

000
001 **Unsupervised Deep Domain Adaptation on**
002 **People Detection**

003
004 Anonymous ECCV submission
005

006 Paper ID ***
007
008

009 **Abstract.** 这篇论文提出了基于无监督场景迁移下的在密集人群中人头
010 检测的问题。也就是给定一个在原场景充分训练好的深度神经网络模型，
011 如何在待迁移场景没有任何标记数据的情况下，训练一个能够识别待迁移
012 场景中人物的模型。首先，我们利用迭代算法来自动标记待迁移场景中的
013 人物。由于自动标记出来的人物数据中存在错误数据，假阳性的数据比例
014 会随着训练迅速增加，影响到整个网络。因此我们将原场景中的标记数据
015 加入到训练数据中，来抑制错误数据。同时，我们将神经网络最后的全连
016 通层等价转化为两个子层。在新添加的子层上加入非监督的正则化函数来
017 防止训练过程中的过拟合。跟将原场景中训练的原模型在没有利用待迁移
018 场景的数据来微调得到的结果相比，我们提出的方法将待迁移场景中人物
019 检测的召回率提高了xx,同时precision只从xx掉到xx。与此同时，我们将
020 我们的算法运用到了标准的迁移学习基准数据库Office数据库，我们的方
021 法在监督场景迁移喝非监督场景迁移都取得了最好的结果。
022

023 **Keywords:** 无监督场景迁移，深度卷积神经网络，人物检测
024
025
026

027 **1 Introduction**
028

029 深度卷积神经网络在计算机视觉的任务上有非常突出的表现，但是，大部分的
030 任务都需要大量的标记数据。像著名的PASCAL VOC和MS COCO，需要上千万的带标记的图片
031 数据来完成训练的过程。在监控的运用中，比如人物检测，
032 这个人工标记的过程就更加繁重了，因为需要标记到具体的位置。时至今日，
033 世界上又有着无数的摄像头来担任监控的功能。然而，由于监控场景变化巨
034 大，它受到背景，光线，视角，清晰度等等一系列的不稳定因素的影响，这就
035 使得对于所有的监控场景进行人工标记这个工作更加的不可实施。在人物检测
036 的任务中，需要上万的数据来训练一个好的检测神经网络。
037

038 当待迁移场景标记数据非常的少，甚至没有的情况下，场景迁移的任务就是
039 帮忙减少所需要的有标记的数据的数量，当我们在原场景有大量的标记数据，
040 那么这个迁移就需要考虑到原场景和待迁移场景之间的联系。大部分传统的场
041 景迁移的算法[1-5]都尝试找到原场景和待迁移场景之间的共同特征，或者将特
042 征投射到高维空间中和子空间中。最近，也有其他工作[6-8]提出了通过深度网
043 络来学习基于特殊场景的检测工具。在我们的人物检测的任务中，我们尝试着
044 将在原场景中充分训练的深度网络迁移到一个待迁移场景，这个待迁移场景并
没有任何标记数据。

糖趨烫
糖趨掏
糖趨渦
糖趨滔
糖趨縫
糖燙糖
糖燙倘
糖燙躺
糖燙淌
糖燙趟
糖燙烫
糖燙搣
糖燙湧
糖燙滔
糖燙縫
糖掏糖
糖掏倘
糖掏躺
糖掏淌
糖掏趟
糖掏燙
糖掏搣
糖掏湧
糖掏滔
糖掏縫
糖湧糖
糖湧倘
糖湧躺
糖湧淌
糖湧趟
糖湧燙
糖湧搣
糖湧湧
糖湧滔
糖湧縫
糖滔糖
糖滔倘
糖滔躺
糖滔淌
糖滔趟
糖滔燙
糖滔搣
糖滔湧
糖滔滔
糖滔縫

