# 无监督视觉表示学习的动量对比

开明何浩琪范雨欣吴赛宁谢罗斯Girshick

Facebook人工智能研究（公平）

代码：https : //github.com/facebookresearch/moco

## 抽象的

*我们提出动量对比 (MoCo) 用于非超*
*视觉视觉表征学习。从一个角度来看*
*对比学习 [ 29 ] 作为字典查找，我们构建*
*带有队列和移动平均的动态字典*
*编码器。这使得构建一个大型且一致的 dic-*
*动态的促进对比无监督*
*学习。MoCo 提供具有竞争力的结果*
*ImageNet 分类的通用线性协议。更多的*
*重要的是，MoCo 迁移学习到的表征*
*很好地完成下游任务。MoCo 可以超越其超级*
*在7 个检测/分割中看到了预训练对应物*
*PASCAL VOC、COCO 和其他数据集上的任务，一些-*
*次大幅度超过它。这表明*
*无监督和有监督表示之间的差距*
*在许多视觉任务中，学习已经基本上是封闭的。*

## 一、简介

无监督表示学习非常成功——
在自然语言处理中充分，*例如*，如 GPT 所示
[ 50 , 51 ] 和 BERT [ 12 ]。但是有监督的预训练仍然是
在计算机视觉中占主导地位，其中无监督方法
ods 普遍落后。原因可能来自不同的
在它们各自的信号空间中的引用。语言任务
具有离散的信号空间（词、子词单元等）
用于构建无监督的标记化*词典*
学习可以立足。相比之下，计算机视觉进一步
涉及字典构建 [ 54 , 9 , 5 ]，因为原始信号是
在一个连续的高维空间中，并且不是结构化的
适合人类交流（*例如*，与文字不同）。

最近的几项研究 [ 61 , 46 , 36 , 66 , 35 , 56 , 2 ] 提出
无监督视觉表示的有希望的结果
使用与*对比损失*相关的方法学习
[ 29 ]。尽管受到各种动机的驱动，这些方法
可以被认为是构建*动态词典*。这
字典中的“键”（令牌）是从数据中采样的
（*例如*，图像或补丁）并由编码器表示
网络。无监督学习训练编码器执行
字典查找：编码的“查询”应该是相似的

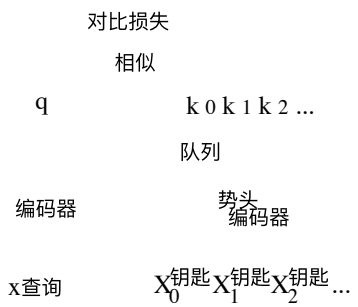



图 1. Momentum Contrast (MoCo) 训练视觉表现
通过将编码的查询 q 与字典匹配来实现编码器
使用对比损失的编码密钥。字典键
{k 0 , k 1 , k 2 , ...} 由一组数据样本动态定义。
字典被构建为一个队列，当前的小批量en-
排队，最旧的小批量出队，将其与
小批量大小。密钥由缓慢进展的
编码器，由查询编码器的动量更新驱动。
这种方法可以使用大而一致的字典进行学习
视觉表示。

从这个角度来看，我们假设它是可取的
构建具有以下特征的词典：(i) 大*和*(ii) 一致
随着他们在训练过程中的发展。直观地说，一个更大的字典——
nary 可能会更好地对底层连续、高
维视觉空间，而字典中的键
应由相同或相似的编码器表示，以便
它们与查询的比较是一致的。然而，前
使用对比损失的 isting 方法可以限制在
这两个方面之一（稍后在上下文中讨论）。

我们提出*动量对比*(MoCo) 作为一种方式
为无监督构建大型且一致的词典
使用对比损失学习（图1）。我们维持
字典作为数据样本*队列*：编码表示
当前小批量的消息被排队，并且
最老的出列。队列解耦字典
小批量大小的大小，使其变大。更多的-
结束，因为字典键来自前面的 sev-
eral mini-batches，一个*缓慢进展的*关键编码器，实现

到它的匹配键并且与其他键不同。学习是
公式化为最小化对比损失 [ 29 ]。

被视为查询的基于动量的移动平均值
编码器，建议保持一致性。

1 9729

## 第2页

MoCo 是一种构建动态词典的机制
用于对比学习，可用于各种
借口任务。在本文中，我们遵循一个简单的实例
区分任务[ 61，63，2 ]：查询，如果密钥匹配
它们是相同的编码视图（*例如*，不同的作物）
图像。使用这个借口任务，MoCo 表现出竞争力
线性分类通用协议下的结果
在 ImageNet 数据集 [ 11 ] 中。

无监督学习的一个主要目的是预训练
可以转移到的表示（*即*特征）
通过微调下游任务。我们表明，在 7 下-
与检测或分割相关的流任务，MoCo
无监督预训练可以*超越*其 ImageNet 超
可见的对应物，在某些情况下是非常重要的。在
在这些实验中，我们探索了在 Ima-
geNet 或在*10 亿张*Instagram 图像集上，演示
相信 MoCo 可以在更真实的世界中运作良好，十亿
图像比例和相对未经策划的场景。这些重新
结果表明，MoCo 在很大程度上缩小了非
监督和监督表示学习在许多
计算机视觉任务，并且可以作为 Im-的替代方案
ageNet 在多个应用中监督预训练。

## 2. 相关工作

无监督/自监督1 种学习方法生成
盟友涉及两个方面：借口任务和损失函数。
术语“借口”意味着正在解决的任务不是
真正感兴趣，但只为真正的目的而解决
学习一个好的数据表示。损失函数可以
通常独立于借口任务进行调查。交通部
专注于损失函数方面。接下来我们讨论相关
这两个方面的研究。

损失函数。定义损失函数的常用方法
是衡量模型预测之间的差异
和*固定*目标，例如重建输入像素
（*例如*，自动编码器）通过 L1 或 L2 损失，或对
输入到预定义的类别（*例如*，八个位置 [ 13 ]，
color bins [ 64 ]) 通过交叉熵或基于边际的损失。
如下所述，其他替代方案也是可能的。

对比损失 [ 29 ] 衡量 sam-
表示空间中的 ple 对。而不是匹配一个
输入到固定目标，在对比损失公式中
目标可以*改变*上即时训练期间和可以被定义
就网络计算的数据表示而言
[ 29 ]。对比学习是最近几项研究的核心
适用于无监督学习[ 61，46，36，66，35，56，2 ]，
我们稍后会在上下文中详细说明（第3.1节）。

对抗性损失 [ 24 ] 衡量的是
概率分布。这是一项广泛成功的技术

1自监督学习是无监督学习的一种形式。他们的不
在现有文献中，这种说法是非正式的。在本文中，我们使用更多
“无监督学习”的经典术语，在“无监督学习”的意义上
通过人工标注的标签”。

用于无监督数据生成。对抗性方法
表征学习在 [ 15 , 16 ]中进行了探索。有
生成对抗网络之间的关系（见 [ 24 ]）
和噪声对比估计 (NCE) [ 28 ]。

借口任务。广泛的借口任务已被支持
构成。示例包括在某些情况下恢复输入
损坏，*例如*，去噪自动编码器 [ 58 ]，上下文自动
编码器 [ 48 ]，或跨通道自动编码器（着色
重）[ 64 , 65 ]。一些借口任务形成伪标签，
*例如*，单个（“示例”）图像的变换 [ 17 ]，
补丁排序 [ 13 , 45 ]、跟踪 [ 59 ] 或分割观察
对象 [ 47 ] 在视频中，或聚类特征 [ 3 , 4 ]。

对比学习与借口任务。各种借口
任务可以基于某种形式的对比损失函数
。实例判别法[ 61 ]相关
到基于范例的任务 [ 17 ] 和 NCE [ 28 ]。借口
对比预测编码 (CPC) [ 46 ] 中的任务是一种形式
上下文自动编码 [ 48 ]，以及对比多视图
编码 (CMC) [ 56 ] 它与着色 [ 64 ] 有关。

