实世界的夹角（视场角 field of view）
用 reprojection 来检验，误差 1pixel 以内。单组数据出错可能外参出问题。

激光雷达给出 ray depth 沿光线，而深度图记录的是 z depth。知道相机内参与 z，自然能求出 x, y, 称为 depth 的 backprojection。Depth 不是真 3D 因为**必须要知道 K 才能还原**。显 3D 会给你三维具体位置和两点距离。
Disparity=Bf/z，注意相纸上对应物要在同一条 IP polar line 上，B 是双目间距
Stereo sensor 对阳光直射鲁棒且 cost 低，但对应点难找：反射 specular，折射 transparent，无标志纹理重复 textureless，特别黑吸光都不行。
主动双目（把一个换成投影）解决 textureless，但红外投影在太阳光下失效。
Mesh 是分片线性近似。Quad mesh 会有奇点，因此 triangle mesh 用得最多。储存顶点信息，以及用哪几个顶点法向量朝外构成三角形。
点云不是向量而是集合，orderless & irregular，轻便且几何精准。点云等于在二维流形上采样，以面积为权均匀分布。在三角形内取点可以转化为平四中取两边加权和，再转回三角形内。Uniform sampling 只是在数学期望上 uniform 而实际看起来仍有疏密，因为局部方差有涨落。先用 Uni 采样一个备选超集，再迭代取最远的点（FPS，更少 miss local structure）Chamfer Dis：双向取最短距离求和。比 EMD 更易实现，后者是全局优化，对采样敏感

Image Gradient: finite difference. Direction: intensity 变化最快的方向

用卷积描述滑动窗口 filtering: $(f*g)[n]=\sum_{m=0}^{N-1} f[m]\cdot g[n-m]$

求导$\frac{\partial}{\partial x}(f*g)[x]=\left(\frac{\partial f}{\partial x}*g\right)[x]=\left(f*\frac{\partial g}{\partial x}\right)[x]$

卷积定理$\mathcal{F}\{f*g\}=\mathcal{F}\{f\}\cdot\mathcal{F}\{g\}$时域卷积对应频域乘积

Moving average: smoothing effect, remove sharp features

卷积是线性运算，能描述线性 filter，不能描述非线性的

Edge: 一个方向上 pixel intensity value 变化显著，垂直方向上几乎不变

Criteria: Precision=TP/TP+FP, Recall=TP/TP+FN. 准确度：检出的都是对的（FP 为错检出）recall：应检尽检（FN 为应检未检）TP 为正确检出

同时要求 localization，并限制 single response（去冗余 response）

图像梯度对噪声太敏感，需要 smoothing（Gaussian Filter-Low Pass）

对 filter 卷积后求导可得所需的图像梯度（卷积+求导可合并成一步）

2D Gaussian: g=1/(2πσ^2)exp-(x^2+y^2)/2σ^2, σ控制胖瘦（越参，越大 smoothing 越强，但 localization 会变差，因为图像更模糊）

NMS 去冗余：对每一点 q，在它梯度方向正反各走一步（步长可以作为一个超参），对得到的两个点计算 bilinear interpolation 得到它们的近似梯度值，若 q 的梯度值比它们大，q 就要保留。简化版：对每个点分上、右上、右、右下四个方向，梯度落在哪个区间就用哪个区间方向的邻居作比较。NMS 把一个 multi-pixel 的 ridge 变成 single pixel wide。

Hysteresis Thresholding and **Edge Linking**: maxVal 起笔，minVal 截断。此外还要求成为 Edge 的 pixel 梯度落在同一区间里（方向相近）

循环此步骤直到所得图像不再改变。Canny 证明了 Gaussian Filter 的一阶导近似最优了信噪比和 localization 的乘积。(Why Gaussian)

Line 是比 Edge 更低维的表征（直线方程）拟合直线最简单：最小二乘法本质是对残余取 L2（取不同的范数改变了优化问题的 Energy landscape）

最小二乘是凸优化问题，有解析解$B=(X^{\mathrm{T}}\mathrm{X})^{-1}X^{T}Y$ 。但最小二乘法对离群点（outlier）非常不鲁棒（非常敏感），且无法处理竖直线

采用直线一般方程，解析解 Ah=0 ， h=(a, b, d), 限制||h||=1 防止 0 解（不直接限制 a 或 b，防止排除一些直线）。Minimize ||Ah||: 对 A 做 SVD，h 即为 V 的最后一列（对应奇异值最小的维度）

RANSAC: w=inlier 比例，n=自由度个数（线 2 面 3），k=所选 sample 数

Prob.至少有一个 sample 成功 1-(1-w^n)^k, 因此 n 尽可能小 k 尽可能大

Loop：随机选 n 个点构建 hypothesis 算 inlier，足够多就对所有 inlier 拟合（SVD）结束后要再算一次 inlier，循环几次找 inlier 最多的一个。

RANSAC 只能解决低维（n 小）问题，而且 outlier 太多则计算量爆炸

这些传统方法（modular-based system 模块化）每一步都在努力提升鲁棒性（去噪），但是仍有很多没预想到的错误，需大量后处理

模块化与端到端间平衡：E2E 可解释性差、训练优化复杂、修改困难

Corner：在领域内梯度有 2 个以上的主要方向。Corners are salient, repeatable, sufficient, easy to localize，因此是好的 keypoint