在这篇文章中，我们提出了基于人物检测的无监督场景迁移的新算法。使用原场景下的原模型进行初始化，我们首先利用迭代算法来自动标记待迁移场景的图片作为伪真实数据。在每一次迭代中，待迁移场景的模型被更新并用来自动标记下一个迭代需要的训练数据。然后，由于缺乏真阴性数据和伪阳性数据，自动标记数据中的错误将导致伪阳性的比例迅速上升，同时进一步影响到了召回率的提高。为了消除数据噪声的影响，我们提出了两个方法来正则化深度网络的训练。一方面，我们将来自原场景的有标记数据的图片加入了自动标记的数据中，来弥补真阴性数据的不足。另一方面，我们将深度网络中最后的全连通层的操作切分为两个子运算-元素点乘和求和。最后的全连通层也因此可以转化为两个子层，元素点乘层和求和层。基于此，一个无监督的正则化函数可以被添加到深度网络中作为无监督的损失函数来防止伪阳性结果的迅速增加以及增强召回率。

在人物检测的任务之外，我们也添加了基于标准迁移学习基准数据库的实验来评估我们的算法的适用性。我们的算法的迁移结果同时在监督和无监督的设置下超过了之前发布的其他人的工作，这证明了我们的迁移算法有足够的灵活度来处理检测和分类的任务。

这篇文章的贡献有这样三个方面：

- 我们提出了一个可行的解决方案来将在原场景充分训练的深度模型迁移到待迁移场景，而待迁移场景并不需要任何标记数据。这使得深度网络在不同监控情况下的部署变得更为容易。
- 由于大部分的算法都关注于最后一层的特征向量，我们往后一步来研究全连通层，并将最后一层全连通层转化为元素点乘层和求和层。因此一个无监督正则化函数可以加到元素点乘层，这样可以在抑制伪阳性和增加召回率方面取得更好的效果。
- 我们在标准迁移学习基准数据库的实验也表明了我们在其他深度场景迁移的任务中也可以取得好的结果。

这篇文章剩下的部分是这样组织的。首先章节2回顾了相关的工作，章节3给出了算法的细节部分，实验部分的结果在章节4 中做出介绍。章节5 总结了这篇论文。

2 Relate Work

在许多检测的工作中，在原场景中的大量数据的训练下得到的模型直接拿来检测待迁移场景的图片。他们假设待迁移场景的图片数据是原场景中图片数据的子集。然而当待迁移场景的数据分布跟原场景的数据分布差别很大时，原模型的表现会下降很多。场景迁移致力于得到更好的模型。

许多场景迁移的工作致力于找到原场景和待迁移场景之间的共同特征。Saenko et al. [1, 2]提出了一个基于线性变换的算法和一个基础核变化的算法来减少场景之间的变化。Gopalan et al. [3]将特征向量映射到Grassmann manifold而不是直接在元数据的特征上进行操作。或者，Mesnil et al. [9]使用迁移学习的算法来获得好的数据特征表示。然而这些方法都相当的局限，因为他们并没有学习基于特殊场景的特征来增加准确率。我们的正则化函数受到了这些工作的启发。

另外一组场景迁移的工作[4, 5, 10]是将原场景的数据分布跟待迁移场景的数据分布弄得尽量相似。在这些工作中，Maximum Mean Discrepancy (MMD) 被

用到作为重新从原场景中挑选数据的评估依据，来使得其于待迁移场景的数据分布尽量相似。在工作[11]中，MMD作为正则化工具加入到模型中以减少数据分布之间的差别。

同时，在学习基于特殊场景学习的检测算法也有一些工作。Wang et al.[6, 7]利用了上下文的信息来计算检测物体的置信度，同时他也提出了学习待迁移场景数据分布以及为特定场景的视觉模式设计了聚类层的算法。这些工作依赖于重新给自动标记的数据以计算权重并加入最后的损失函数以及需要额外的上下文设计以获得可靠的结果。同时，Hattori el al. [8]通过生成空间变化的行人模型来学习特定场景的检测模型。Pishchulin et al. [12]利用了一个3D的形状模型来生成训练数据。然而为了场景迁移来进行合成工作也很耗费人力。

