

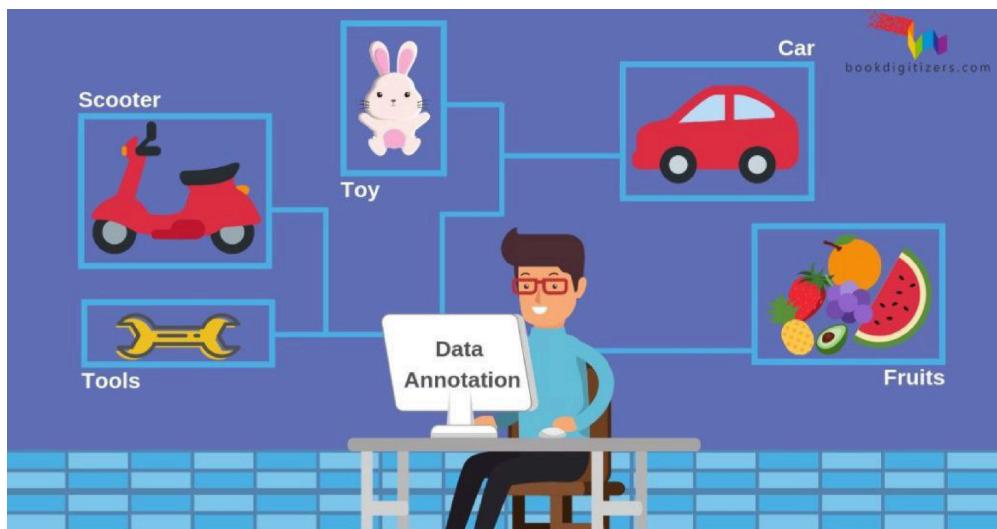
Lecture 11 - Self Supervised Learning

本章节主要围绕着自监督学习展开，在第一章节主要介绍一些准备工作以及领域发展的动机，第二章节介绍首个自监督学习的模型，被称为 Task-specific 模型，第三章介绍 Pretext Tasks (借口任务)，作为通用的视觉表征，可以在之后针对特定任务进行微调；最后一章节介绍最新的自监督学习——被称为 Contrastive Learning (对比学习)

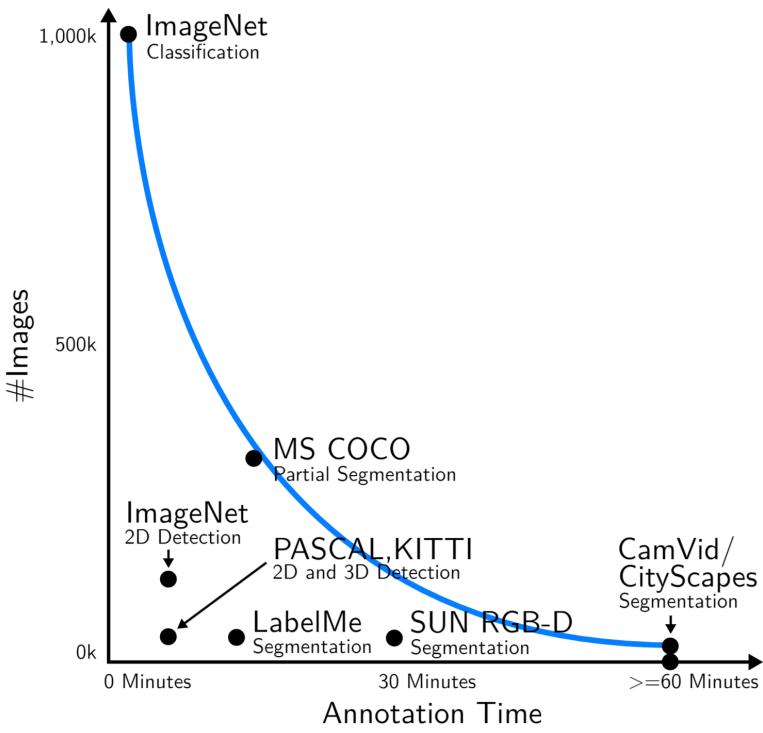
11.1 Prelimiaries

到目前为止，我们介绍的所有“监督学习”的任务，目的都是希望能够评估、获得一组理想的参数，而为了实现这一点，我们就需要输入大量的标注好的数据来进行训练（标注可以理解为，告诉模型在这个任务上，什么属于好的，什么属于坏的）

- Data Annotation



- 为了使参数评估的效果更好，监督学习需要非常庞大的标注数据
- 这些标注数据又必须具有统一的标准（例如我们统一将所有种类的兔子都标注为“兔子”，不能有的是“棉尾兔”，有的是“兔”），所以这个过程需要非常多的人参与，也需要设置严谨的验证环节
- 下图展示了关于不同数据集上，标注与对应所需的时间



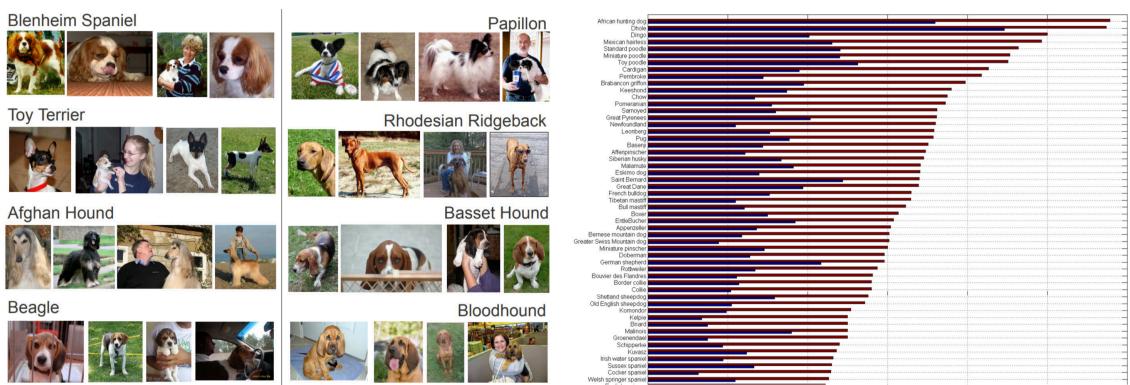
- 可以看到，一般来说分类任务的标注工作量比较小，分割任务的标注工作量比较大（因为需要把图中每个分割的区域精确地绘制出来）
- 因此，对标注数据集的人而言，就有一个权衡——要么做一个粗略但很大的数据集，要么做一个精细但很小的数据集

让我们看一个具体的数据标注任务，图像分类任务中经典的 ImageNet 数据集

- Image Classification [What I learned from competing against a ConvNet on ImageNet](#)
 - 在 ImageNet 数据集中，经统计研究，认为可能是由于人为标注的错误有以下几种
 - **细粒度的识别 28 例 (37%)**
 - 在 ImageNet 数据集中有超过 120 种狗的图片，但是对于一个并非犬类专家的标注者而言，想要分清楚每一种狗的类别很难（同一种狗也可能看上去非常不同）
 - **类别未知 18 例 (24%)**
 - 对于标注者而言，他们需要给每一幅图片分配 1000 种类别中的一个，但是他们不一定能够完整的记住 1000 种类别，有时候他可能甚至不知道某些类别已经存在于 1000 种当中

■ 不充分的训练数据 4 例 (5%)

- 标注者在正式进行标注工作前也经过了训练，但是标注者在每个类别下只经过了 13 个样本的训练，这对于概括一个类别来说恐怕是不够的
- 另一方面来说，根据计算，ImageNet 标注的时间相当于 22 个人全天候的坐在电脑前持续 1 年
- Standford Dog Dataset [Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs](#)



- 这是 Standford 大学的研究团队制作的一个犬类数据集，它们是 ImageNet 的一个子集，共包含了 120 种犬类，20580 份样本
- 即使是这样，分类标注的难度仍然非常大，更何况对于 ImageNet 数据集而言了

当我们从分类任务转向分割任务，问题会变得更加困难

• Dense Semantic and Instance Annotation



- 对于分割任务而言，有时候目标可能非常小，只有几个像素，因此即使放大图像仍然难以识别，尤其是类似于 Cityscape 这样复杂的环境
- 因此必须耗费很多的时间，而且需要很多人一起标注同一幅图来降低错误率（每幅图可能需要大约 90min 的时间）
- Stereo, Monocular Depth Estimation and Optical Flow



- 对于立体视觉或单目的深度估计，或者光流任务的数据，例如 KITTI 数据集
- 其标注任务对于人类而言几乎是不可能的——对于同一棵树不同角度的几张图片，你很难精确的看出哪些点是对应的同一个点，哪些点不是
- 因此，实际上 KITTI 数据集采用了激光雷达 LiDAR 获得地面深度数据的真实值，并且进行手动分割和跟踪
- 然而，对于屋顶这样场景的光流任务，我们只能通过一些复杂且带有估计性的方法来估算

- Human Labeling is sparse

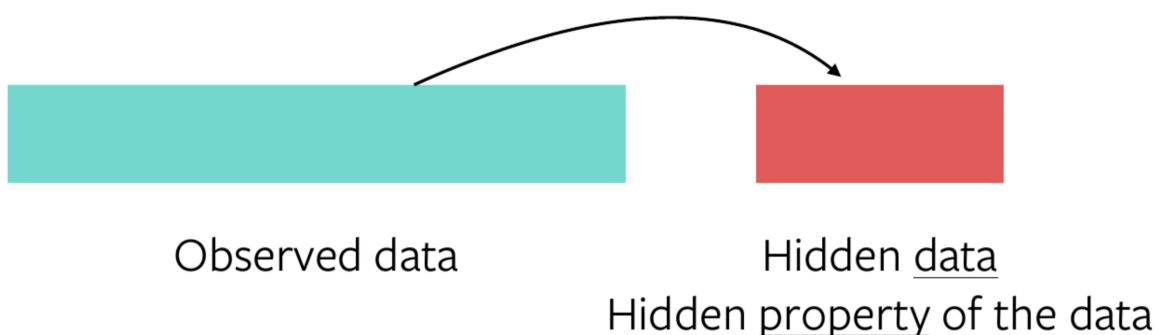


- 我们之前一直在讨论的是关于机器如何从带标签的数据集中“学习”，那么人类是如何学习的呢？人类同样没有太多的标签可以进行主动感知
- 正如图中这个例子，当从未见过狮子的小孩看到狮子，父亲会向他进行解释——这种过程可能会出现 1 次、2 次、3 次，但是一般来说在第 3 次以后，小孩就已经能够自主认出狮子——也即，对于人类而言，向他展示一个样本仅仅 3 次左右之后，他就可以记忆、识别一个新的实体，这是一种相当了不起的能力
- 目前，计算机还无法完全实现这种能力，但是本章要讨论的技术正是向着这种方向发展——如何在没有大量，或者没有任何标签数据的情况下进行学习
- 人类的学习主要通过与外界的主动交互和被动观察