## 3. 方法

### 3.1. 作为字典查找的对比学习

对比学习 [ 29 ] 及其最近的发展，
可以被认为是训练*字典*的编码器
*查找*任务，如下所述。

考虑一个编码的查询 q 和一组编码的 sam-
ples {k 0 ,k 1 ,k 2 , ...} 是字典的键。作为-
假设在 dic- 中有一个单一的键（表示为 k +）
q 匹配的句子。对比损失 [ 29 ] 是一个函数
当 q 与其正键 k +相似时，其值较低
并且与所有其他键不同（被认为是负键
对于 q)。用点积来衡量相似性，一种形式
一个对比损失函数，称为 InfoNCE [ 46 ]，被认为是
在本文中：

$$L_q = -\log \frac{\exp(q\cdot k_+/\tau)}{\sum_{i=0}^{K}\exp(q\cdot k_i/\tau)} \quad (1)$$

其中 τ 是根据 [ 61 ]的温度超参数。总和
超过 1 个正样本和 K 个负样本。直觉上，
此损失是 (K+1) 路基于 softmax 的类的对数损失
sifier 试图将 q 分类为 k +。对比损失函数
也可以根据其他形式的[ 29，59，61，36 ]，如
基于保证金的损失和 NCE 损失的变体。

对比损失作为一个无监督的目标
用于训练编码器网络的函数
查询和键 [ 29 ]。一般来说，查询表示
is q = f q (x q ) 其中 f q是一个编码器网络，x q是一个
查询样本（同样，k = f k (x k )）。它们的实例化
取决于具体的借口任务。输入 x q和 x k可以
是图像[ 29，61，63 ]，贴剂[ 46 ]，或上下文由
一组补丁 [ 46 ]。网络 f q和 f k可以相同
[ 29，59，63 ]，部分共享的[ 46，36，2 ]，或不同的[ 56 ]。

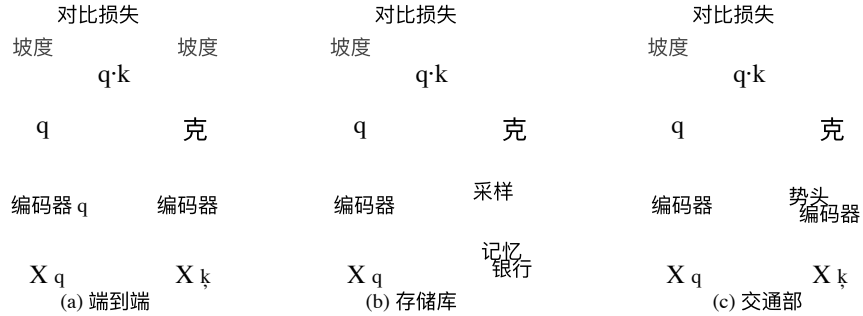



图 2. 三种对比损失机制的概念比较（经验比较在图3和表3 中）。在这里，我们
说明一对查询和键。这三种机制的不同之处在于密钥的维护方式和密钥编码器的更新方式。
(a)：用于计算查询和键表示的编码器通过反向传播端到端更新（两个编码器可以
不同）。(b)：密钥表示是从存储库中采样的 [ 61 ]。(c)：MoCo 对新密钥进行动态编码
动量更新编码器，并维护一个键队列（图中未显示）。

### 3.2. 动量对比

从以上角度看，对比学习是一种方式
在高维结构上构建离散字典
连续输入，如图像。字典是*动态*的
密钥是随机采样的，并且
关键编码器在训练期间进化。我们的假设是
好的特征可以通过一个大字典来学习
提供一组丰富的负样本，而编码器用于
字典键尽可能保持一*致*，尽管它
进化。基于这个动机，我们提出 Momentum
对比如下所述。

字典作为队列。我们方法的核心是
将字典维护为数据样本*队列*。这
允许我们重用来自即时预编码的密钥
放弃小批量。队列解耦的引入
来自小批量大小的字典大小。我们的字典
尺寸可以比典型的小批量尺寸大得多，并且
可以灵活独立地设置为超参数。

字典中的样本被逐步替换。
当前的 mini-batch 被加入到字典中，并且
队列中最旧的小批量被删除。口述——
nary 总是代表所有数据的采样子集，而
维护这本字典的额外计算是人工
老化。此外，可以删除最旧的小批量
有益，因为它的编码密钥是最过时的
因此与最新的最不一致。

动力更新。使用队列可以使dictio-
不大，但它也使得更新密钥变得棘手
通过反向传播编码器（梯度应该传播
到队列中的所有样本的门）。一个天真的解决方案是
从查询编码器 f q复制密钥编码器 f k，忽略-
这个梯度。但是这个解决方案在
实验（第4.1节）。我们假设这种失败
是由快速变化的编码器引起的
关键表示的一致性。我们提出了一个势头
更新以解决此问题。

形式上，将 f k的参数表示为 θ k和那些
f q作为 θ q，我们通过以下方式更新 θ k：

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q。 \quad (2)$$

这里 m ∈ [0, 1) 是一个动量系数。只有帕-
参数 θ q通过反向传播更新。那一刻——
Eqn.( 2 ) 中的tum update使得 θ k 的演化比
θ q。结果，虽然队列中的键被编码
通过不同的编码器（在不同的小批量中），差异
这些编码器之间的参考可以变小。在前-
periments，一个相对较大的动量（*例如*，m = 0.999，
我们的默认值）比较小的值（*例如*.，
m = 0.9），表明缓慢演化的密钥编码器是
使用队列的核心。

与之前机制的关系。MoCo 是个将军
使用对比损失的机制。我们将它与
图2中现有的两种通用机制。他们展出
字典大小和一致性的不同属性。

反向传播的端到端更新是一种自然的
机构（*例如*，[ 29，46，36，63，2，35 ]，图2一个）。它用
当前小批量中的样本作为字典，所以
键是一致编码的（由同一组编码器
参数）。但是字典大小与
小批量大小，受 GPU 内存大小限制。也是
受到大型小批量优化的挑战 [ 25 ]。一些重新
分的方法[ 46，36，2 ]基于由驱动由头任务
局部位置，字典大小可以变大
由多个职位。但这些借口任务可能需要
特殊的网络设计，例如修补输入 [ 46 ]
或自定义感受野大小 [ 2 ]，这可能会导致
将这些网络转移到下游任务。

另一种机制是存储库方法
通过[构成61 ]（图2的B）。一个存储库由
数据集中所有样本的表示。词典
对于每个小批量从内存中随机采样
没有反向传播的银行，因此它可以支持大量
字典大小。但是，样本的表示

## 第 4 页

算法 1 类似 PyTorch 风格的 MoCo 伪代码。

```
# f_q, f_k: 用于查询和密钥的编码器网络
# queue: 字典作为 K 个键的队列 (CxK)
# m：动量
# t：温度

f_k.params = f_q.params # 初始化
for x in loader: # 加载一个带有 N 个样本的 minibatch x
    x_q = aug(x) # 随机增加的版本
    x_k = aug(x) # 另一个随机增强的版本

    q = f_q.forward(x_q) # 查询：NxC
    k = f_k.forward(x_k) # 键：NxC
    k = k.detach() # 键没有渐变

    # 正对数：Nx1
    l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

    # 负对数：NxK
    l_neg = mm(q.view(N,C), queue.view(C,K))

    # logits：Nx(1+K)
    logits = cat([l_pos, l_neg], dim=1)

    # 对比损失，方程（1）
    标签 = zeros(N) # 正数是第 0 个
    损失 = CrossEntropyLoss(logits/t, 标签)

    # SGD更新：查询网络
    损失.向后（）
    更新（f_q.params）

    # 动量更新：关键网络
    f_k.params = m*f_k.params+(1-m)*f_q.params

    # 更新字典
    enqueue(queue, k) # 将当前 minibatch 入队
    dequeue(queue) # 使最早的 minibatch 出队
```

bmm：批量矩阵乘法；mm：矩阵乘法；猫：串联。

记忆库在最后一次被看到时被更新了，所以
采样键本质上是关于多个编码器的
整个过去时代的不同步骤，因此不太受控制
坚持。内存采用动量更新
存入 [ 61 ]。它的动量更新是在代表
同一个样本，*而不是*编码器。这势头
更新与我们的方法无关，因为 MoCo 不
跟踪每个样本。此外，我们的方法更
内存效率高，可以在十亿级数据上进行训练，
这对于存储库来说可能是棘手的。