3 我们的方法

在这个部分中，我们将介绍在人物检测的任务中的无监督场景迁移算法。我们将来自原场景的训练图片标记为 $\mathbf{X}^S = \{x_i^S\}_{i=1}^{N^S}$ ，把待迁移场景中的训练图片标记为 $\mathbf{X}^T = \{x_j^T\}_{j=1}^{N^T}$ 。对于原场景中的图片，我们将其对应的标记标记为 $\mathbf{B}_i^S = \{b_{i,k}^S\}_{k=1}^{N_i^S}$ ，其中 $b_{i,k}^S = (x, y, w, h) \in R^4$ 。然而待迁移场景中的图片的标记是自动标记的，我们标记为 $\tilde{\mathbf{B}}_j^T = \{\tilde{b}_{j,k}^T\}_{k=1}^{N_j^T}$ ，其中 $\tilde{b}_{j,k}^T = (\tilde{x}, \tilde{y}, \tilde{w}, \tilde{h}) \in R^4$ 。在迁移的过程中，待迁移场景的标记随着每次迭代而变化。在待迁移场景中，人工标记的数据仅仅用来做迁移算法的评估用。

我们的迁移算法框架分为两个框架流—愿框架流和待迁移框架流。图片x展示了这个框架。愿框架流将原场景的数据作为输入，待迁移框架流将待迁移场景的数据作为输入。这两个框架流可以利用任何的端到端的深度检测网络作为它们的模型。这里我们使用了在章节3.1中介绍的网络作为两个框架流中的模型。在初始化阶段，我们首先利用足够的原场景的标记数据来训练原框架流中的模型，有监督的损失函数来在预测区域进行回归。在训练收敛之后，原框架流的权重用来初始化待迁移框架流中的待迁移模型。在迁移阶段，迭代算法用来作为训练的方法。待迁移框架流中的待迁移模型在迭代过程中被不断更新，同时原框架流中的原模型权重不再更新。

我们的算法同时利用了监督的损失函数和非监督的损失函数来训练学习基于特定场景的检测模型，同时避免训练中的过拟合。对于监督的损失函数，我们利用自动标记的数据作为训练数据。由于自动标记的数据存在数据错误，我么需要利用非监督的损失函数来正则化整个网络。我们将愿框架的原模型作为数据分布的参考。因此我们在迁移过程中结合的无监督和监督的损失函数可以被这样描述：

$$L(\theta^T | \mathbf{X}^S, \mathbf{B}^S, \mathbf{X}^T, \tilde{\mathbf{B}}^T, \theta^S) = L_S + \alpha * L_U \quad (1)$$

$$L_S = \sum_{j=1}^{N^T} \sum_{k=1}^{N_j^T} (r(\theta^T | x_j^T, \tilde{b}_{j,k}^T) + c(\theta^T | x_j^T, \tilde{b}_{j,k}^T)) \quad (2)$$

$$L_U = L_{MMD}(\theta^T | \mathbf{X}^S, \mathbf{X}^T, \theta^S) \quad (3)$$

其中 L_S 是有监督的损失函数来学习基于场景的检测模型， L_U 是无监督的损失函数。 $r(\cdot)$ 是为了预测区域定位的回归函数，比如norm-1损失函数， $c(\cdot)$ 是

为了预测区域置信度的分类函数，比如cross-entropy损失函数。 $L_{MMD}(\cdot)$ 是基于MMD的无监督正则化函数。其中常数 α 用来平衡监督损失函数和无监督损失函数的影响。在我们的实验中，它对于不同的场景具有鲁棒性，我们取 $\alpha = 10$ 作为所有实验的 α 值。