- 这是人类互动的一个例子，它发现这是一个可以拿、有一定味道、晃动会发出响声的东西；
- 但是目前机器还无法做到这一点，我们能看到的只有 xy 这样的数据对作为标签

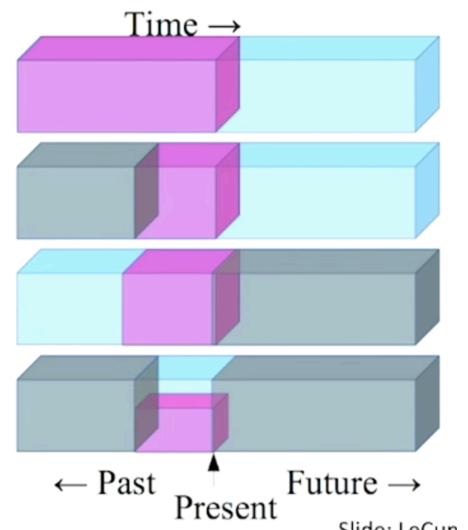
- Self-Supervision



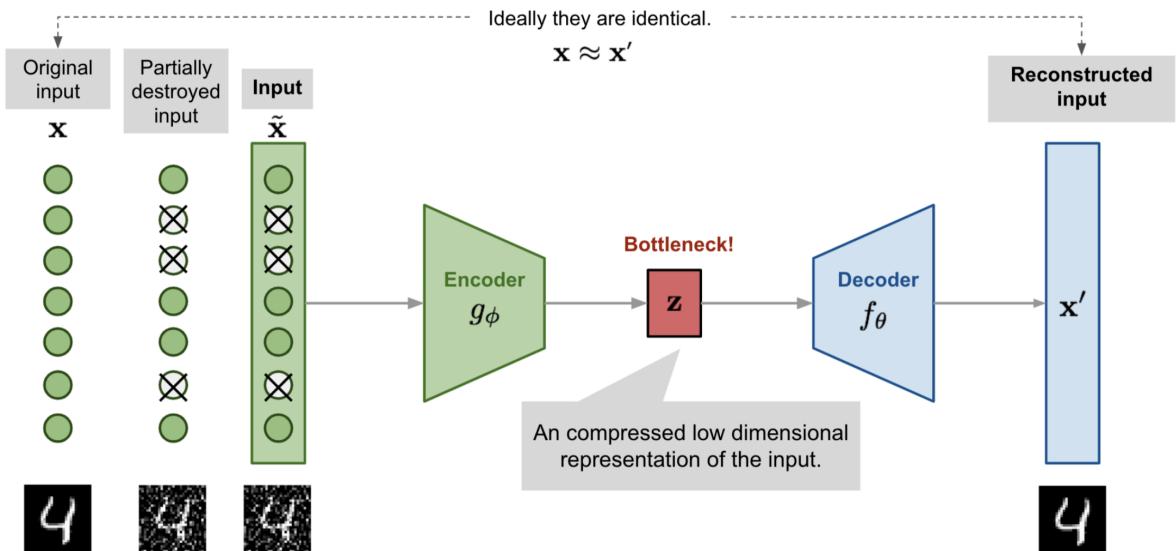
- 自监督的理念

- 从没有标注的数据本身获得标签
- 怎么做到这一点呢？举个例子，假如输入的数据是一幅图，那么我们遮挡这幅图的部分区域，然后根据未遮挡的部分来预测遮挡的部分 (Predict parts of the data from other parts)
 - ▶ Predict any part of the input from any other part.
 - ▶ Predict the future from the past.
 - ▶ Predict the future from the recent past.
 - ▶ Predict the past from the present.
 - ▶ Predict the top from the bottom.
 - ▶ Predict the occluded from the visible
 - ▶ Pretend there is a part of the input you don't know and predict that.

- ■ 我们有很多办法来做到这一点，如上图所示



- 从过去预测未来
- 从最近的过去预测未来
- 从现在预测过去
- 从底部预测顶部
- 总而言之，我们从输入的一部分来预测输入的另一部分，这就是 Pretext Learning (借口学习) 的思想
- Example: Denoising Autoencoder [Extracting and composing robust features with denoising autoencoders](#)



- 这可能是自监督学习最古老的例子之一，称为自去噪编码器 (DAE)
- 这里的 Encoder 和 Decoder 也是一些参数化的神经网络。我们从 MNIST 数据集中取一张样本，然后对其添加部分噪声，将这个噪声版本的样本作为输入，然后训练编码-解码器，试图预测原来的输入样本
- 也即，我们基于原本的数据重新构造了一个输入数据，然后将原本的数据作为预测的标签
- 训练完成之后，我们保留的是其中 Encoder 的部分，也即可以将带有噪声的样本预测为去噪声的样本的特征编码器——这个过程是自监督学习中的一个阶段，称为 **Pre-train**
- 然后我们通过添加一个 head，例如分类用的 head，在此基础上微调 (fine-tune) 为别的任务，例如分类头，将其微调为一个分类任务

对比总结一下我们目前遇到的几种学习任务

- Reinforcement Learning (强化学习)

- 从环境中得到的稀疏的奖励主动探索，通过一个代理对模型做出奖励或惩罚措施，从而学习模型的参数
- **Unsupervised Learning** (无监督学习)
 - 用没有任何标签的数据集来学习模型参数 $\{x_i\}_{i=1}^N$
- **Supervised Learning** (监督学习)
 - 用带标签对的数据集来学习模型参数 $\{(x_i, y_i)\}_{i=1}^N$
- **Self-supervised Learning** (自监督学习)
 - 利用数据集自身的部分作为标签进行学习 $\{(x_i, x'_i)\}_{i=1}^N$

学习方式	数据需求	优点	缺点	典型例子
RL	依赖环境的奖励信息	适合动态环境，可以优化长期策略	学习过程不稳定，设计奖励函数难度大	Deep Q-Learning、Actor-Critic
UL	无标注数据集	不依赖标注，适合挖掘数据隐藏模式	难以评估效果，优化目标不明确	K-means、PCA、GAN、VAE
SL	大量标注数据集	性能稳定，适用任务广泛	标注成本高，对数据质量依赖大	ResNet、回归、结构化预测
SSL	无标注数据集	无需人工标注，特征迁移能力强	伪任务设计复杂，训练耗时	SimCLR、Contrastive Learning、BERT

How Much Information is the Machine Given during Learning?

Y. LeCun

► “Pure” Reinforcement Learning (cherry)

- The machine predicts a scalar reward given once in a while.

► A few bits for some samples

► Supervised Learning (icing)

- The machine predicts a category or a few numbers for each input
- Predicting human-supplied data
- $10 \rightarrow 10,000$ bits per sample

► Self-Supervised Learning (cake génoise)

- The machine predicts any part of its input for any observed part.
- Predicts future frames in videos
- Millions of bits per sample



- ○ 这是来自法国计算机学家 Yann LeCun 的一幅非常著名的图，他用一个黑森林樱桃蛋糕来比喻，在几种学习方式中实际给予机器的信息量
 - “Pure” Reinforcement Learning
 - 比作图中的樱桃
 - 说明它从环境中获取的信息很少，实际上只是利用了标量级别的信息，就像整个蛋糕上的樱桃
 - 大概只有几个 bit / 样本
 - Supervised Learning
 - 比作图中的糖霜层
 - 对于每个输入，机器预测一个分类结果，或者一些 2D 边界框，或者几个数字，受限于人工提供的标签部分
 - 大概 10——10,000 bit / 样本
 - Self-supervised Learning
 - 比作图中的蛋糕主体
 - 对于每个输入，机器可能预测它观察到输入的任何一个部分，是能够利用数据占比最大的一种学习方式
 - 大概上百万 bit / 样本

最后，是本章节的 PPT 课件的致谢页，我也原封不动的放在这里，感谢这些计算机科学家的工作！

- Credits
 - Ishan Misra — Self-supervised learning in computer vision
[YouTube Video](#), [Slides](#)
 - Stanford CS231n — Convolutional Neural Networks for Visual Recognition
[CS231n Website](#)
 - Y. LeCun and I. Misra — Self-supervised learning: Dark matter of intelligence
[Facebook AI Blog](#)
 - Lilian Weng — Self-Supervised Representation Learning
[Blog Post](#)

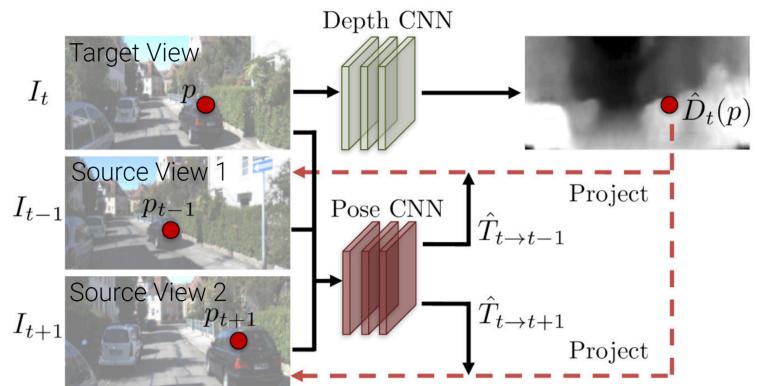
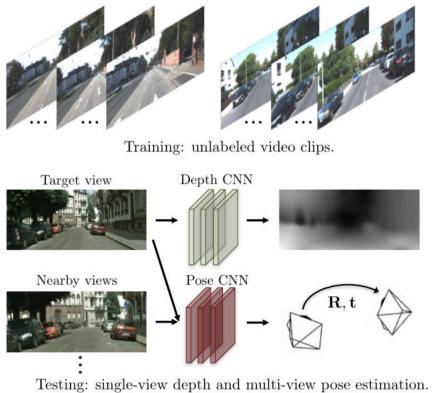
11.2 Task-specific Models