秒。图 4 根据经验比较了这三种机制。

### 3.3. 借口任务

对比学习可以推动各种借口任务。
由于本文的重点不是设计一个新的借口
任务，我们使用一个简单的主要跟随*实例*
[ 61 ]中的*歧视*任务，最近的一些工作[ 63，
2 ] 相关。

在[ 61 ]之后，我们将一个查询和一个键视为一个可能
如果它们来自相同的图像，则为 itive 对，而其他 -
明智地作为负样本对。根据 [ 63 , 2 ]，我们取
随机数据下同一图像的两个随机“视图”
增强以形成正对。查询和键
分别由它们的编码器 f q和 f k编码。这
编码器可以是任何卷积神经网络 [ 39 ]。

算法1为此提供了 MoCo 的伪代码
借口任务。对于当前的小批量，我们编码
查询及其相应的键，它们构成了位置
样本对。负样本来自队列。

技术细节。我们采用 ResNet [ 33 ] 作为编码器，
其最后一个全连接层（在全局平均池之后）
ing) 具有固定维度的输出 (128-D [ 61 ])。这出——
put 向量通过其 L2 范数进行归一化 [ 61 ]。这是
查询或键的表示。温度 τ in
方程( 1 )被设置为0.07[ 61 ]。数据增强设置
遵循 [ 61 ]：从运行中获取 224×224 像素的裁剪
domly 调整大小的图像，然后进行随机颜色 jit-
tering、随机水平翻转和随机灰度控制
版本，都可以在 PyTorch 的 torchvision 包中找到。

洗牌 BN。我们的编码器 f q和 f k都有 Batch
标准化 (BN) [ 37 ] 与标准 ResNet [ 33 ] 中的一样。在
实验，我们发现使用 BN 可以防止模型
从学习良好的表征，如类似报道
在 [ 35 ]（避免使用 BN）。该模型似乎
“欺骗”借口任务并轻松找到低损失解决方案
重刑。这可能是因为批内通信
样本之间的偏差（由 BN 引起）泄漏信息。

我们通过改组 BN 来解决这个问题。我们训练
多个 GPU 并在样本上独立执行 BN
为每个 GPU 减少（如通常做法）。为了
关键编码器 f k，我们将当前的样本顺序打乱
在将其分配到 GPU 之前进行小批量处理（并洗牌
编码后返回）；小批量的样品顺序
对于查询编码器 f q没有改变。这确保了
用于计算查询及其正键的批处理统计信息
来自两个不同的子集。这有效地解决了
作弊问题，并允许培训从 BN 中受益。

我们在我们的方法和它的端到端中都使用了 shuffled BN
结束消融配对（图2的A）。它与
存储库对应物（图2 b），它不满足
来自这个问题，因为正键来自不同的
过去小批量生产。

## 4. 实验

我们研究在以下方面进行的无监督训练：

ImageNet-1M (IN-1M)：这是 ImageNet [ 11 ]训练-
荷兰国际集团镶有～ 128万倍的图像在1000级（常
称为 ImageNet-1K；我们改为计算图像编号，
因为类没有被无监督学习利用）。这
数据集在其类别分布方面非常平衡，并且它的
年龄通常包含对象的标志性视图。

Instagram-1B (IG-1B)：按照[ 44 ]，这是一个数据集
的～*1十亿*（940M）的公众形象*的Instagram*。这
图像来自～ 1500 个标签 [ 44 ]，这些标签与
ImageNet 类别。此数据集相对*未经整理*
与 IN-1M 相比，具有*长尾*、*不平衡*
真实世界数据的分布。该数据集包含
标志性对象和场景级图像。

## 第 5 页

训练。我们使用 SGD 作为我们的优化器。新元权重
衰减为 0.0001，新元动量为 0.9。对于 IN-1M，
我们在 8 中使用 256 的小批量（算法1中的N ）
GPU，初始学习率为 0.03。我们训练 200
学习率在 120 时乘以 0.1 和
160 个 epochs [ 61 ]，训练 ResNet-50需要大约53 小时。为了
IG-1B，我们在 64 个 GPU 中使用了 1024 的小批量，并且
0.12 的学习率呈指数衰减
每 62.5k 次迭代后 0.9 倍（64M 图像）。我们训练
为1.25M迭代（～ IG-1B的1.4时期），服用～ 6天
对于 ResNet-50。

### 4.1. 线性分类协议

我们首先通过*线性*分类验证我们的方法
*冻结*功能，遵循通用协议。在这个分
部分我们在 IN-1M 上执行无监督的预训练。
然后我们冻结特征并训练一个有监督的线性
分类器（一个全连接层，后跟 softmax）。我们
在全局平均池化特征上训练这个分类器
一个 ResNet，100 个 epoch。我们报告 1-crop，top-1 分类
ImageNet 验证集上的阳离子准确度。

对于这个分类器，我们执行网格搜索并找到
最佳初始学习率为 30，权重衰减为 0
（[ 56 ] 中也有类似的报道）。这些超参数每
形成一致的所有消融条目
本节。这些超参数值意味着
特征分布（*例如*，幅度）可以是实质性的
与 ImageNet 监督训练明显不同，
我们将在第二节重新讨论一个问题。4.2 .

消融：对比损失机制。我们比较
图2所示的三种机制。集中
关于对比损失机制的影响，我们实施
它们都在 Sec 中描述的同一个借口任务中。3.3 .
我们也使用相同形式的 InfoNCE 作为对比
损失函数，方程（1）。因此，比较仅针对
三大机制。

结果在图3中。总的来说，这三款机
nisms 受益于更大的 K。类似的趋势是
在 [ 61 , 56 ] 中观察到的记忆库机制下，
而在这里我们表明这种趋势更普遍，可以
在所有机制中都可以看到。这些结果支持我们的动机
建立一个大词典。

端到端机制与 MoCo 类似
当 K 小时。但是，字典大小有限
由于端到端的要求，由小批量大小决定。
这里最大的小批量一台高端机器（8 Volta
32GB GPUs) 是 1024。更本质上，大
小批量训练是一个开放的问题 [ 25 ]：我们找到了
必须使用线性学习率缩放规则 [ 25 ]
在这里，没有它，准确度会下降（大约2%
1024 小批量）。但是使用更大的小批量进行优化
更难 [ 25 ]，趋势是否可以
即使内存足够，也可以外推到更大的 K。

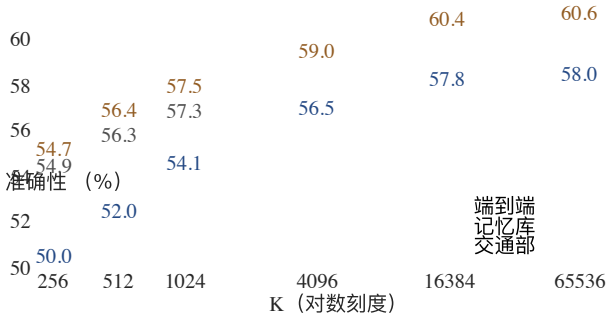



图 3. 三种对比损失机制的比较
der ImageNet 线性分类协议。我们采用相同的
借口任务（第3.3节）并且只改变对比损失机制
nism（图2）。存储库中的负数为 K
和 MoCo，并且在端到端中是 K-1（偏移 1，因为
正键在同一个小批量中）。网络是 ResNet-50。