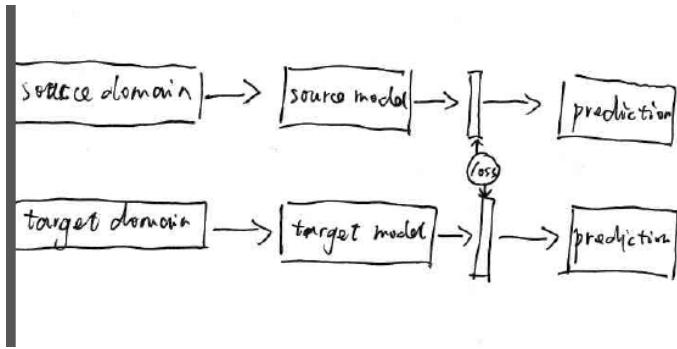


Fig. 1. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*) lead to the same summed estimate at x_s . This shows a figure consisting of different types of lines. Elements of the figure described in the caption should be set in italics, in parentheses, as shown in this sample caption. The last sentence of a figure caption should generally end without a full stop

3.1 检测网络

我们利用Russel et al.提出的模型来作为我们迁移框架流中的模型，这是一个端到端的人物检测模型，它不需要任何提前计算好的可能人物区域。这个检测网络包含了一个GoogLeNet [13]来将一张图片编码为一个(15x20x1024)的特征图，其中每一个1024维都是对应感应区域的数据表征，每一个感应区域对应了原图中的一个子区域。然后一个基于RNN的层将批量大小为300的数据表征解码为具体的300*5个输入，包括预测区域的位置和其对应的置信度。最后所有子图上的输出总结在一起作为最好的检测结果。当我们在原场景的大量数据充分训练之下，得到的深度网络在待迁移场景有较高的准确度，然后它的召回率并不高。同时，不同于其他需要提前计算可能的预测区域并一一给出置信度的检测网络[14, 15]，这个检测网络直接输出了所有的具有高置信度的预测区域。因此，负样本中可能包含人头区域也可能包含非人头区域，这并不能作为迁移时的训练数据。

3.2 迭代算法

这个章节中我们将介绍迁移过程中的迭代算法。自动检测工具将待迁移场景中置信度高的图片作为下一次迭代需要的训练数据。为了生成第一次迭代的自动标记数据，我们利用在原场景充分训练好的原模型来生成待迁移场景的训练数据。就像章节3.1中提到的那样，由原模型自动标记得到的待迁移场景下的训练数

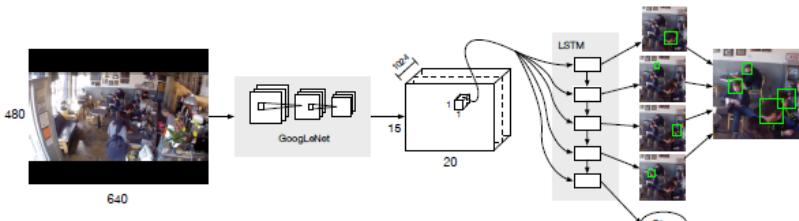


Fig. 2. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*) lead to the same summed estimate at x_s . This shows a figure consisting of different types of lines. Elements of the figure described in the caption should be set in *italics*, in parentheses, as shown in this sample caption. The last sentence of a figure caption should generally end without a full stop

数据具有低的召回率和高的准确率。在随后的迭代中，每次迭代需要的训练数据都有上一次迭代得到的更新的待迁移模型自动标记得到。在这些迭代中，这些自动标记的图片就作为训练数据来更新待迁移模型。

由于自动标记的待迁移场景的数据包含低的召回率。同时人物区域和非人物区域都混杂在负样本中，我们忽略了低置信度的区域的误差反传。也就是在训练的过程中我们鼓励网络更加大胆的预测正样本，而对于负样本保持保守的态度。这个策略无疑将导致许多非人物的区域都被预测为人物。当错误积累到一定程度，进一步的训练也无法提高召回率，反而适得其反。为了弥补真阴性数据的不足，我们将原场景中人工标记的图片数据加入到训练数据中。在我们的实验中，当训练这些来自原场景的标记图片时，我们只对那些具有低置信度的区域进行反传。同时章节3.3中的无监督损失函数加入了网络中进行正则化约束。完整的迁移算法在算法1中给出。在一个提前指定的迭代次数极限到达之后，我们就获得了最后的在迁移场景下学习到的检测模型。