Task-specific 模型，根据 11.1 节内容我们了解了，自监督学习一般来说有两个部分——通用的、预训练的、大型的；以及特定的、微调的、小型的

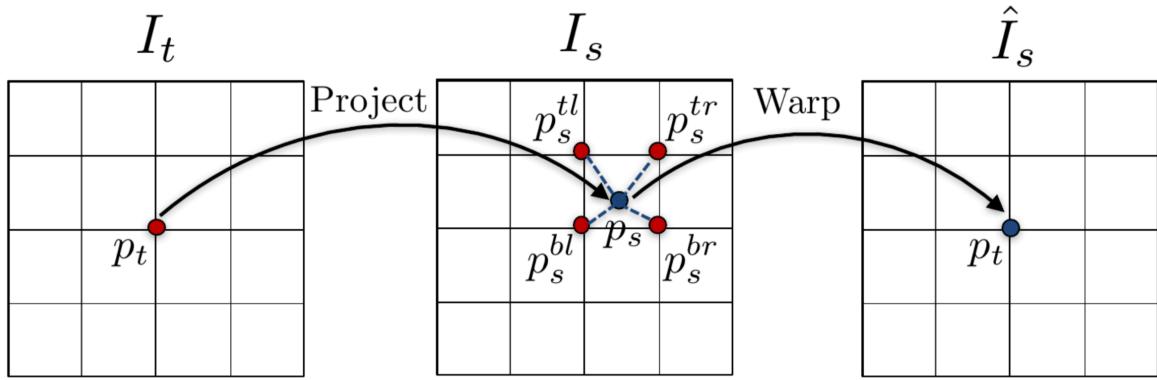
而 Task-specific 模型指的就是后者

下面我们从关于深度预测和自主运动预测的无监督学习模型开始

- Unsupervised Learning of Depth and Ego-Motion [Zhou, Brown, Snavely and Lowe: Unsupervised Learning of Depth and Ego-Motion from Video. CVPR, 2017.](#)



- 我们输入连续的视频帧，训练两个 CNN，分别为图中绿色、红色所示的网络；其中绿色 CNN 用来预测图像的 depth，红色的 CNN，用来预测相邻 2-3 帧的变换矩阵（旋转矩阵 R 和平移向量 t ，共 $3+3=6$ 个自由度）；我们用目标帧 I_t 和源帧 I_{t-1}, I_{t+1} 来称呼这些帧
- 其中右侧展示了这两个神经网络必须结合在一起使用——我们根据绿色的 CNN 预测到 I_t 的某个点的深度；同时我们知道 I_{t-1} 和 I_{t+1} 中的变换矩阵，假设这些结果都是正确的，那么我们可以对 I_t 上的这个点施加这个位置的变换矩阵，从而得到 I_{t-1} 和 I_{t+1} 帧中这个点的位置
- 也即，如果一切都是理想的，我们假设场景是静态的，符合 Lambertian 假设，那么 I_t 图片根据 depth 的预测结果投影到 3D 中的位置，经过变换矩阵，可以得到 I_{t-1} 和 I_{t+1} 的图片；那么对 depth 投影、变换后的 I'_{t-1} 、 I'_{t+1} 和真实的 I_{t-1} 、 I_{t+1} 作差，就可以作为反向梯度传播的 loss 值
- 也即，我们在目标帧的基础上，通过两个 CNN 的预测结果将其“扭曲”(warp)为一个预测的源帧，再将其和真正的源帧进行比对，借助这种图片连续性的 loss 来训练两个 CNN

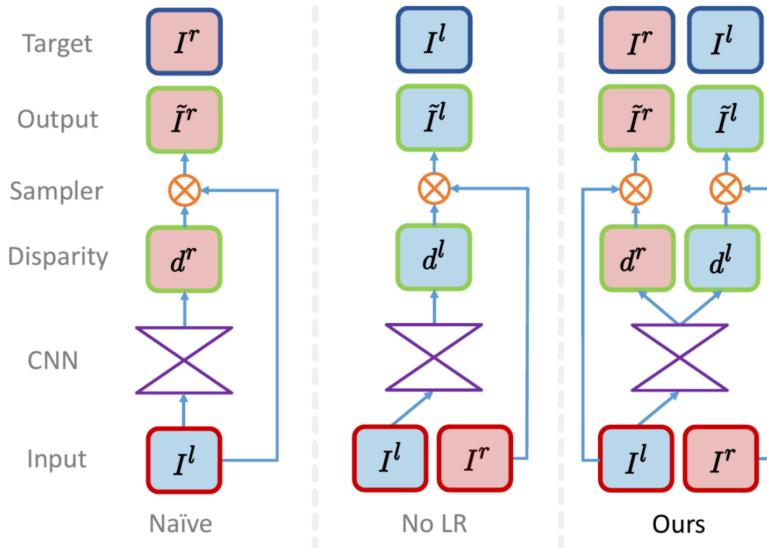


- 具体来说，每个点借助 depth 和 pose 从目标帧投影到源帧的公式如下 \$\$
\tilde{\mathbf{p}}_s = \mathbf{K} \left(\mathbf{R} \mathbf{D} (\bar{\mathbf{p}}_t) \mathbf{K}^{-1} + \mathbf{t} \right)

= 由于 $\tilde{\mathbf{p}}_s$ 投影变换后的结果往往是一个浮点数类型，我们利用双线性插值

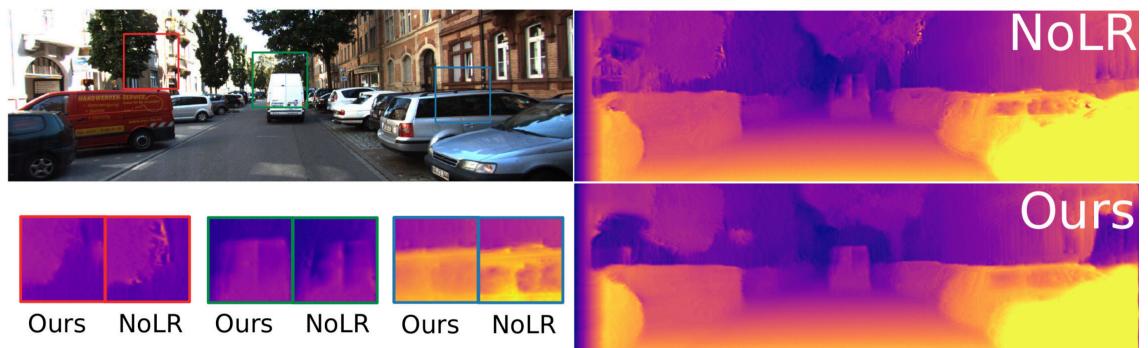
下面介绍另一种，基于立体视觉训练单目深度估计模型的方法

- Unsupervised Monocular Depth Estimation from Stereo [Godard, Aodha and Brostow: Unsupervised Monocular Depth Estimation with Left-Right Consistency. CVPR, 2017.](#)



- 基于立体视觉的好处
 - 我们不再从连续视频帧来预测，而是从同一场景的不同视角（假设是同时拍摄的）来预测，因此支持动态场景
 - 立体视觉方法可以提供更多的信息，使得预测结果更加清晰
- 该研究团队测试了 3 种方法
 - 第一种方法称为朴素 (naive) 的方法

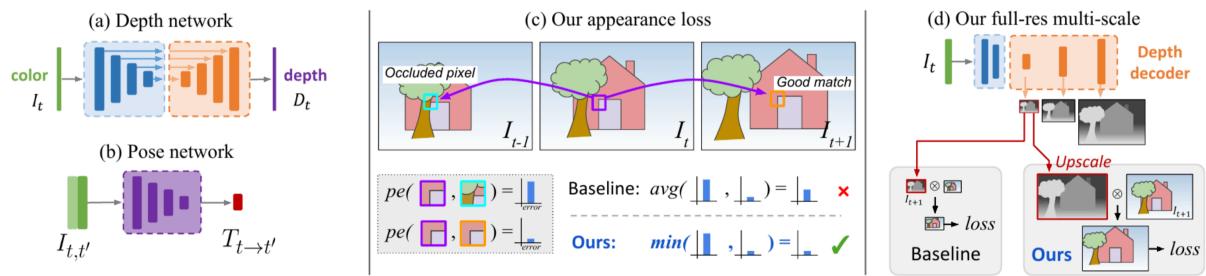
- 将左侧视角输入 CNN，然后将左侧视角变换到右侧视角，根据左侧视角预测出的 *disparity* 得到右侧视角的预测，与变换得到的右侧视角比较得到 loss
- 然而，这样做我们必须从左侧视角预测出右侧视角坐标系下的 *disparity*，而这样做并不合理，因此这种并不是我们希望的方法
- 第二种方法称为没有使用 LR（左右一致性）的方法
 - 将左侧视角输入 CNN，预测左侧视角的 *disparity*，进而得到左侧视角的预测，然后将右侧视角变换为左侧视角，然后做差得到 loss
- 第三种方法是结合了 LR 的方法
 - 将左侧视角输入 CNN，同时预测左侧、右侧的 *disparity*，然后分别根据左侧、右侧的图像得到左侧、右侧的预测图，然后与其对应的 ground truth 进行比较，得到 loss
 - 这种方法的一种好处是，我们同时预测了左右视角的 *disparity*，可以将左右视角的视差一致性作为一个额外的约束条件；在之前的章节讲过，我们知道利用左右视差一致性可以消除一些模糊性



- ■ 采用的 loss：图像一致性、*disparity* 平滑性、左右一致性
 - 从结果中可以看出，结合了 LR 之后，图中的噪声明显减少了，并且增加了一定的平滑度

后续的工作将单目视频和立体视频结合起来，得到了更好的效果

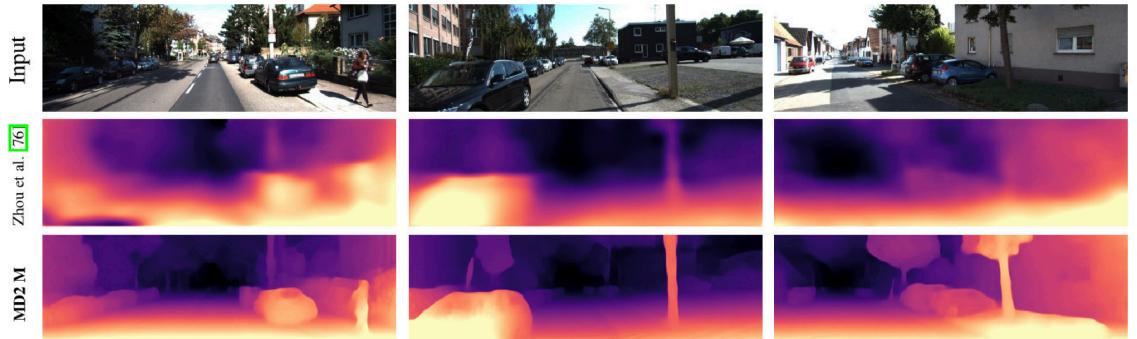
- Digging Into Self-Supervised Monocular Depth Estimation | Monodepth2 leads to significantly sharper depth boundaries and more details [Godard, Aodha, Firman and Brostow: Digging Into Self-Supervised Monocular Depth Estimation. ICCV, 2019.](#)