存储库 [ 61 ] 机制可以支持更大的
字典大小。但它比 MoCo 差 2.6%。这是
符合我们的假设：存储库中的密钥
来自过去时代非常不同的编码器，并且
它们不一致。注意存储库结果
58.0% 反映了我们对 [ 61 ] 的改进实施。2

消融：动量。下表显示了 ResNet-50
不同 MoCo 动量值的精度（m in
Eqn.( 2 )) 用于预训练（此处为 K = 4096）：

| 动量 m | 0 | 0.9 | 0.99 | 0.999 | 0.9999 |
|---|---|---|---|---|---|
| 准确性（%） | *失败* | 55.2 | 57.8 | 59.0 | 58.9 |

当 m 在 0.99 ~ 0.9999 时，它表现得相当好，
表明缓慢进展（*即*，相对较大的mo-
精神）键编码器是有益的。当 m 太小时
（*例如*., 0.9），准确度显着下降；在极端
的*没有动量*（m是0），训练损失振荡并
无法收敛。这些结果支持我们的动机
建立一个一致的字典。

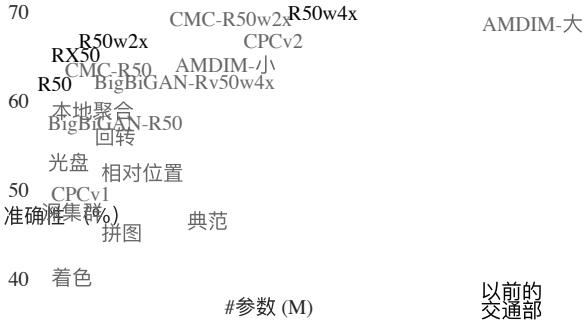
与之前的结果进行比较。以前的超级
视觉学习方法可以在模型上有很大不同
尺寸。为了公平和全面的比较，我们报告
准确性与#parameters 3权衡。除了 ResNet-50
(R50) [ 33 ]，我们还报告了它的 2× 和 4× 变体
更宽（更多频道），遵循 [ 38 ]。4我们设置 K = 65536
并且 m = 0.999。表1是比较。

具有 R50 的 MoCo 具有竞争力并实现了
60.6%的准确率，优于同类同类产品的所有竞争对手
模型大小（～ 24M）。MoCo 受益于更大的模型和
使用 R50w4× 达到 68.6% 的准确率。

值得注意的是，我们使用*标准*取得了有竞争力的结果
ResNet-50 并且不需要特定的架构设计，*例如*，

2这里 58.0% 是 InfoNCE，K=65536。再现54.3%
当使用 NCE 且 K=4096（与 [ 61 ]相同）时，接近 [ 61 ] 中的54.0% 。
3 个参数是*特征提取器的*参数：*例如*，我们不计算 pa-
的rameters *CONV* X如果*CONV* X不包括在线性分类。
4我们的 w2× 和 w4× 模型对应于“×8”和“×16”的情况
在[ 38 ]，这是因为标准尺寸RESNET在[称为“×4” 38 ]。

**第 6 页**




| 方法 | 建筑学 | #params (M) | 准确率 (%) |
|---|---|---|---|
| 范例 [ 17 ] | R50w3× | 211 | 46.0 [ 38 ] |
| 相对位置 [ 13 ] | R50w2× | 94 | 51.4 [ 38 ] |
| 拼图 [ 45 ] | R50w2× | 94 | 44.6 [ 38 ] |
| 旋转 [ 19 ] | RV50w4× | 86 | 55.4 [ 38 ] |
| 着色 [ 64 ] | R101 * | 28 | 39.6 [ 14 ] |
| 深集群 [ 3 ] | VGG [ 53 ] | 15 | 48.4 [ 4 ] |
| BigBiGAN [ 16 ] | R50 | 24 | 56.6 |
| | RV50w4× | 86 | 61.3 |
| *基于对比学习的方法如下：* | | | |
| InstDisc [ 61 ] | R50 | 24 | 54.0 |
| 本地聚合 [ 66 ] | R50 | 24 | 58.8 |
| 每次点击费用 v1 [ 46 ] | R101 * | 28 | 48.7 |
| 每次点击费用 v2 [ 35 ] | R170 * | 303 | 65.9 |
| 中央军委 [ 56 ] | R50 L+ab | 47 | 64.1 † |
| | R50w2× L+ab | 188 | 68.4 † |
| AMDIM [ 2 ] | AMDIM小 | 194 | 63.5 † |
| | AMDIM大 | 626 | 68.1 † |
| 交通部 | R50 | 24 | 60.6 |
| | RX50 | 46 | 63.9 |
| | R50w2× | 94 | 65.4 |
| | R50w4× | 375 | 68.6 |

表 1. 线性分类协议下的比较
在 ImageNet 上。该图将表格可视化。全部报告为
在 ImageNet-1M 训练集上进行无监督预训练，如下
通过在冷冻特征上训练的监督线性分类来降低
tures，在验证集上评估。参数计数是
那些特征提取器。我们与改进的重新比较
实现（如果可用）（在数字后引用）。
符号：R101 ∗ /R170 ∗是 ResNet-101/170 和最后一个残差阶段
移除 [ 14 , 46 , 35 ]，R170 变宽 [ 35 ]；Rv50 是可逆的
net [ 23 ]，RX50 是 ResNeXt-50-32×8d [ 62 ]。
†： *预训练使用由 ImageNet 标签监督的 FastAutoAugment [ 40 ]。*

修补输入 [ 46 , 35 ]，精心定制的感受野
[ 2 ]，或结合两个网络[ 56 ]。通过使用架构
不是为借口任务定制的，更容易
将特征转移到各种视觉任务中
型坯，在下一小节中研究。

本文的重点是一般控制的机制。
传统学习；我们不探索正交因素（例如
作为特定的借口任务），这可能会进一步提高准确性。
例如，“MoCo v2”[ 8 ]，一个预
这份手稿的 inary 版本，达到 71.1% 的准确率
与 R50（从 60.6% 上升），考虑到数据的微小变化
增强和输出投影头 [ 7 ]。我们相信
这个额外的结果显示了一般性和鲁棒性——
MoCo 框架的重要性。

| 预训练 | 美联社50 | 美联社 | 美联社75 |
|---|---|---|---|
| 随机初始化 | 64.4 | 37.9 | 38.6 |
| 极好的。IN-1M | 81.4 | 54.0 | 59.1 |
| 莫科IN-1M | 81.1 ( - 0.3) | 54.6 ( + 0.6) | 59.9 ( + 0.8) |
| MoCo IG-1B | 81.6 ( + 0.2) | 55.5 ( + 1.5) | 61.2 ( + 2.1) |

(a) 更快的 R-CNN，R50-dilated-C5

| 预训练 | 美联社50 | 美联社 | 美联社75 |
|---|---|---|---|
| 随机初始化 | 60.2 | 33.8 | 33.1 |
| 极好的。IN-1M | 81.3 | 53.5 | 58.8 |
| 莫科IN-1M | 81.5 ( + 0.2) | 55.9 ( + 2.4) | 62.6 ( + 3.8) |
| MoCo IG-1B | 82.2 ( + 0.9) | 57.2 ( + 3.7) | 63.7 ( + 4.9) |

(b) 更快的 R-CNN，R50-C4

表 2. 在 PASCAL VOC 上微调的目标检测
trainval07+12。评估在 test2007 上：AP 50（默认
VOC 指标）、AP（COCO 风格）和 AP 75平均超过 5 次试验。
所有被微调为24K迭代（∼ 23个时期）。在括号中
是与 ImageNet 监督预训练对应物的差距。
绿色是至少+0.5点的差距。

| | R50-扩张-C5 | | | R50-C4 | | |
|---|---|---|---|---|---|---|
| 预训练 | 美联社50 | 美联社 | 美联社75 | 美联社50 | 美联社 | 美联社75 |
| 端到端 | 79.2 | 52.0 | 56.6 | 80.4 | 54.6 | 60.3 |
| 记忆库 | 79.8 | 52.9 | 57.9 | 80.6 | 54.9 | 60.6 |
| 交通部 | 81.1 | 54.6 | 59.9 | 81.5 | 55.9 | 62.6 |