3.3 Unsupervised weights regularizer on Element-wise Multiply Layer

Element-wise Multiply Layer

在深度神经网络中，最后一层的特征向量被作为输入图片的重要表征。然而，在这篇文章中我们往后一步，关注于最后一层的全连通层。这一层作为解码器将最后一层特征向量中包含的丰富信息表达为最后的输出层。由于原模型实在充分的原场景数据下训练得到的，因此其最后一层全连通层的参数也很好的收敛过。我们认为相比于在最后一层的特征向量加入正则化限制，如果在最后这一层全连通层加入了正则化的约束将会取得更好的效果。首先，我们标记最后一层特征向量、最后一层全连通层的参数和最后的输出对应为 $\mathbf{F}_{(N^B \times N^D)}$ ，

Algorithm 1 Deep domain adaptation algorithm (to be completed)

```

225 1: procedure DEEP DOMAIN ADAPTATION
226 2: Train source model on source stream with abundant annotated data
227 3: Use well-trained source model on source stream to initialize model on target stream
228 as  $M_0$ 
229 4:   for  $i = 0:N^I$  do
230      $M_i$  generate "fake ground truth"  $G_i$  of target domain
231     balbal
232     balbabla
233      $G_i = G_i +$  random samples from source domain
234     Take  $G_i$  as training data to upgrade  $M_i$  into  $M_{i+1}$ 
235 10:  end for
236 11:  $M_{N^I}$ : final model.
237 12: end procedure
238
239
240  $\mathbf{C}_{(N^D \times N^O)}$  and  $\mathbf{P}_{(N^B \times N^O)}$ 。最后一层全连通层的操作可以被表达为矩阵乘法:
241
242 
$$\mathbf{P} = \mathbf{F} * \mathbf{C} \quad (4)$$

243 
$$P_{b,o} = \sum_d F_{b,d} * K_{d,o} \quad (5)$$

244
245 从中得到可以得到启发，我们将上述共识切分为两个子操作—元素点乘和求
246 和，同样这两个操作可以描述为:
247
248
249 
$$M_{b,o,d} = F_{b,d} * C_{d,o} \quad (6)$$

250 
$$P_{b,o} = \sum_d M_{b,o,d} \quad (7)$$

251
252 其中  $\mathbf{M}_{(N^B \times N^O \times N^D)} = [\mathbf{m}_{b,o}]$  是元素点乘操作的参数张量。 $\mathbf{m}_{b,o}$  是一个具
253 有  $N^D$  维的向量，这将作为无监督正则化约束的基础。最后，我们可以将最
254 后一层特征向量和最后的输出之间的最后的全连通层转化一个元素点乘层和
255 求和层。这个转化来的元素点乘层是最后一个在输出层之前的带有参数权重的
256 层。图片x展示了这个转化。
257
258
259 Unsupervised weights regularizer on Element-wise Multiply Layer 在
260 一些工作中[decaf][1]，最后一层的特征向量被当做图片最后的表征。在场景迁
261 移的任务中，当原模型在大量原场景数据的训练之下，章节3.3中提到的元素
262 点乘层也包含了丰富且重要的信息，直接导致了最后的输出。对于输出层的特
263 定节点来说，它的最后输出值取决于被转化来的最后的元素点乘层，并且元素
264 点乘层每一维对其的贡献并不是随机确定的。在这篇文章中，我们认为原场景
265 和待迁移场景中数据在最后这层元素点乘层的分布应该是相似的。虽然在最后
266 一层元素点乘层中，每一维可能对最后的输出产生影响，但是由于原模型是在
267 大量的原场景的数据的训练之下得到的，那么它在元素点乘层上的分布应该具
268 有代表性，且待迁移模型在迁移的过程中也应该保持一致。这里我们利用MMD
269 (maximum mean discrepancy)来编码原场景数据和待迁移场景数据在元素点乘