- 该方法在之前的方法基础上提出了一些改进的点：

- 假如有连续的3帧，同一个点在第1帧中被遮挡，在后两帧中可见；按照baseline的想法，第1帧会有一个较高的惩罚度，和后两帧的惩罚度做平均后，仍然容易超过阈值，得到一个错误的结果——因此，我们改为取最小值，从而避免这一问题
- 原方法中将输入图像缩放到多尺度下计算loss，但是这样容易产生由于复制纹理导致的伪影——因此，它们将图像缩放到各个尺度后，利用双线性插值将其恢复到输入分辨率，然后再计算loss，从而解决这一问题
- 在数据集例如KITTY中，可能会出现两帧之间几乎没有变化的情况（例如拍摄车辆停下等红灯），这种情况下会导致整个方法不适用；如果这种情况在数据集中频繁出现，那么就会影响整个训练效果——因此使用一个自动遮罩算法，将所有重复的帧，或者近似重复的帧筛掉，从而解决这一问题

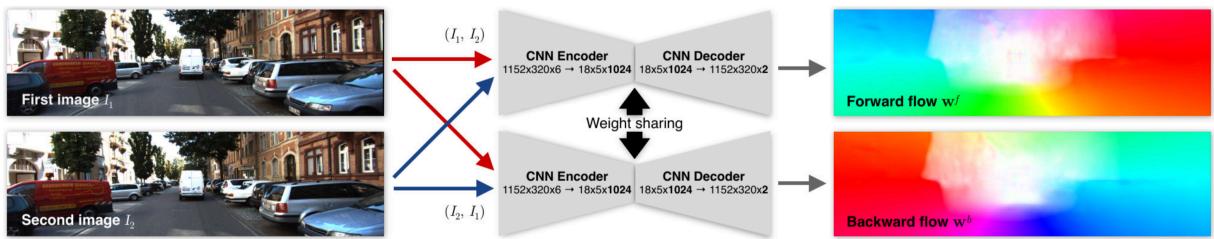
- 效果展示视频



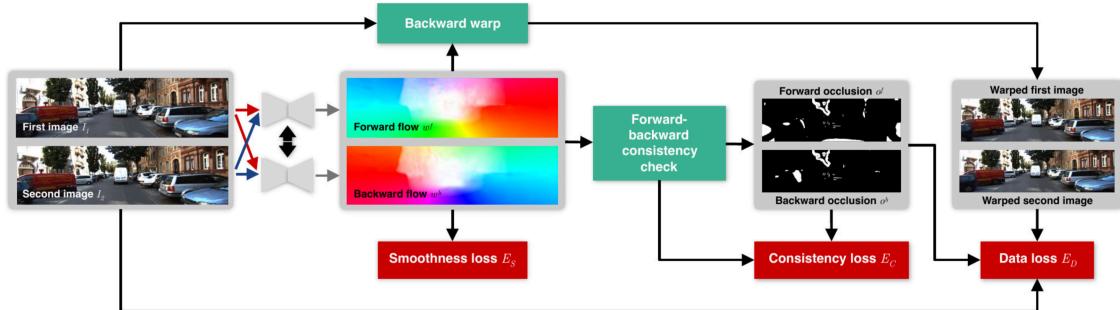
- 可以看到，原来的方法的结果还是模糊的，而这种方法得到的结果非常清晰，尽管在视频中部分边缘仍然会有过度平滑的情况出现，但是结果的可靠性已经大大提升

同样，我们也可以将这种思想应用于 optical flow 的预测任务上

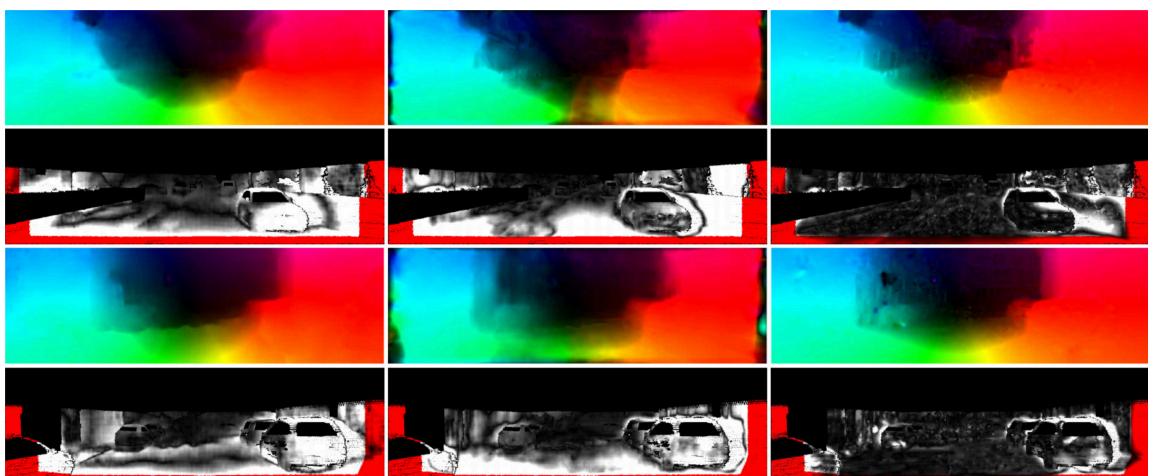
- Unsupervised Learning of Optical Flow [Meister, Hur and Roth: UnFlow: Unsupervised Learning of Optical Flow With a Bidirectional Census Loss. AAAI, 2018](#)



- 这种方法采用了一种称为“双向训练”的思想，将输入图像分别按照 (I_1, I_2) 和 (I_2, I_1) 的顺序输入两个完全一样的神经网络（共享权重），从而分别预测向前、向后两个方向的光流
- 这种思想也被借鉴在了 FlowNetC [Dosovitskiy et al., 2015] 中



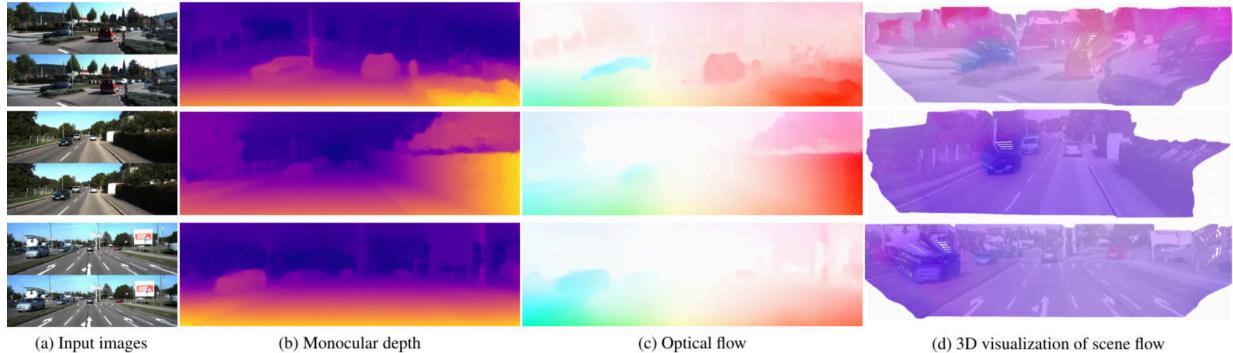
- 总共用到了3个loss：平滑度loss、一致性loss、数据loss
 - 平滑度loss通过对前向、后向光流进行平滑度约束获得
 - 数据loss通过比较光流变换得到的图像和原输入图像之间的差距获得
 - 一致性loss通过计算前向和后向光流的一致性得到
 - 通过数据loss和一致性loss来预测遮挡的情况



- 从左到右分别展示了：监督学习 FlowNetS、无监督学习 FlowNet、自监督学习 UnFlow 三种方法
 - 第一行表示评估的光流场、第二行表示评估的误差（白色越重代表误差越大）

同样，也可以应用到一种称为场景流 (Scene Flow) 的预测任务中，这是一种非常困难的预测任务

- Self-Supervised Monocular Scene Flow Estimation [Hur and Roth: Self-Supervised Monocular Scene Flow Estimation. CVPR, 2020.](#)

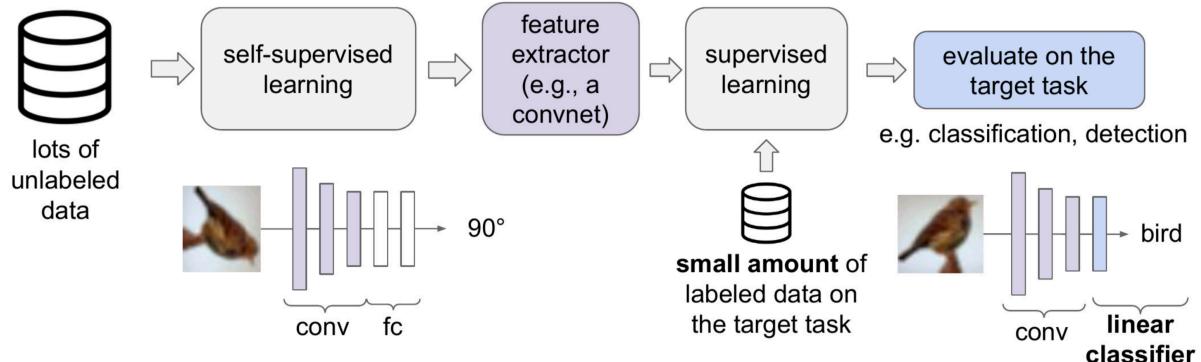


- 结合单目深度和光流，得到了所谓的单目场景流
- 仅在训练时，利用一个立体相机，通过平滑性和图像一致性的 loss 预测每一帧的光流、深度图，然后在推理时将其结合起来

11.3 Pretext Tasks

11.3 和 11.4 实际上都属于 Pretext Tasks 的范畴，只不过 Contrastive Learning 已经发展成为了一个单独的领域，因此将其分为两个章节

- Pretext Task



- 之前说过，Pretext Tasks 的目的是希望构建一个强大的、通用的上游结果，使得我们可以在其基础上进行各种方向的具体任务的下游工作，这个阶段通常称为 **Pre-trained** (预训练) 阶段
- 具体来说，我们希望为预训练阶段定义一个辅助任务（例如旋转图像），我们对巨量的未标注的数据执行辅助任务，从而训练出一个较好的权重

- 举个例子，我们从网上下载数以百万计、百亿计的鸟的图像，然后将它们随机旋转某一个固定步长的倍数，然后让神经网络去预测它被旋转的倍数
- 然后，我们将最后一层 fc 层移除，在一个小型数据集上为某一个特定任务训练一个小型网络
 - 例如我们用 20 张左右的标注的鸟类样本，施加一个线性分类器，训练整个网络进行图像分类任务
- 为什么这被称为 Pretext Task?