表 3. 三种对比损失机制的比较
PASCAL VOC物体检测，在trainval07+12上微调
并在 test2007 上进行评估（平均超过 5 次试验）。所有型号
由我们实现（图3），在 IN-1M 上进行预训练，并且
使用与表2 中相同的设置进行调整。

### 4.2. 传输功能

无监督学习的一个主要目标是*学习特征
*是可转让的*。ImageNet 监督预训练是
最有影响力的初始化
调谐在下游任务（*例如*，[ 21，20，43，52 ]）。下一个
我们将 MoCo 与 ImageNet 监督预训练进行比较，
转移到各种任务，包括 PASCAL VOC [ 18 ]，
可可[ 42 ]等。作为先决条件，我们讨论两个重要的
涉及的问题 [ 31 ]：规范化和时间表。

正常化。如第 2 节所述。4.1、特征产生
无监督预训练可以有不同的分布
与 ImageNet 监督预训练相比。但是一个
用于下游任务的系统通常具有超参数
（*例如*，学习率）选择用于监督预训练。到
缓解这个问题，我们在期间采用*特征归一化*
微调：我们用经过训练的 BN（和同步
跨 GPU 同步 [ 49 ]），而不是通过
仿射层 [ 33 ]。我们也在新初始化的时候使用了BN
层（*例如*，FPN [ 41 ]），这有助于校准幅度。

我们在微调监督时执行归一化
和无监督的预训练模型。MoCo*使用相同的
超参数*作为 ImageNet 监督对应物。

时间表。如果微调时间够长，
从随机初始化训练检测器可以很强大
基线，并且可以匹配 ImageNet 监督计数器
部分关于可可 [ 31 ]。我们的目标是研究可*转移性*

| 预训练 | 美联社50 RelPos，由 [ 14 ] | 多任务 [ 14 ] | Jigsaw，由 [ 26 ] | LocalAgg [ 66 ] | 交通部 | 美联社 交通部 | 多任务 [ 14 ] | 美联社75 交通部 |
|---|---|---|---|---|---|---|---|---|
| 极好的。IN-1M | 74.2 | 74.2 | 70.5 | 74.6 | 74.4 | 42.4 | 44.3 | 42.7 |
| 取消。IN-1M | 66.8 ( - 7.4) | 70.5 ( - 3.7) | 61.4 ( – 9.1) | 69.1 ( - 5.5) | 74.9 ( + 0.5) | 46.6 ( + 4.2) | 43.9 ( – 0.4) | 50.1 ( + 7.4) |
| 取消。IN-14M | — — | — — | 69.2 ( – 1.3) | — — | 75.2 ( + 0.8) | 46.9 ( + 4.5) | — — | 50.2 ( + 7.5) |
| 取消。YFCC-100M | — — | — — | 66.6 ( – 3.9) | — — | 74.7 ( + 0.3) | 45.9 ( + 3.5) | — — | 49.0 ( + 6.3) |
| 取消。IG-1B | — — | — — | — — | — — | 75.6 ( + 1.2) | 47.6 ( + 5.2) | — — | 51.7 ( + 9.0) |

表 4. 与之前在 PASCAL VOC trainval2007 上微调的对象检测方法的比较。评估已开启
测试2007。所述ImageNet监督同行是从相应的文件，并报告为具有*相同的结构*作为
各自的无监督预训练对应物。所有条目都基于 C4 主干。[ 14 ]中的模型是R101 v2 [ 34 ]，并且
其他的是R50。RelPos（相对位置）[ 13 ] 结果是多任务论文 [ 14 ] 中最好的单任务案例。Jigsaw [ 45 ] 结果是
来自 [ 26 ] 中基于 ResNet 的实现。我们的结果是 9k 迭代微调，平均超过 5 次试验。括号里是
与 ImageNet 监督预训练对应物的差距。绿色是至少+0.5点的差距。

*两者均*的特点，所以我们的实验是在控制sched-
ules，*例如*，1×（~ 12 epochs）或 2× 调度 [ 22 ] 用于
COCO，与[ 31 ] 中的6× ~ 9×形成对比。在较小的数据集上
像 VOC 一样，训练时间更长可能赶不上 [ 31 ]。

尽管如此，在我们的微调中，MoCo*使用相同的*
作为 ImageNet 监督对应的*调度*，并运行 -
dom初始化结果作为参考提供。

放在一起，我们的微调使用与
有监督的预训练对应物。这可能会导致 MoCo
处于*劣势*。即便如此，MoCo 仍然具有竞争力。这样做
也使得在多个方面进行比较成为可能
数据集/任务，无需额外的超参数搜索。

#### 4.2.1 PASCAL VOC 对象检测

设置。检测器是具有主干的Faster R-CNN [ 52 ]
R50-dilated-C5 或 R50-C4 [ 32 ]（详情见附录），
使用 BN 调整，在 [ 60 ] 中实现。我们微调所有铺设-
ers 端到端。图像比例为 [480, 800] 像素
训练和 800 推理。相同的设置用于所有
条目，包括有监督的预训练基线。我们
评估 AP 50的默认 VOC 指标（*即*IoU 阈值
是 50%) 以及更严格的 COCO 式 AP 指标
和 AP 75。评估是在 VOC test2007 集上进行的。

消融：骨干。表2显示了微调的结果
在 trainval07+12（~ 16.5k 图像）上。对于 R50-扩张-
C5（表2 a）中，上莫科IN-1M预先训练是可比
到有监督的预训练对应方，以及 MoCo 预训练
在 IG-1B 上训练*超过*它。对于R50-C4（表2 B），
带有 IN-1M 或 IG-1B 的 MoCo*优于*受监督的
对应物：最多 +0.9 AP 50、+3.7 AP 和 +4.9 AP 75。

有趣的是，传输精度取决于
探测器结构。对于 C4 主干，默认使用
在现有RESNET基于结果[ 14，61，26，66 ]，则AD-
无监督预训练的优势更大。关系
在预训练*与*. 探测器结构已被掩盖
过去，应该是一个考虑因素。

消融：对比损失机制。我们指出
这些结果部分是因为我们建立了可靠的检测
对比学习的基线。确定增益
即*仅*贡献使用莫科机制
在对比学习中，我们对预训练的模型进行微调
使用端到端或存储库机制，
由我们补充（*即*，图3 中最好的），使用
与 MoCo 相同的微调设置。

这些竞争对手表现不错（表3）。他们的AP
和具有 C4 骨干网的AP 75也高于
ImageNet 监督对应的，*参见*。表2 b，但其他
指标较低。他们在所有指标上都比 MoCo 差。
这显示了 MoCo 的好处。另外，如何训练
这些在大规模数据中的竞争者是一个悬而未决的问题，
他们可能不会从 IG-1B 中受益。

与之前的结果进行比较。继COM-
petitors，我们对 trainval2007 进行了微调（~ 5k 图像）
使用 C4 主干。比较见表4。

对于AP 50度，*没有*以前的方法能赶上
与其各自的监督预训练对应方。
MoCo 在任何 IN-1M、IN-14M（完整的 Ima-
遗传学），YFCC-100M [ 55 ]，和IG-1B可以*超越*的
监督基线。更大的收益出现在更严格的
gent 指标：高达 +5.2 AP 和 +9.0 AP 75。这些收获是
大于在 trainval07+12 中看到的增益（表2 b）。