把一个 window w 移动(u, v)，I[x+u, y+v]-I[x, y]≈Ixu+Iyv. 移动后的 Energy Ex0y0(u, v)≈[u, v]M(x0, y0)[u, v]T. M=w 卷积 I 的二次偏导数对称矩阵。M 可对角化且半正定，E 是个半正定二次型，E 的标准型是一个抛物面。若两个特征值都≈0，flat。若一个远大于另一个，则为 edge。否则 corner。若为 rectangle window 要求两个特征值都大于 b，即长度一半。比值要在 1/k 和 k 之间。近似方法：θ=det(M)-αtr(m)^2-t，t=b^2/2，α为超参。

α和 k 都为控制特征值的经验常数，通常α取 0.04~0.06，θ>0 为 corner，约等于 0 为 flat，小于 0 为 edge。θ是 x0，y0 的函数，即检测位置。

采用 Gaussian 函数可以做到 rotation-invariant。最后还要 NMS 去噪。

等变性：输出随输入改变。不变性：输出固定。

Low-level vision：Image processing, edge/corner detection, feature extraction

Mid-level vision: Grouping, inferring scene geometry (3D reconstruction), inferring camera and object motion

High-level vision: object recognition, scene understanding, activity understanding

Line fitting 可以看做我们找到了一个两个参数的线性模型，两个参数的 learning 模仿这个想法，我们通过 Sigmoid 加线性模型找到了一组函数，然后通过 learning 挑一个最好的。但是这一族函数不一定包含那个理论最优的函数。

Loss 函数采用 NLL，求平均 loss 的最小值。推导：MLE 取负对数

优化方法：梯度下降。学习率太大-overshoot，太小-迭代太慢

Naïve GD will trap at local minima: 如果用所有 data 直接训练，必然会陷入离起始点最近的 local minima，因此采用 SGD 随机梯度下降/mini-batch，随机采样一个 Batch，算它们的 average gradient。训练速度快而且有几率跳出 local minima

NLL 的下降与 accuracy 不正相关，训练集和测试集上有 generalization gap，模型需要有泛化能力。

单层网络只能处理线性可分类问题，不能处理更复杂分界面，模型的 capacity 和 expressivity 不够（欠拟合）。因此引入 MLP。

线性层相当于矩阵乘，堆叠线性层相当于多个矩阵做乘，实际只有一个矩阵，因此无意义。堆叠非线性会使输出被局限，如 Sigmoid 嵌套。因此线性非线性交替。

Bias 在经过 activate 后可以造成一定扭曲，表达力增强。

Input 28x28-hidden vector-output 1dim. 维度下降对应着信息提取，剔除无关信息，最后输出预测值，只和 hidden vector 有关。

层数多就需要 BP 来更新参数：ReLU 方便 BP，Sigmoid 和 tanh BP 不友好。ReLU 局部线性，但是次数够多就会形成复杂的高位多面体，很好地分割空间。

MLP 因为把图片 Flatten 成一个高维向量，对高分辨率图像很 expensive，而且打破了 local structure，因此对天然具有线性结构的简单输入有效，但是不适用于二维信息复杂的图形。MLP 对平移和旋转没有鲁棒性。

一个卷积核包含 K 的 Filter, 每个 Filter 对应不同的提取特征, 每个 filter 的 channel 数和输入的 channel 数一致。设输入图片为 NxN，Filter 为 FxF，padding 为 P，输出边长为 **（N+2P-F）/stride+1**. 一个卷积层的参数量为 **F^2CK+K,** 每个 filter 是 F^2C 个参数+1 个 bias。因为 bias 也是学习的，因此一个 filter 共享一个即可。一个 conv layer 有四个超参 F, K, S, P，S 对维度收缩速度影响大。Bias 本质是设置了一个 threshold，高于-b 才能通过 ReLU，它调整输出基线，可以增强表达力。

Conv 层输出是 activation map，在一个 Conv block 之后要经过 pooling 缩减去粗取精，防止过拟合，减少计算量和内存消耗，并扩大了感受野。2x2 pooling 把输入的 WH 减半，C 不变。Pooling 层无参数。

FC is densely connected，参数量为 W1W2H1H2CK 巨大，但 Conv 参数量小，因为**权值共享**，和 WH 无关。而且输出中的 Cell 有**近场效应**，不是和输入全连接。近场效应使得信息在平面逐渐延展。FC 是 CNN 的超集，FC 只要把多余的参数学成 0 就退化成了 CNN。FC 表达力更高，但是难优化，全域都有致密的 local minima，很难找到一个好的，但是 CNN 因为其稀疏性就容易优化（各处有差不多深的 local minima，质量都比较高）。权值共享则确保了平移等变性，pooling 让模型对小的旋转也没响应。

Shuffle dataset：使 Batch 和整个 dataset 的特征匹配

对 data 的每一维进行处理，-mean /std，要保证 equivariant，对一张图必须全局减去同一个 mean。目的使数据变得 learning friendly，风险是导致一些信息丢失。

Zero-mean data 的好处：less sensitive to small changes in weights，易优化。

数据预处理的方法可以很多样。可能的问题：若一张图所有 pixel 都是 red=100±1，视觉上看上去没什么差别，但是 normalize 大大放大了差异，使原图内容改变。|| pointnet 对每一维取 max，排列不变性，最后一步形成 global feature。PN 非常鲁棒，最后提取的基本是边框点集，对增删点很鲁棒。PointNet++改进了没有 local feature 和依赖绝对坐标，FPS+grouping+pointnet