```

$\mathbf{C}_{(N^D \times N^O)}$ and $\mathbf{P}_{(N^B \times N^O)}$ 。最后一层全连通层的操作可以被表达为矩阵乘法:

$$\mathbf{P} = \mathbf{F} * \mathbf{C} \quad (4)$$

$$P_{b,o} = \sum_d F_{b,d} * K_{d,o} \quad (5)$$

从中得到可以得到启发，我们将上述共识切分为两个子操作—元素点乘和求和，同样这两个操作可以描述为:

$$M_{b,o,d} = F_{b,d} * C_{d,o} \quad (6)$$

$$P_{b,o} = \sum_d M_{b,o,d} \quad (7)$$

其中 $\mathbf{M}_{(N^B \times N^O \times N^D)} = [\mathbf{m}_{b,o}]$ 是元素点乘操作的参数张量。 $\mathbf{m}_{b,o}$ 是一个具有 N^D 维的向量，这将作为无监督正则化约束的基础。最后，我们可以将最后一层特征向量和最后的输出之间的最后的全连通层转化一个元素点乘层和求和层。这个转化来的元素点乘层是最后一个在输出层之前的带有参数权重的层。图片x展示了这个转化。

Unsupervised weights regularizer on Element-wise Multiply Layer 在一些工作中[decaf][1]，最后一层的特征向量被当做图片最后的表征。在场景迁移的任务中，当原模型在大量原场景数据的训练之下，章节3.3中提到的元素点乘层也包含了丰富且重要的信息，直接导致了最后的输出。对于输出层的特定节点来说，它的最后输出值取决于被转化来的最后的元素点乘层，并且元素点乘层每一维对其的贡献并不是随机确定的。在这篇文章中，我们认为原场景和待迁移场景中数据在最后这层元素点乘层的分布应该是相似的。虽然在最后一层元素点乘层中，每一维可能对最后的输出产生影响，但是由于原模型是在大量的原场景的数据的训练之下得到的，那么它在元素点乘层上的分布应该具有代表性，且待迁移模型在迁移的过程中也应该保持一致。这里我们利用MMD (maximum mean discrepancy) 来编码原场景数据和待迁移场景数据在元素点乘

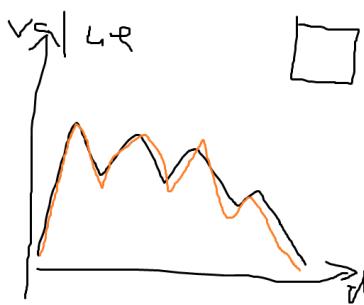


Fig. 3. One kernel at x_s (dotted kernel) or two kernels at x_i and x_j (left and right) lead to the same summed estimate at x_s .

层上的分布的相似性。

$$L_{MMD}(\theta^T | \mathbf{X}^S, \mathbf{X}^T, \theta^S) = \frac{1}{N^O} \sum_{o=1}^{N^O} \left\| \frac{1}{N^B} \sum_{b=1}^{N^B} \mathbf{m}_{b,o}^T - \frac{1}{N^B} \sum_{b=1}^{N^B} \mathbf{m}_{b,o}^S \right\|^2 \quad (8)$$

这个公示也可以解释为对于所有输出节点的 $\mathbf{m}_{b,o}^T$ 和 $\mathbf{m}_{b,o}^S$ 的中心之间的欧式距离的平均值。在实验的过程中，将所有的数据集都拿进来计算数据分布是不现实的，然后数据太少得到的数据分布不稳定，难以做正则化约束，因此在我们的实验中， $L_{MMD}(\cdot)$ 损失函数是由每一次的批量训练中得到的。 $\mathbf{m}_{b_i,o}^S$ 和 $\mathbf{m}_{b_j,o}^S$ 的比较可以在图x中看到。从图中可以看到原场景不同的数据在最后一层元素点乘层的分布是很接近的。

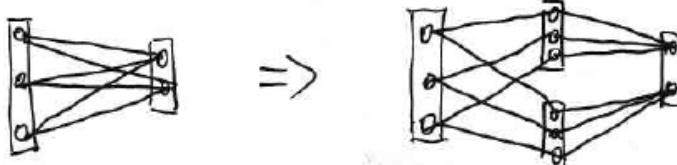


Fig. 4. One kernel at x_s (dotted kernel) or two kernels at x_i and x_j (left and right) lead to the same summed estimate at x_s .