#### 4.2.2 COCO 对象检测和分割

设置。该模型是带有 FPN [ 41 ] 的Mask R-CNN [ 32 ]
或 C4 主干，使用 BN 调整，在 [ 60 ] 中实现。这
训练期间图像比例为 [640, 800] 像素，为 800
在推理。我们对所有层进行端到端的微调。我们很好——
调整 train2017 集（~ 118k 图像）并评估
在 val2017 上。时间表是 [ 22 ] 中的默认 1× 或 2× 。

结果。表5显示了使用 FPN 在 COCO 上的结果
(表5 a, b) 和 C4 (表5 c, d) 骨架。随着
1× schedule，所有模型（包括 ImageNet 超
可见的同行）训练严重不足，如所示
由~ 2分间隙以2×时间表的情况。随着
2× schedule，MoCo*优于*其 ImageNet 监督
在两个主干中的所有指标中对应。

#### 4.2.3 更多下游任务

表6显示了更多下游任务（实现去
尾在附录中）。总体而言，MoCo 的表现具有竞争力

| 预训练 | AP BB | $AP^{BB}_{50}$ | $AP^{BB}_{75}$ | AP MK | $AP^{MK}_{50}$ | $AP^{MK}_{75}$ |
|---|---|---|---|---|---|---|
| 随机初始化 | 31.0 | 49.5 | 33.2 | 28.5 | 46.8 | 30.4 |
| 极好的。IN-1M | 38.9 | 59.6 | 42.7 | 35.4 | 56.5 | 38.1 |
| MoCo IN-1M | 38.5 (-0.4) | 58.9 (-0.7) | 42.0 (-0.7) | 35.1 (-0.3) | 55.9 (-0.6) | 37.7 (-0.4) |
| MoCo IG-1B | 38.9 (+0.0) | 59.4 (-0.2) | 42.3 (-0.4) | 35.4 (+0.0) | 56.5 (+0.0) | 37.9 (-0.2) |

(a) Mask R-CNN, R50-FPN, 1x schedule

| 预训练 | AP BB | $AP^{BB}_{50}$ | $AP^{BB}_{75}$ | AP MK | $AP^{MK}_{50}$ | $AP^{MK}_{75}$ |
|---|---|---|---|---|---|---|
| 随机初始化 | 36.7 | 56.7 | 40.0 | 33.7 | 53.8 | 35.9 |
| 极好的。IN-1M | 40.6 | 61.3 | 44.4 | 36.8 | 58.1 | 39.5 |
| MoCo IN-1M | 40.8 (+0.2) | 61.6 (+0.3) | 44.7 (+0.3) | 36.9 (+0.1) | 58.4 (+0.3) | 39.7 (+0.2) |
| MoCo IG-1B | 41.1 (+0.5) | 61.8 (+0.5) | 45.1 (+0.7) | 37.4 (+0.6) | 59.1 (+1.0) | 40.2 (+0.7) |

(b) Mask R-CNN, R50-FPN, 2x schedule

| 预训练 | AP BB | $AP^{BB}_{50}$ | $AP^{BB}_{75}$ | AP MK | $AP^{MK}_{50}$ | $AP^{MK}_{75}$ |
|---|---|---|---|---|---|---|
| 随机初始化 | 26.4 | 44.0 | 27.8 | 29.3 | 46.9 | 30.8 |
| 极好的。IN-1M | 38.2 | 58.2 | 41.2 | 33.3 | 54.7 | 35.2 |
| MoCo IN-1M | 38.5 (+0.3) | 58.3 (+0.1) | 41.6 (+0.4) | 33.6 (+0.3) | 54.8 (+0.1) | 35.6 (+0.4) |
| MoCo IG-1B | 39.1 (+0.9) | 58.7 (+0.5) | 42.2 (+1.0) | 34.1 (+0.8) | 55.4 (+0.7) | 36.4 (+1.2) |

(c) Mask R-CNN, R50-C4, 1x schedule

| 预训练 | AP BB | $AP^{BB}_{50}$ | $AP^{BB}_{75}$ | AP MK | $AP^{MK}_{50}$ | $AP^{MK}_{75}$ |
|---|---|---|---|---|---|---|
| 随机初始化 | 35.6 | 54.6 | 38.2 | 31.4 | 51.5 | 33.5 |
| 极好的。IN-1M | 40.0 | 59.9 | 43.1 | 34.7 | 56.5 | 36.9 |
| MoCo IN-1M | 40.7 (+0.7) | 60.5 (+0.6) | 44.1 (+1.0) | 35.4 (+0.7) | 57.3 (+0.8) | 37.6 (+0.7) |
| MoCo IG-1B | 41.1 (+1.1) | 60.7 (+0.8) | 44.8 (+1.7) | 35.6 (+0.9) | 57.4 (+0.9) | 38.1 (+1.2) |

(d) Mask R-CNN, R50-C4, 2x schedule

表 5. 在 COCO 上微调的对象检测和实例分割：评估了边界框 AP (AP bb ) 和掩码 AP (AP mk )
在 val2017 上。括号中是与 ImageNet 监督预训练对应物的差距。绿色是至少+0.5点的差距。

COCO关键点检测

| 预训练 | AP KP | 美联社 $^{kp}_{50}$ | 美联社 $^{kp}_{75}$ |
|---|---|---|---|
| 随机初始化 | 65.9 | 86.5 | 71.7 |
| 极好的。IN-1M | 65.8 | 86.9 | 71.9 |
| 莫科IN-1M | 66.8 ( + 1.0) | 87.4 ( + 0.5) | 72.5 ( + 0.6) |
| MoCo IG-1B | 66.9 ( + 1.1) | 87.8 ( + 0.9) | 73.0 ( + 1.1) |

COCO密集姿态估计

| 预训练 | AP DP | 美联社 $^{dp}_{50}$ | 美联社 $^{dp}_{75}$ |
|---|---|---|---|
| 随机初始化 | 39.4 | 78.5 | 35.1 |
| 极好的。IN-1M | 48.3 | 85.6 | 50.6 |
| 莫科IN-1M | 50.1 ( + 1.8) | 86.8 ( + 1.2) | 53.9 ( + 3.3) |
| MoCo IG-1B | 50.6 ( + 2.3) | 87.0 ( + 1.4) | 54.3 ( + 3.7) |

LVIS v0.5 实例分割

| 预训练 | AP MK | $AP^{MK}_{50}$ | $AP^{MK}_{75}$ |
|---|---|---|---|
| 随机初始化 | 22.5 | 34.8 | 23.8 |
| 极好的。IN-1M † | 24.4 | 37.8 | 25.8 |
| 莫科IN-1M | 24.1 ( − 0.3) | 37.4 ( − 0.4) | 25.5 ( − 0.3) |
| MoCo IG-1B | 24.9 ( + 0.5) | 38.2 ( + 0.4) | 26.4 ( + 0.6) |

| 预训练 | 城市景观实例段。 AP MK | $AP^{MK}_{50}$ | 语义赛格。（米欧） 城市景观 | 挥发性有机化合物 |
|---|---|---|---|---|
| 随机初始化 | 25.4 | 51.1 | 65.3 | 39.5 |
| 极好的。IN-1M | 32.9 | 59.6 | 74.6 | 74.4 |
| 莫科IN-1M | 32.3 ( − 0.6) | 59.3 ( − 0.3) | 75.3 ( + 0.7) | 72.5 ( − 1.9) |
| MoCo IG-1B | 32.9 ( + 0.0) | 60.3 ( + 0.7) | 75.5 ( + 0.9) | 73.6 ( − 0.8) |

表 6. MoCo 与 ImageNet 监督预训练，精细
调整各种任务。对于每个任务，相同的架构和
schedule 用于所有条目（见附录）。括号里是
与 ImageNet 监督预训练对应物的差距。在
绿色是至少+0.5点的差距。
†：此条目与 BN 冻结，这提高了结果；见正文。

使用 ImageNet 监督预训练：

*COCO 关键点检测*：有监督的预训练有
与随机初始化相比*没有*明显的优势，而
MoCo 在所有指标上都表现出色。

*COCO密集姿态估计*[ 1 ]：MoCo实质上
优于监督预训练，*例如*，3.7 分
美联社 $^{dp}_{75}$，在这个高度本地化敏感的任务中。

*LVIS v0.5 实例分割*[ 27 ]：这个任务有
~ 1000 个长尾分布式类别。具体在
用于 ImageNet 监督基线的 LVIS，我们发现
使用冻结的 BN (24.4 AP mk ) 进行调谐比可调谐更好
BN（详见附录）。所以我们将 MoCo 与
在这个任务中更好的监督预训练变体。交通部
IG-1B 在所有指标上都超过了它。

*城市景观实例分割*[ 10 ]：使用 IG-1B 的 MoCo
与它的监督预训练对应物相当
AP mk，并且在 AP mk 中更高 $^{}_{50}$ .