From Wikipedia:

"A **pretext** (adj: pretextual) is an excuse to do something or say something that is not accurate. **Pretexts** may be based on a half-truth or developed in the context of a misleading fabrication. **Pretexts** have been used to conceal the true purpose or rationale behind actions and words."

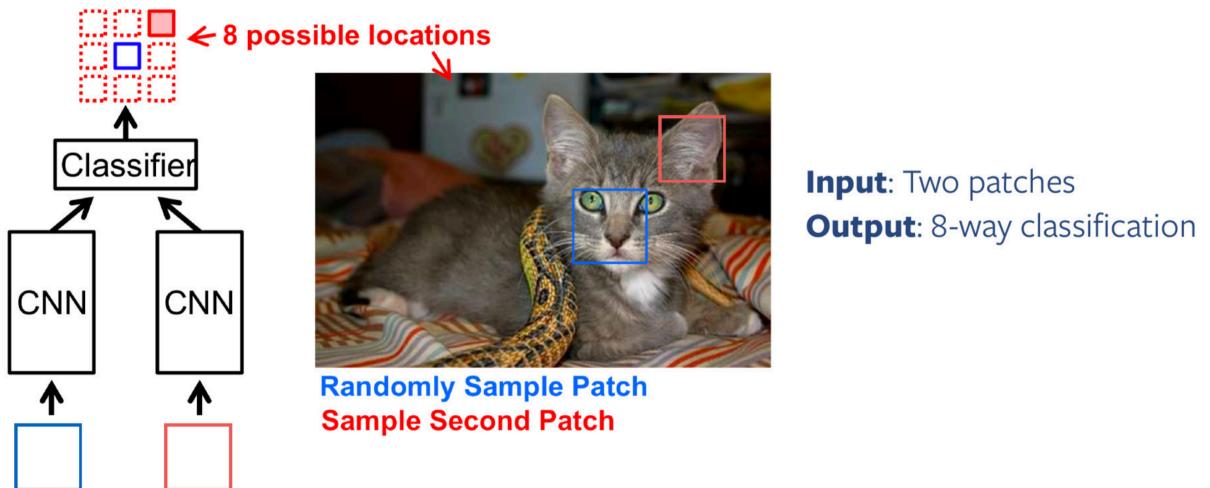


- 上图是 Wiki 上关于 Pretext 的解释——Pretext 是做某些其它事情的借口，它们或许不准确，但是更容易；通过做这些事情来侧面实现自己真正想要达到的目的

当然，可能在现实社会中，这并不是一个很好的意思，不过我们只是借鉴这种思想

下面我们看几个在文献中提出的 Pretext Tasks，这些 Pretext Tasks 都是人工设计的，目前还没有理论依据可以支持它们的优越性，只是根据经验和实验结果证明它们确实在某些方面表现出众

- Visual Representation Learning by Context Prediction [Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.] (Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.)



- 这是关于利用“上下文预测”来做视觉表征学习的研究，具体来说，我们从图像中取一个 patch，然后在这个 patch 的四周（共 8 种可能）随机选取采样另一个 patch；然后将这两个 patch 输入神经网络，输出这两个 patch 的相对位置——也即，上下文预测指的是预测 patch 的相对位置（相对位置取自离散集）
- 研究者希望通过这种任务，可以使得模型学习如何识别物体和它的某个部分
- 一个小游戏

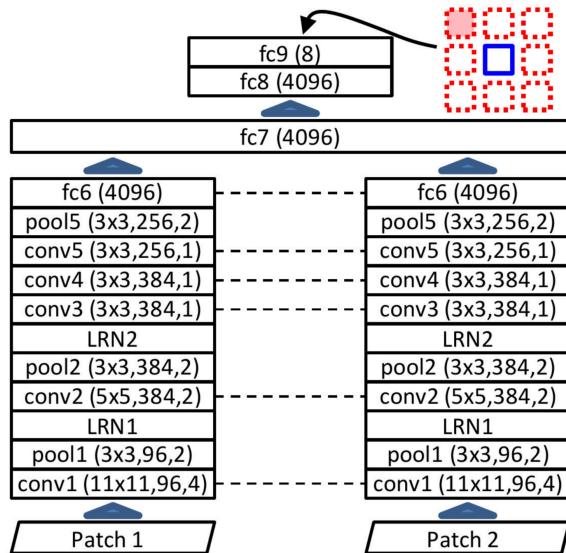
Question 1:



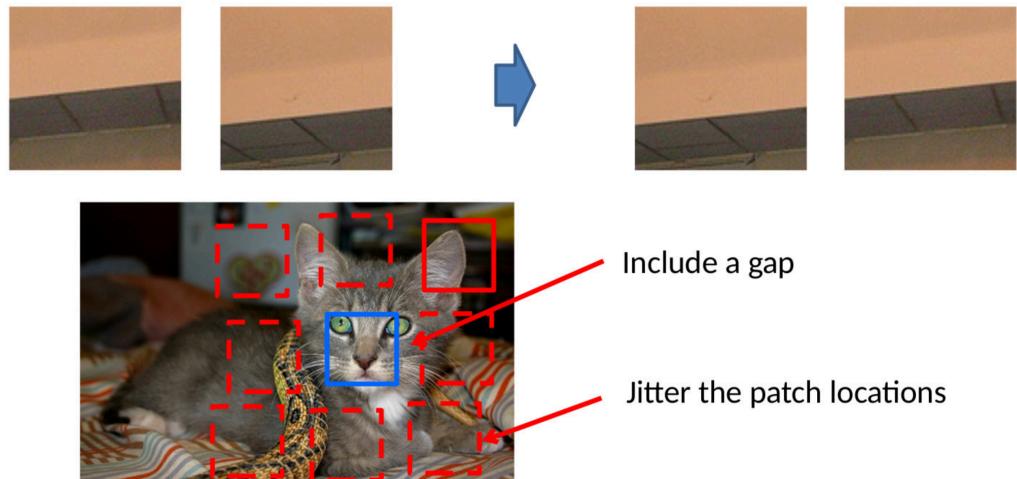
Question 2:



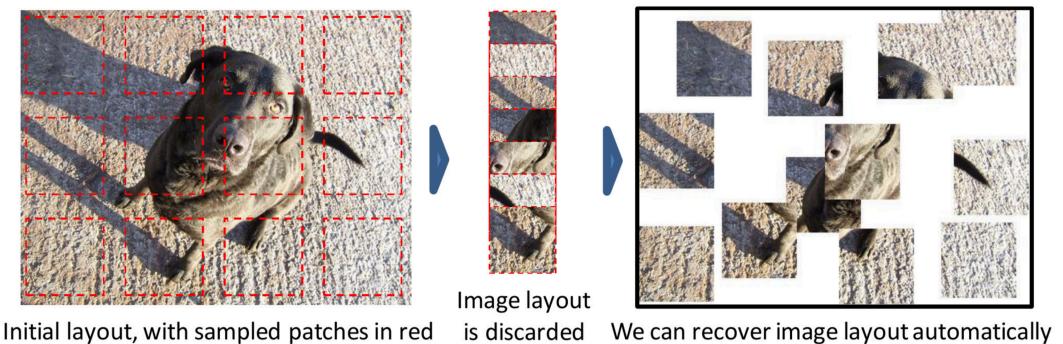
- 我们很可能在报纸之类的地方见过这样的游戏，给出一个物体的某两个区域，猜测它们最有可能出现在原物体的什么位置
- 对人类而言，你可能会觉得，Q1 的左图展示了公交车的灯牌，它应该出现在顶部；Q1 的右图可能展示了公交车的左后侧，因此右图应该位于左图的右下方的位置——我们正是希望训练一种可以实现这种能力的模型



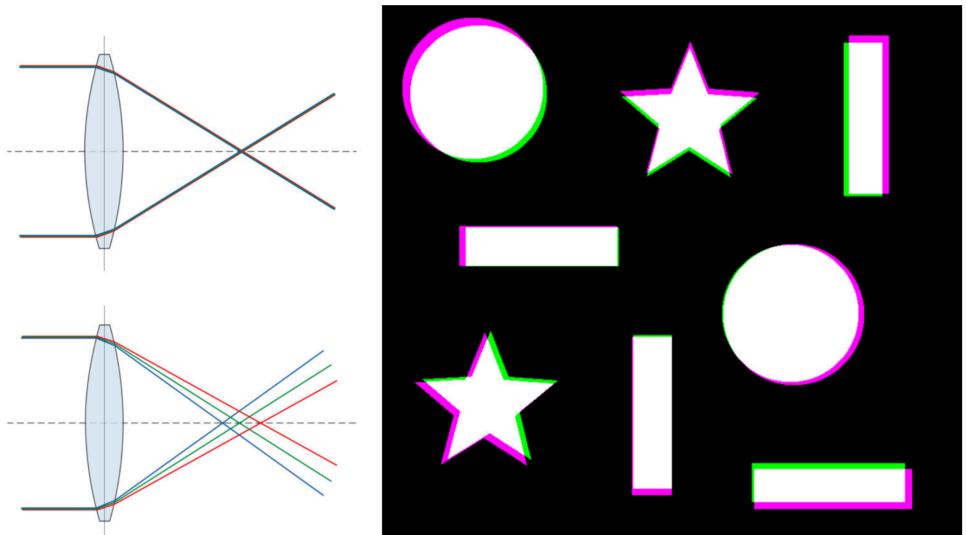
- - 其网络结构如图所示，将两个 patch 分别输入两个 CNN，共采用 5 层卷积，5 层池化，并且使用 LRN (Local Response Normalization) 局部表征归一化层，用于增强局部对比度——最后利用 fc 层将它们的结果组合起来，然后在经过 fc 层输出最终的预测结果
- **avoid trivial shortcuts**