4 实验结果

在这个章节中，我们将介绍我们在监控场景中的表现和在标准场景迁移基准数据集上的表现。由于我们最早做场景迁移的动机在于更方便的部署深度检测网络到监控场景中，因此我们首先展示我们在密集场景中人物检测的表现。然后我们将我们的迁移算法应用到标准场景迁移基准数据集上来验证我们的算法的适用性。

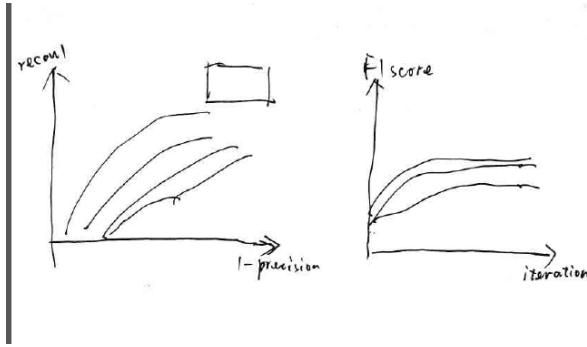


Fig. 5. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*) lead to the same summed estimate at x_s . This shows a figure consisting of different types of lines. Elements of the figure described in the caption should be set in *italics*, in parentheses, as shown in this sample caption. The last sentence of a figure caption should generally end without a full stop

4.1 监控场景中的场景迁移

数据集和评估标准

为了展示我们的场景迁移算法在人物检测任务中的效果，我们收集了一个包含了3个待迁移场景的数据集。这三个待迁移场景分别包含了1308, 1213和331张没有标记的图片。对于每个场景，我们额外标记了100张图片作为最后的测试数据。我们在人工标记的工程中并不是将整个人的身体都标记出来，而是将其头部标记出来。这样做的原因是在室内的人物检测或者密集场景中的人物检测，由于遮挡等原因，一个人的身体部分很可能是看不到的，选择人物头部作为人物的代表可以解决这个问题。原场景的数据集是来自Brainwash数据集 (<http://d2.mpi-inf.mpg.de/datasets>)。这个数据集包含了超过11917张来自三个场景的带标记的图片。原场景和待迁移场景的示意图如图x所见。

我们对于检测结果的评估方法来自于PASCAL VOC [16]中所定义的标准。为了判断一个预测区域是不是能够匹配一个真实的人物区域，他们之间的交集面积必须超过他们的合集面积的50%。如果多个检测结果预测了同一个人物，那么也只能当作一个正确率的预测。我们在图x中画出了准确率-召回率曲线图，同时，在迁移算法的迭代过程中的F1值 $F1 = 2 * precision * recall / (precision + recall)$ 也展示在图x中。

360 实验设定

361 我们使用了深度学习框架Caffe [17]作为我们的迁移学习平台。在场景
 362 迁移的过程中，我们将网络的学习率设为0.01，动量设为0.5。在初始化阶
 363 段，GoogLeNet的权重首先用来初始化愿框架流的原模型，同时RNN层的参数
 364 根据均一分布随机初始化。在每一次迭代中，100张自动标记的待迁移场景的图
 365 片和1000张原场景的标记图片交互的用来训练待迁移场景的网络。检测网络的
 366 输出包括预测区域的坐标及其对应的置信度，因此在最后一层特征向量和最后
 367 的输出之间有两个全连通层。经过我们的实验，发现当无监督的MMD正则化函
 368 数已经加到输出预测区域置信度的元素点乘层时，再将无监督的MMD正则化函
 369 数加到输出预测区域坐标的元素点乘层并不能更好的表现。我们的迁移算法
 370 在3个待迁移场景分别做了实验。