*语义分割*：On Cityscapes [ 10 ]，MoCo out-
执行其监督预训练对应物高达 0.9
观点。但是在VOC语义分割上，MoCo更差
至少 0.8 点，这是我们观察到的负面情况。

概括。总而言之，MoCo 可以胜过其 ImageNet
7 检测或分割中的监督预训练对应物
心理任务。5此外，MoCo 与 Cityscapes 不相上下
实例分割，在VOC语义上落后
分割; 我们在 iNatu 上展示了另一个类似的案例
ralist [ 57 ] 在附录中。总体而言，*MoCo 已基本关闭
无监督和有监督表示之间的差距
多视觉任务中的学习*。

值得注意的是，在所有这些任务中，MoCo 预先训练了
IG-1B 始终优于预先训练的 MoCo
IN-1M。这表明*MoCo 可以在这方面表现良好
大规模、相对未经整理的数据集*。这代表一个
*现实世界*无监督学习的场景。

## 五、讨论与结论

我们的方法显示了无监督的积极结果
在各种计算机视觉任务和数据集中学习。
一些悬而未决的问题值得讨论。MoCo 的印象——
从 IN-1M 到 IG-1B 的证明是一致的
但相对较小，表明更大规模的数据
可能没有被充分利用。我们希望有一个先进的借口
任务将改善这一点。超越了简单的实例discrim-
化任务[ 61 ]，可以采用MoCo为借口
诸如屏蔽自动编码之类的任务，*例如*，在语言中 [ 12 ] 和
在视觉中 [ 46 ]。我们希望 MoCo 对其他
涉及对比学习的借口任务。

5即VOC/COCO上的物体检测，实例分割
COCO/LVIS，COCO 上的关键点检测，COCO 上的密集姿态，以及
Cityscapes 上的语义分割。

## 参考


[1] Rıza Alp Güler、Natalia Neverova 和 Iasonas Kokkinos。DensePose：野外密集人体姿态估计。在*CVPR*，2018 年。

[2] 菲利普·巴赫曼、R Devon Hjelm 和威廉·布赫瓦尔特。通过最大化互信息来学习表示跨视图。*arXiv：1906.00910，2019*年。

[3] Mathilde Caron、Piotr Bojanowski、Armand Joulin 和马蒂斯·杜泽。无监督学习的深度聚类的视觉特征。在*ECCV*，2018 年。

[4] Mathilde Caron、Piotr Bojanowski、Julien Mairal 和 Ar-曼德乔林。图像特征的无监督预训练在非策划数据上。在*ICCV*，2019 年。

[5] Ken Chatfield、Victor Lempitsky、Andrea Vedaldi 和 An-画齐瑟曼。魔鬼在细节中：进行评估最近的特征编码方法。在*BMVC 中*，2011 年。

[6] 陈良杰、乔治·帕潘德里欧、亚索纳斯·科基诺斯、凯文·墨菲和Alan L制作Yuille。DeepLab：语义识别使用深度卷积网络进行年龄分割，atrous con-进化和完全连接的 CRF。*TPAMI*，2017 年。

[7] Ting Chen、Simon Kornblith、Mohammad Norouzi 和 Ge-offrey韩丁。一个简单的对比学习框架的视觉表现。*arXiv:2002.05709* , 2020。

[8] 陈新磊、范浩琪、罗斯·吉尔希克、何开明。通过动量对比学习改进基线。*arXiv:2003.04297* , 2020。

[9] Adam Coates 和 Andrew Ng。编码的重要性与使用稀疏编码和矢量量化进行训练。在*ICML*，2011 年。

[10] 马里乌斯·科尔兹、穆罕默德·奥姆兰、塞巴斯蒂安·拉莫斯、蒂莫雷菲尔德、马库斯·恩茨韦勒、罗德里戈·贝南森、乌维弗兰克、斯特凡·罗斯和伯恩特·席勒。城市景观语义城市场景理解数据集。在*CVPR 中*，2016 年。

[11] 邓嘉、魏东、理查德·索切尔、李嘉丽、李凯、还有李飞飞。ImageNet：大规模分层图像数据库。在*CVPR*，2009 年。

[12] Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina图塔诺娃。BERT：深度双向传输的预训练用于语言理解的前者。在*NAACL*，2019 年。

[13] Carl Doersch、Abhinav Gupta 和 Alexei A Efros。非超通过上下文预测视觉表示学习。在*ICCV*，2015 年。

[14] 卡尔·多尔施和安德鲁·齐瑟曼。多任务自有监督的视觉学习。在*ICCV*，2017 年。

[15] Jeff Donahue、Philipp Krähenbühl 和 Trevor Darrell。广告-通用特征学习。在*ICLR*，2017 年。

[16] 杰夫·多纳休和凯伦·西蒙尼安。大规模对抗表征学习。*arXiv：1907.02544，2019*年。

[17] Alexey Dosovitskiy、Jost Tobias Springenberg、Martin Ried-米勒和托马斯·布洛克斯。判别无监督使用卷积神经网络进行特征学习。在*NeurIPS*，2014 年。

[18] 马克·埃弗林厄姆、吕克·范古尔、克里斯托弗·基·威廉姆斯、约翰·温恩和安德鲁·齐瑟曼。帕斯卡视觉观察项目类 (VOC) 挑战。*IJCV*，2010 年。

[19] Spyros Gidaris、Praveer Singh 和 Nikos Komodakis。联合国-通过预测图像旋转的监督表示学习。在*ICLR*，2018 年。

[20] 罗斯·吉希克。快速 R-CNN。在*ICCV*，2015 年。

[21] 罗斯·吉希克、杰夫·多纳休、特雷弗·达雷尔和吉滕德拉马利克。丰富的特征层次结构，可实现准确的对象检测和语义分割。在*CVPR*，2014 年。

[22] 罗斯·吉尔希克、伊利亚·拉多萨沃维奇、乔治亚·吉奥萨里、皮奥特 Dollár 和 Kaiming He。检测器，2018 年。

[23] Aidan N Gomez、Mengye Ren、Raquel Urtasun和Roger B格罗斯。可逆残差网络：反向传播不存储激活。在*NeurIPS 中*，2017 年。

[24] 伊恩·古德费罗、让·普杰-阿巴迪、迈赫迪·米尔扎、宾徐、大卫·沃德-法利、谢尔吉尔·奥扎尔、亚伦·库维尔和约书亚·本吉奥。生成对抗网络。在*NeurIPS 中*，2014 年。

[25] Priya Goyal、Piotr Dollár、Ross Girshick、Pieter Noord-休斯、卢卡斯·韦索洛夫斯基、阿波·凯罗拉、安德鲁·图洛克、贾扬清、何开明。准确、大的小批量SGD：在 1 小时内训练 ImageNet。*arXiv：1706.02677，2017*年。

[26] Priya Goyal、Dhruv Mahajan、Abhinav Gupta 和 Ishan米斯拉。缩放和基准测试自我监督的视觉代表怨恨学习。在*ICCV*，2019 年。