- 在实际应用中，我们还需要注意，网络可能存在“偷懒”的情况——神经网络总是优先选择“捷径”来解决问题；如果我们直接将相邻的 patch 输入网络，那么它可能会直接尝试将它们以不同方式组合在一起，观察其纹理的连续性，从而进行判断
- 对于上游任务的模型，我们希望它尽可能学习到更高级的语义特征，因此我们会在采样第 2 个 patch 的时候，加入一些间隙，而不是采样相邻的 patch；并且采样时还会加入一个空间上的抖动，从而迫使网络学习到更高级别的语义特征



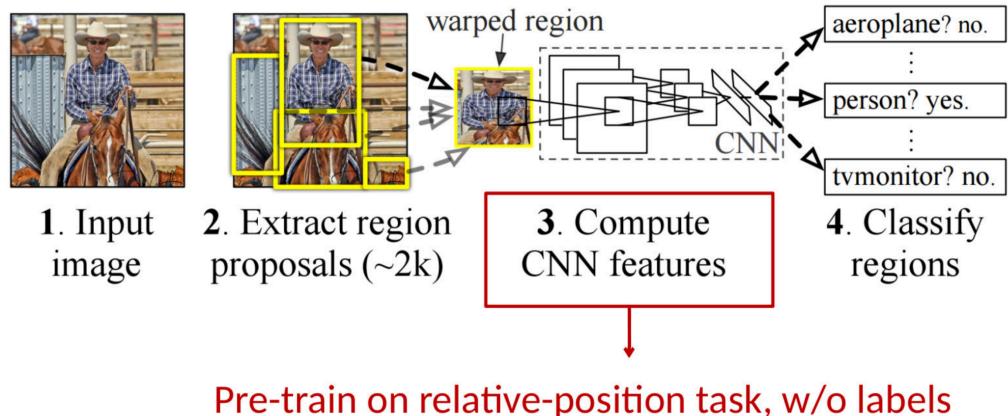
- - 这篇文章的研究团队在汇报时给出了一个令人不可置信的事实——一个网络可以从随机采样的 patch 中学习到图像的绝对位置
 - 具体来说，对于图中这张狗狗的图像，我们在上面随机采样一些 patch，然后输入网络——结果网络可以自动还原图像补丁在图像中的绝对位置
 - 为什么？网络并没有得到任何关于 patch 在图像中的位置编码，却能够得到 patch 在图像中的绝对位置，进而推断出其相对位置，这是怎么做到的？



- ○ 其原理类似于“像差”现象，对于那些镜头质量不是很好的相机，在摄像时，不同颜色的光线由于波长不同，折射角也不同，因此会出现如上图这样不同颜色通道之间不完全重合的现象
- 而具体的偏移程度取决于该图形在图像中的相对位置——换句话说，偏移程度能够表示其在图像中的相对位置
- 因此，模型忽略了其它一些的信息，只是关注于图像的颜色偏差，从而获取其相对位置的特征信息——这又是另一个捷径（我们希望模型关注更抽象的语义信息），因此我们希望解决这个问题

- 解决的方法包括随机去除掉一些颜色通道，或者将图像投影为灰度图，从而去除掉像差的情况

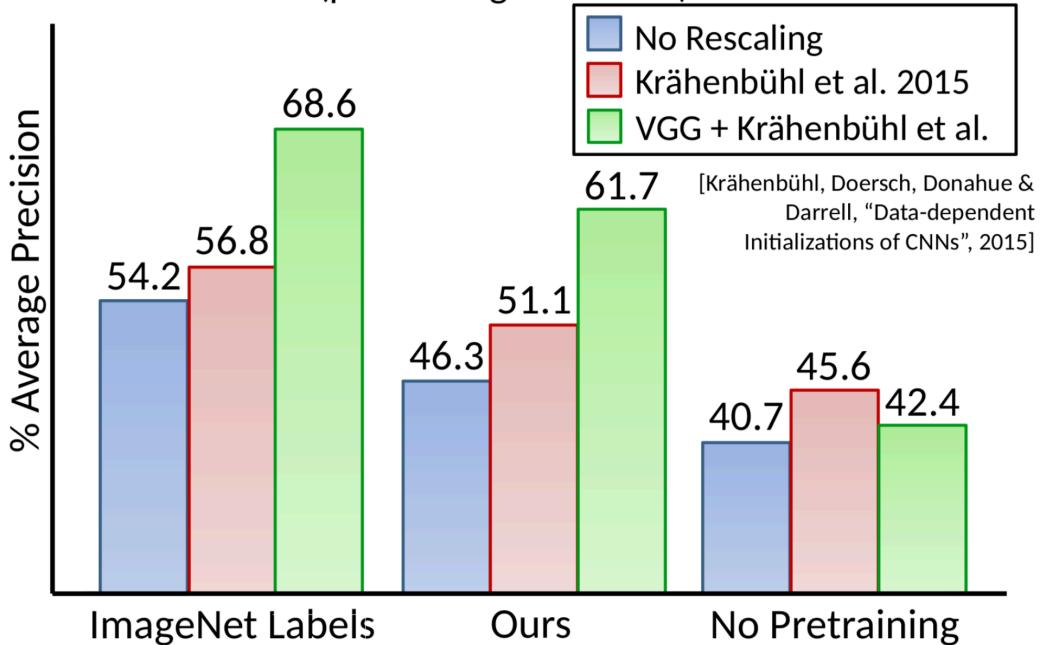
Pre-Training for R-CNN



- 解决这些捷径问题后，网络可以真正学习一些语义上有意义的东西
- 这是我们在 object detection 任务中讲到过的 R-CNN，他们拿这个网络做测试——不使用任何标注的数据，仅仅是采用上游模型
- 然后使用一个较小的数据集来微调剩余的参数

VOC 2007 Performance

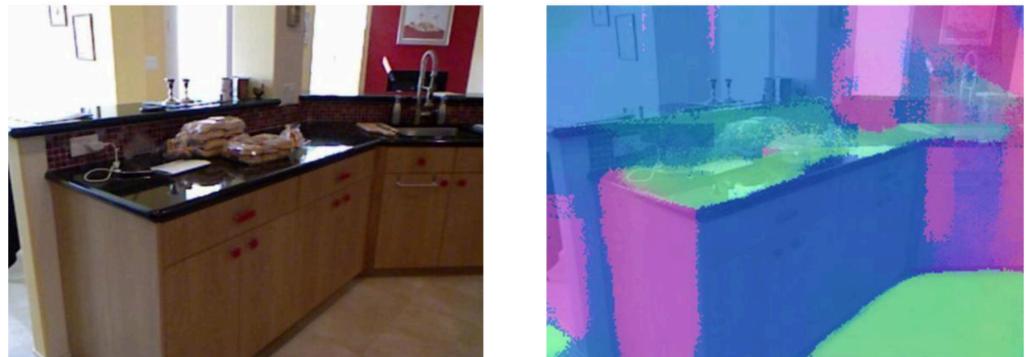
(pretraining for R-CNN)



- 可以看到，不使用预训练的自监督会比使用自监督的结果差很多，尽管自监督方法还没有达到使用 ImageNet 进行监督训练的结果——因为对于 object detection 任务而言，它是一个非常适合识别、非常语义化的任务，因此它从监督方法中的受益真的很大；但是我们可以看到自监督的方法

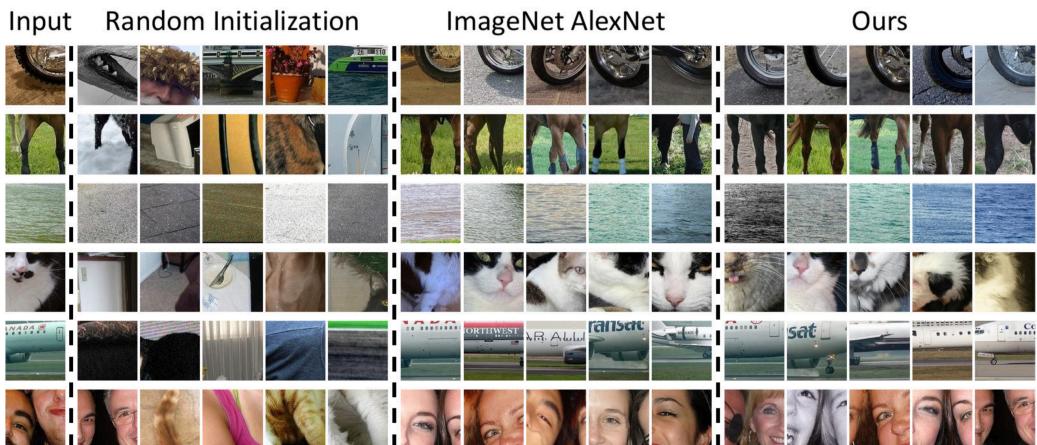
(没有使用任何标注数据) 的结果也几乎可以媲美监督方法的结果了

Surface-normal Estimation

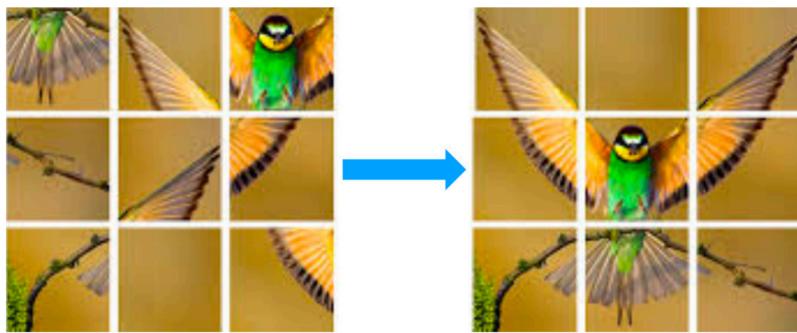


Method	Error (Lower Better)		% Good Pixels (Higher Better)		
	Mean	Median	11.25°	22.5°	30.0°
No Pretraining	38.6	26.5	33.1	46.8	52.5
Ours	33.2	21.3	36.0	51.2	57.8
ImageNet Labels	33.3	20.8	36.7	51.7	58.1

- Context Prediction 的上游模型同样也适用于表面法线估计任务，表面法线估计任务是一个语义目标没有那么明确的任务，因此可以看出，无监督学习（带预训练）方法的结果甚至有时会超过 ImageNet 监督方法的结果



- 为了评估模型学到的特征的质量，我们通常会使用 **Nearest Neighbor Retrieval** (最近邻检索) 方法
 - 我们选择一组输入，然后再图像特征空间（例如 fc6 层特征）中计算所有其它数据与选定输入的相似性，找出最邻近的一组，用于验证模型是否有学习到有意义的视觉特征
 - 由此可以看到，实际上无监督方法可以学习到高质量的视觉表征
- Visual Representation Learning by Solving Jigsaw Puzzles [Noroozi and Favaro: Unsupervised Learning of Visual Representations by Solving](#)

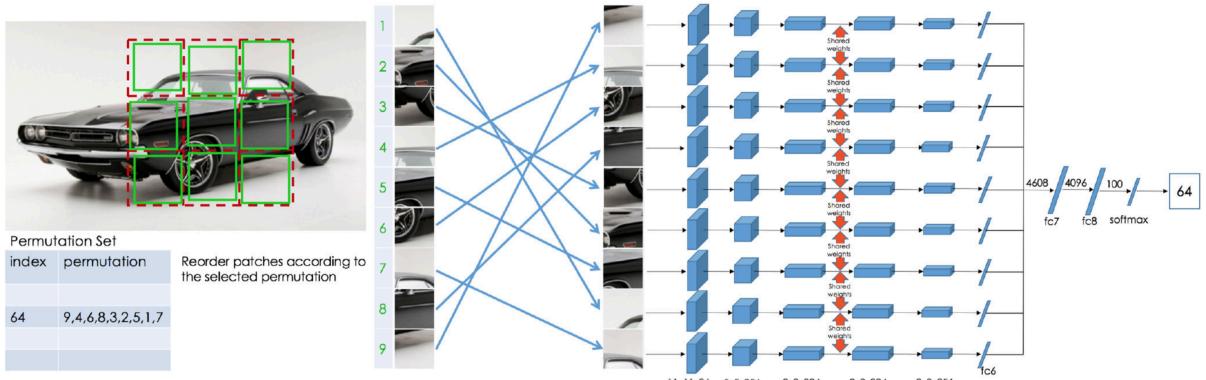


Input: nine patches
Permute using one of N permutations

Output: N-way classification

Set $N \ll 9!$

- 另一个经典的 Pretext tasks 被称为“拼图游戏”，如图所示，他们将一个图像拆分成 3×3 的若干块，然后随机打乱，希望模型能够将其还原为初始的顺序
- 他们使用 N 维向量的方式作为一种输出结果的表示，依据 Hamming 距离选择随机的排列来增加难度
- 理论上，对于 3×3 的情况而言，一幅图的还原方法共有 $9! = 362880$ 种可能，这个输出空间过于庞大，因此他们将输出固定在 1000 个可能的随机排列中



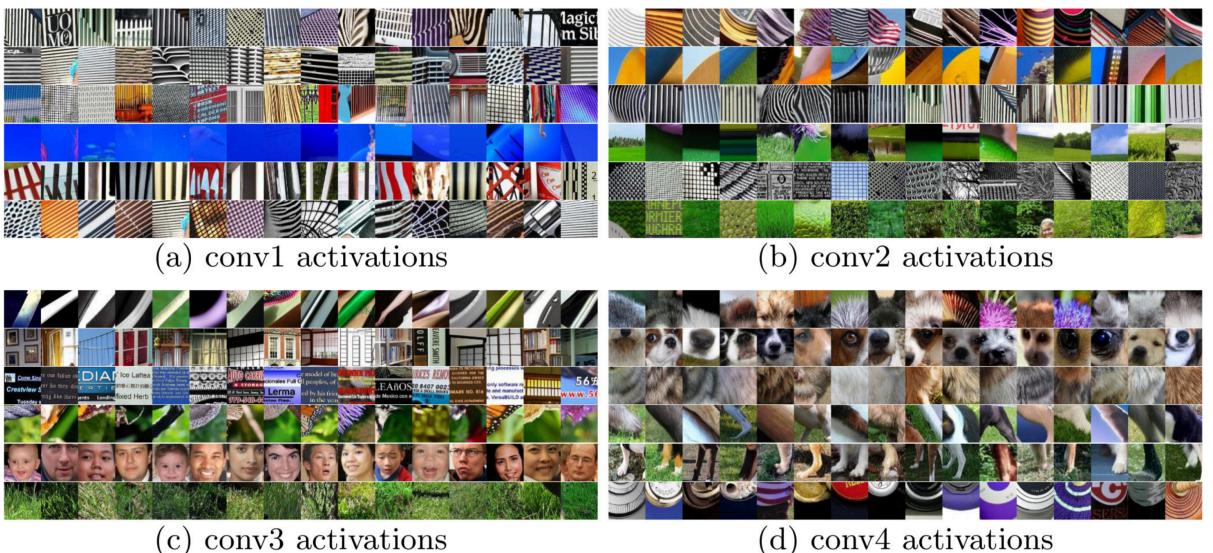
- 关于网络结构，他们采用 Siamese (暹罗网络，包含两个或多个结构完全相同、共享参数的结构，之前讲过) 结构，将 patch 按照打乱的序列输入网络，最后级联特征，并输入一个 MLP 中，经过 1000 通道的分类头输出
- Avoid Shortcuts
 - 和之前的论文一样，他们也需要避免网络走一些“捷径”，学习一些对于目标任务无关却能解决借口任务的特征
 - Low level statistics**
 - 一些相邻的 patch 可能包含一些相似的低级统计数据（例如均值、方差等）
 - 因此，我们对 patch 的均值和方差进行标准化
 - Edge continuity**

- 和之前的上下文预测任务一样，拼图游戏预测任务也存在边缘连续性的捷径问题
- 因此，从 85×85 的像素单元中随机选择 64×64 像素的补丁，并且也应用一些空间抖动

■ Chromatic aberration

- 同样的，也存在颜色偏差的问题
- 通过将图像映射为灰度图，或者在每个颜色通道中对几个像素进行空间上的抖动

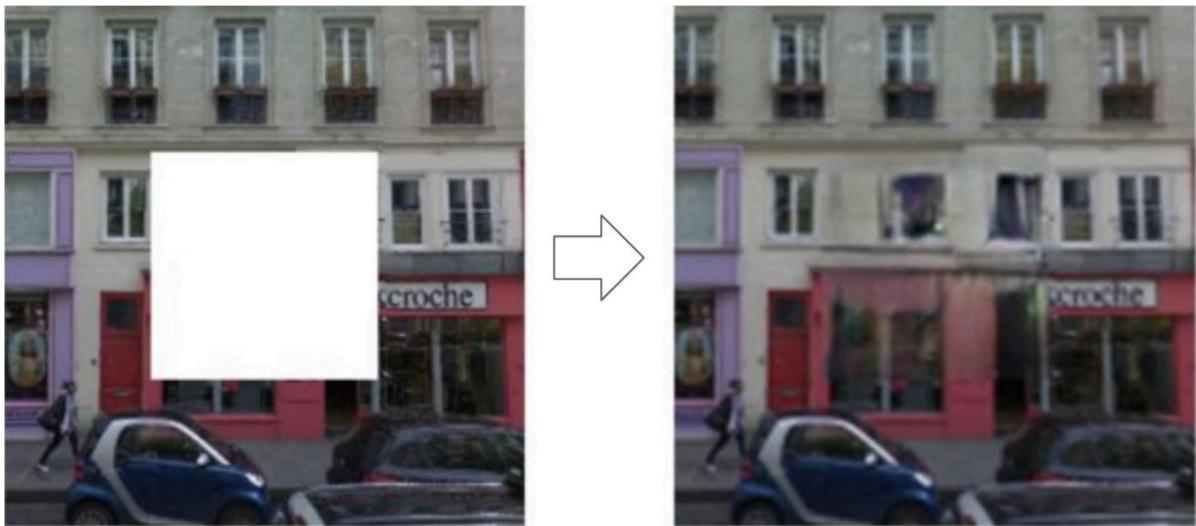
■ 如果对于深度学习中的捷径问题感兴趣，还可以查看这篇文献：[Geirhos et al.: Shortcut Learning in Deep Neural Networks. Arxiv, 2020.](#)



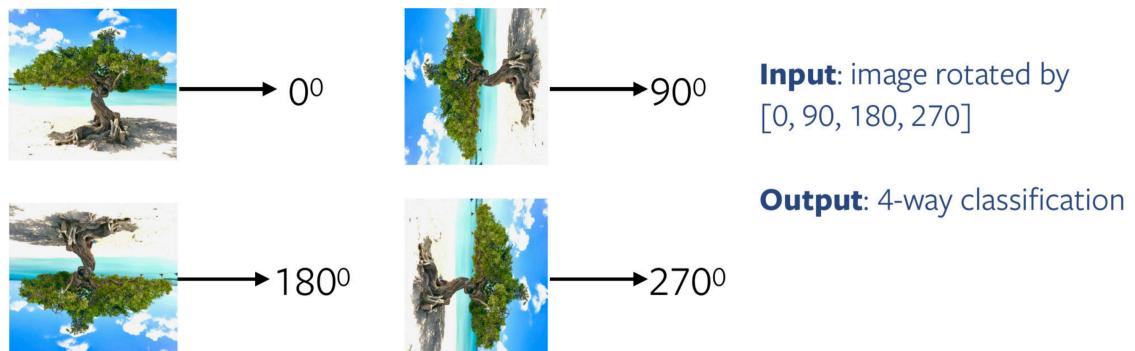
- 可以看到，在低层卷积层时，学习到的都是一些低级语义特征；到了高层卷积层后，确实就能够学习到一些富有语义信息的特征信息

Method	Pretraining time	Supervision	Classification	Detection	Segmentation
Krizhevsky et al. [25]	3 days	1000 class labels	78.2%	56.8%	48.0%
Wang and Gupta [39]	1 week	motion	58.4%	44.0%	-
Doersch et al. [10]	4 weeks	context	55.3%	46.6%	-
Pathak et al. [30]	14 hours	context	56.5%	44.5%	29.7%
Ours	2.5 days	context	67.6%	53.2%	37.6%

- 在此上游模型的基础上，他们微调特征用于 PASCAL VOC (一个计算机视觉挑战赛) 的分类、检测、分割任务
- 其中，第一行的方法是利用 ImageNet 进行监督学习的方法。可以看到，无监督学习方法在三个方面的表现结果几乎可以匹配监督学习的性能，并且有着近乎相同的训练时间，只是去掉了数据集标注的成本
- Feature Learning by Inpainting Pathak, Krishenbahl, Donahue, Darrell and Efros: Context Encoders: Feature Learning by Inpainting. CVPR, 2016.