[27] Agrim Gupta、Piotr Dollar 和 Ross Girshick。LVIS：A用于大词汇实例分割的数据集。在*CVPR 中*，2019。

[28] Michael Gutmann 和 Aapo Hyvärinen。噪音对比估计：非归一化状态的新估计原理理论模型。在*AISTATS*，2010 年。

[29] Raia Hadsell、Sumit Chopra 和 Yann LeCun。尺寸-通过学习不变映射来减少能力。在*CVPR 中*，2006 年。

[30] Bharath Hariharan、Pablo Arbeláez、Lubomir Bourdev，Subhransu Maji 和 Jitendra Malik。语义轮廓来自逆检测器。在*ICCV*，2011 年。

[31] Kaiming He、Ross Girshick 和 Piotr Dollár。重新思考 ImageNet 预训练。在*ICCV*，2019 年。

[32] Kaiming He、Georgia Gkioxari、Piotr Dollár 和 Ross Gir-shick。面膜R-CNN。在*ICCV*，2017年。

[33] 何开明，张翔宇，任少清，孙健。用于图像识别的深度残差学习。在*CVPR 中*，2016 年。

[34] 何开明，张翔宇，任少清，孙健。深度残差网络中的身份映射。在*ECCV 中*，2016 年。

[35] Olivier J Hénaff、Ali Razavi、Carl Doersch、SM Eslami 和亚伦·范登·奥尔德。数据高效的图像识别对比预测编码。*arXiv:1905.09272* , *2019*. Up-在https://openreview.net/访问的日期版本pdf?id=rJerHlrYwH 。

[36] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon，卡兰·格雷瓦尔、亚当·特里施勒和约书亚·本吉奥。学习-通过互信息估计进行深度表示和最大化。在*ICLR*，2019 年。

[37] Sergey Ioffe 和 Christian Szegedy。批量归一化：通过减少内部协作来加速深度网络训练变量转变。在*ICML*，2015 年。

[38] 亚历山大·科列斯尼科夫、翟晓华和卢卡斯·拜尔。关于-
访问自我监督的视觉表示学习。在
*CVPR*，2019 年。

[39] Yann LeCun, Bernhard Boser, John S Denker, Donnie
亨德森、理查德·E·霍华德、韦恩·哈伯德和
劳伦斯·D·杰克尔。应用于手写的反向传播
十个邮政编码识别。*神经计算*，1989。

[40] Sungbin Lim、Ildoo Kim、Taesup Kim、Chiheon Kim 和
金圣雄。快速AutoAugment。*的arXiv：1905.00397*，
2019。

[41] 林宗义、Piotr Dollár、Ross Girshick、何开明，
Bharath Hariharan 和 Serge Belongie。特征金字塔
用于物体检测的网络。在*CVPR*，2017 年。

[42] 林宗义、迈克尔·梅尔、塞尔吉·贝隆吉、詹姆斯·海斯、
Pietro Perona、Deva Ramanan、Piotr Dollár 和 C Lawrence
齐特尼克。Microsoft COCO：上下文中的常见对象。在
*ECCV*，2014 年。

[43] 乔纳森·朗、埃文·谢尔哈默和特雷弗·达雷尔。完全
用于语义分割的卷积网络。在
*CVPR*，2015 年。

[44] 德鲁夫·马哈詹、罗斯·吉尔希克、维涅什·拉马纳坦、
Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe,
和劳伦斯范德马腾。探索弱的极限
有监督的预训练。在*ECCV*，2018 年。

[45] Mehdi Noroozi 和 Paolo Favaro。无监督学习
通过解决拼图的视觉表现。在*ECCV 中*，
2016 年。

[46] Aaron van den Oord、Yazhe Li 和 Oriol Vinyals。代表-
使用对比预测编码的怨恨学习。
*arXiv：1807.03748，2018*年。

[47] 迪帕克·帕塔克、罗斯·吉希克、彼得·多拉尔、特雷弗·达雷尔、
和巴拉特·哈里哈兰。通过观察观察学习特征
对象移动。在*CVPR*，2017 年。

[48] 迪帕克·帕塔克、菲利普·克拉亨布尔、杰夫·多纳休、特雷弗
达雷尔和阿列克谢 A Efros。上下文编码器：功能
通过修复学习。在*CVPR*，2016 年。

[49] 彭超、肖太特、李泽明、江育宁、翔宇
张、凯佳、钢宇和孙健。MegDet：一个大
小批量对象检测器。在*CVPR*，2018 年。

[50] 亚历克·拉德福德、卡西克·纳拉辛汉、蒂姆·萨利曼斯和伊利亚
苏斯凯弗。通过生成提高语言理解
预训练。2018 年。

[51] 亚历克·拉德福德、杰弗里·吴、Rewon Child、大卫·栾、达里奥
阿莫迪和伊利亚·萨茨克弗。语言模型是超强的
可见多任务学习者。2019。

[52] 任少清、何开明、罗斯·吉尔希克、孙健。
Faster R-CNN：通过重新检测实现实时目标检测
gion 提议网络。在*NeurIPS 中*，2015 年。

[53] 凯伦西蒙尼安和安德鲁齐瑟曼。非常深入的对话
用于大规模图像识别的解决网络。在*ICLR 中*，
2015 年。

[54] Josef Sivic 和 Andrew Zisserman。视频谷歌：一段文字
视频中对象匹配的检索方法。在*ICCV 中*，
2003 年。

[55] 巴特·托梅、大卫·A·沙马、杰拉尔德·弗里德兰、本-
jamin Elizalde、Karl Ni、Douglas Polish、Damian Borth 和
李佳丽。YFCC100M：多媒体研究中的新数据。
*ACM 通讯*，2016 年。

[56] Yonglong Tian、Dilip Krishnan 和 Phillip Isola。康-
传递多视图编码。*arXiv:1906.05849 , 2019*.更新
在https://openreview.net/pdf 上访问的版本?
id=BkgStySKPB 。

[57] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui,
Chen Sun、Alex Shepard、Hartwig Adam、Pietro Perona 和
塞尔吉。iNaturalist 物种分类和
检测数据集。在*CVPR*，2018 年。

[58] 帕斯卡·文森特、雨果·拉罗谢尔、约书亚·本吉奥和
皮埃尔·安托万·曼扎戈尔。提取和组合健壮
具有去噪自编码器的功能。在*ICML*，2008 年。

[59] Xiaolong Wang 和 Abhinav Gupta。无监督学习
使用视频的视觉表示。在*ICCV*，2015 年。

[60] 吴宇新、亚历山大·基里洛夫、弗朗西斯科·马萨、万彦
罗和罗斯·吉尔希克。检测器2。https://github。
com/facebookresearch/detectron2，2019年。

[61] 吴志荣，熊元军，Stella Yu，林大华。联合国-
通过非参数实例离散的监督特征学习
犯罪。在*CVPR* , 2018. 更新版本访问：
https://arxiv.org/abs/1805.01978v1 。

[62] 谢赛宁、Ross Girshick、Piotr Dollár、涂卓文和
何开明。深度聚合残差转换
神经网络。在*CVPR*，2017 年。

[63] 孟野、张旭、Pong C Yuen 和 Shih-Fu Chang。联合国-
通过不变和传播的监督嵌入学习
实例功能。在*CVPR*，2019 年。

[64] Richard Zhang、Phillip Isola 和 Alexei A Efros。丰富多彩的
图像着色。在*ECCV*，2016 年。

[65] Richard Zhang、Phillip Isola 和 Alexei A Efros。裂脑
自动编码器：通过跨通道预编码的无监督学习
措辞。在*CVPR*，2017 年。

[66] Chengxu Zhuang、Alex Lin Zhai和Daniel Yamins。当地的
用于视觉嵌入的无监督学习的聚合。
在*ICCV* , 2019 中。从补充中访问的其他结果-
材料。