- 这是一项有趣的任务，称为“图像修补”，有点类似于之前说到的 Denoising Autoencoder，不过相比起去除掉几个像素级的噪点，图像修补要求模型能够根据周围的图像信息补全一个 patch 区域的空白信息，这显然会困难得多



- ■ 同之前讲到的借口任务一样，它采用旋转的方式进行上游模型的构建——模型为了找出旋转的实际角度，需要深入学习其语义特征信息（本篇文章中实际使用的就是 4 个方向，并输出 4 通道的分类结果）

- Summary

- Prtextext Tasks
 - 借口任务关注于“视觉常识”，例如图像补全(inpainting)、图像上色(colorization)、图像重排(rearrangement)、预测旋转角度(predicting rotations)等
 - 模型被迫学习自然图像中的有用特征，例如对象类别的语义表示，图像中的上下文信息和局部结构
 - 我们不关心借口任务的表现结果，而是其学习到的特征是否具有足够的通用性，可以用于下游任务，例如分类、检测、分割等
- Problems

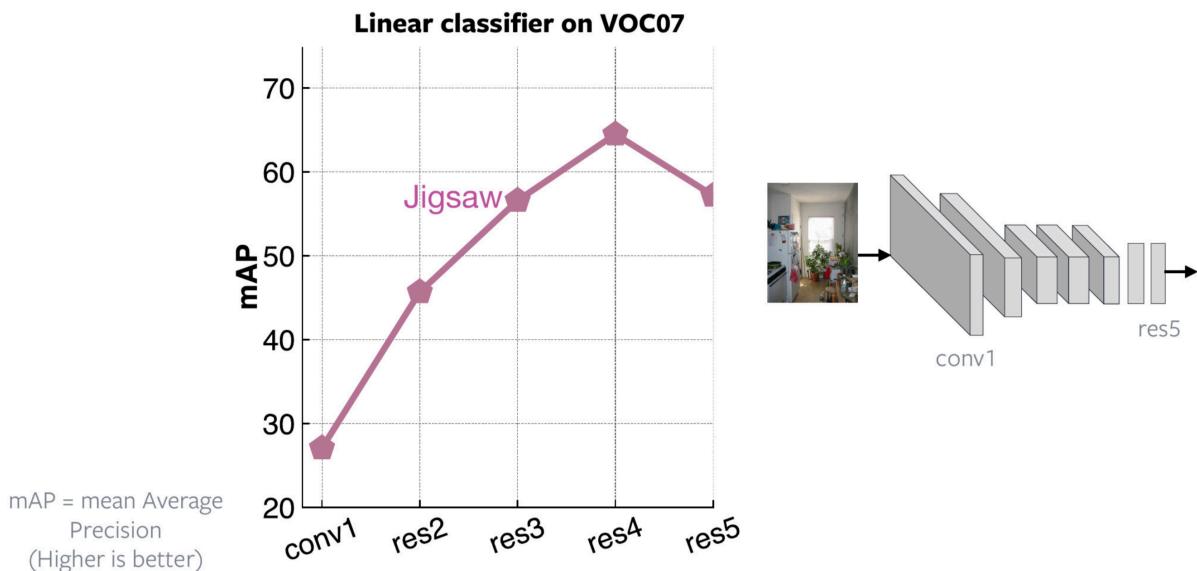
- 设计一个好的借口任务非常困难，甚至可以说是一门“艺术”；没有实在的理论支持，很多时候只能依靠直觉、经验和实验
- 学习到的特征可能缺乏通用性——某些借口任务学习到的特征可能仅针对特定类型的数据有效

11.4 Contrastive Learning

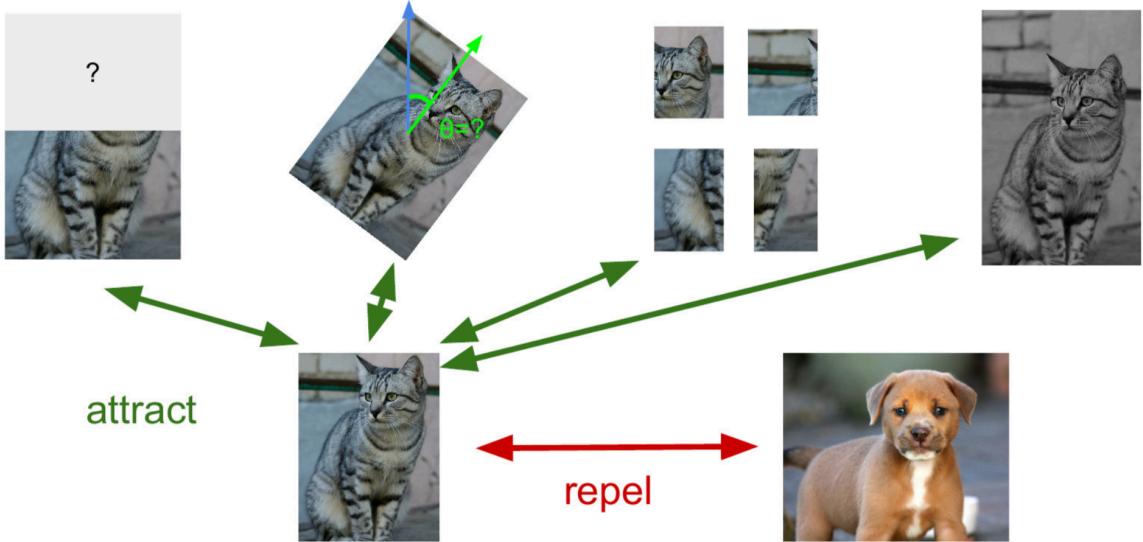
- Hope of Generalizaiton



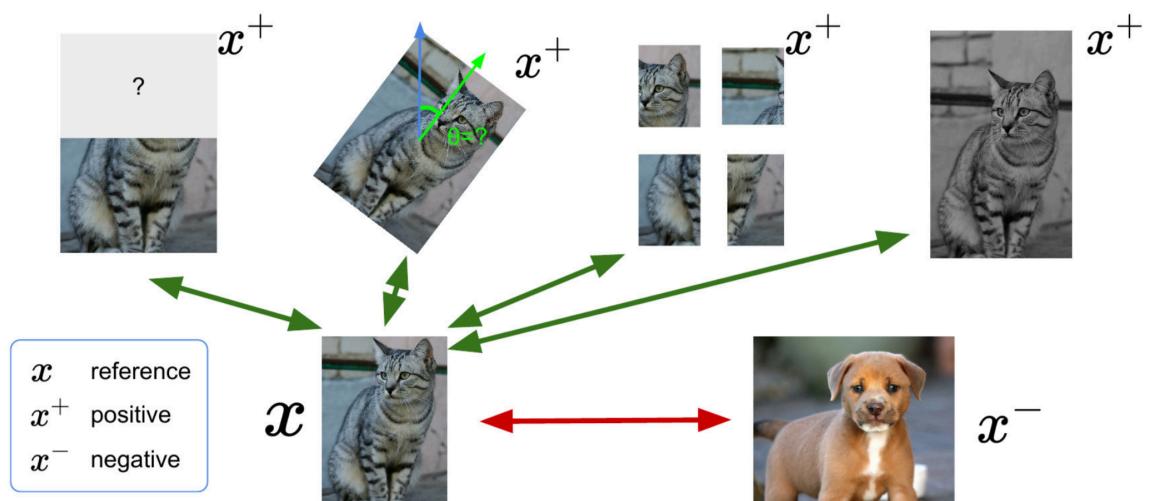
- 目前为止，我们设计上游任务和下游任务是完全独立开的——我们并不知道旋转图像的学习任务和图像分类的学习任务之间有什么关联
- 为了能够使得上游任务得到的模型为下游任务带来更大的收益，我们希望上游任务和下游任务之间能够有所关联，这就是对比学习的任务



- 我们尝试将“拼图游戏”上游模型的各个层接入分类头输出进行测试，结果会发现，在添加更多的层后，分类的性能甚至会下降——为什么呢？因为拼图游戏的任务与分类任务有很大的不同；最后一层会更专注于上下文预测、拼图游戏、旋转估计这些任务，但它们不包含解决图像分类任务所需的语义信息，因此性能下降
- 我们能否找到一种更具通用性的借口任务？



- 我们希望预训练的特征
- 能够适应图像分类任务（例如图中的猫，无论是部分化、旋转后、乱序，都应该分类为一个特征空间）
- 也应该对于一些干扰因素具有不变性（例如位置、光线、颜色等）
- 如图中的猫，从一个应用图像可以利用图像增强生成出很多用于表示应该位于同一特征空间的变换过的图像，这些图像在对比学习中被称为“views”



- 根据上图可以看到，对于一个参考图像 x ，我们可以将其生成出的所有应该位于同一个特征空间的“正面样本” x^+ ，以及不应该位于同一个特征空间的“负面样本” x^-
- 定义一个评分函数 $s(\cdot, \cdot)$ ，我们想要训练一个编码器 f ，使得它在接收到 (x, x^+) 时能产生一个高分，接收到 (x, x^-) 时产生一个低分

$$s(f(x), f(x^+)) \gg s(f(x), f(x^-))$$

- Contrastive Learning——InfoNCE Loss [Oord, Li and Vinyals: Representation Learning with Contrastive Predictive Coding, Arxiv, 2018.](#)

- 假设我们有 1 个参考样本 x , 有 1 个正样本 x^+ , $N-1$ 个负样本 x_j^- , 根据多类交叉熵损失函数, 我们可以得到下式 \$\$

$$\mathcal{L} = -\mathbb{E} \left[\log \frac{\exp(s(f(x), f(x^+)))}{\sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))} \right]$$

- 数学上讲, 我们可以将 s 理解为两个样本的相似度分数, 分子表示正样本的相

- 其中 $\log(N)$ 表示负样本数量对互信息的贡献

- 换句话说, 最小化 InfoNCE 损失值等价于最大化 $f(x)$ 和 $f(x^+)$ 的互信息
- 关键思想:
 - 通过**最大化**不同 views 之间的互信息, 从而迫使模型学习到高层次的语义信息 (也即, 我们成功将问题转换为了一个优化问题)
 - 负样本数量 $N-1$ 越大, 则互信息下界 $\log(N) - \mathcal{L}$ 越紧密 (也即越能够准确地估计互信息)

互信息(Mutual Information, MI)

互信息衡量了两个变量之间的依赖程度, 可以理解为关联程度, 也即我们希望参考样本 x 和正样本 x^+ 的互信息尽可能大, 和负样本 x^- 的互信息尽可能小