371 372 Comparison with baseline methods

373 为了展示算法的有效性，我们比较了4个不同的方法，其中方法4是我们最后
 374 的迁移算法：

- 376 1. 只有待迁移场景自动标记的数据做训练，不加任何的无监督正则化函数。
- 377 2. 只有待迁移场景自动标记的数据做训练，无监督的MMD正则化函数加到最
 378 后一层全连通层转化的元素点乘层。
- 379 3. 待迁移场景自动标记的数据和来自原场景的标记数据做训练，无监督
 380 的MMD正则化函数加到最后一层特征向量层。
- 381 4. 待迁移场景自动标记的数据和来自原场景的标记数据做训练，无监督
 382 的MMD正则化函数加到最后一层全连通层转化的元素点乘层。

383 图x画出了上诉几个比较方法的在待迁移场景1中的准确率-召回率曲线。同时，
 384 在迁移过程中每一次迭代的F1值的变化也在图x中显示出来。表格x给出了4个比
 385 较方法在三个迁移场景下，当F1的值达到最大时相应的准确率和召回率。迁移
 386 算法的结果示意图如图x所示。

	Scene 1			Scene 2			Scene 3		
	1-Pr	Re	F1	1-Pr	Re	F1	1-Pr	Re	F1
method 0	0.101	0.187	0.309	0.059	0.599	0.732	0.190	0.476	0.599
method 1	0.245	0.408	0.530	0.632	0.905	0.524	0.176	0.778	0.800
method 2	0.284	0.476	0.572	0.012	0.837	0.906	0.078	0.653	0.764
method 3	0.109	0.496	0.637	0.002	0.721	0.838	0.044	0.611	0.746
method 4	0.140	0.530	0.656	0.006	0.811	0.893	0.097	0.778	0.836

399 Performance evaluation

400 从表格x和图x中，我们有以下结论：

- 402 - 使用了迭代算法来更新待迁移场景中的模型的方法1、2、3、4的召回率数值
 403 比直接使用在原场景训练的原模型的方法0更高。这表现了我们的迭代算法
 404 在其自动标记和迭代式训练的有效性。

- 对比于方法1，方法2有着更高的F1值。他们区别在于是否在损失函数里面添加了无监督的正则化函数，证明了我们的算法可以抑制数据错误同时提高最终的召回率。
 - 方法4对比于方法2有着更高的F1值，这表明了增加的来自原场景的标记数据有助于减少在迁移过程中针对待迁移场景负样本不反传的错误。
 - 对比于方法3，方法4的召回率更高了，这表明了转化来的元素点乘层上的无监督正则化函数取得了比特征向量层更好的效果。

4.2 Domain Adaptation on Standard Classification Benchmark

Office dataset

Office数据集[1]包含了来自三个场景(Amazon, DSLR, Webcam)的31个分类。图x给出了样例图片。由于Amazon场景有着最多的2817张标记图片，我们把它作为原场景，将Webcam场景作为待迁移场景。训练数据的使用遵从了标准的有监督和无监督的设定。具体来说，对于无监督学习，我们在原场景Amazon每个类别随机采样了20张图片作为训练数据，对于有监督学习，我们额外在待迁移场景中每类选取了3个标记图片作为训练数据。在两种实验设定中，剩下的图片都用来作为评估以及用于无监督的MMD正则化计算。



Fig. 6. Some examples from three domains in the Office dataset.

Fig. 6. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*)

Experimental settings and network design

在有监督的实验设定中，我们服用了在人物检测中的框架流。我们利用AlexNet [18]作为两个框架流的基本模型。首先，我们利用Amazon场景提供的训练数据训练原框架流的原模型。然后在章节3.2中的迭代算法作为场景迁移的基本训练方法。自动标记工具将待迁移场景中置信度高于0.9的作为下一次迭代的训练数据。在每一次迭代过程中，随机从训练数据中挑选100张来更新模型。无监督的MMD正则化函数加到最后一层全连通层转化的元素点乘层，损失函数3中常数 α 设为10，这同人物检测中利用的常数设定相同。

对于无监督情况，我们使用了于有监督相同的实验设定和网络模型，除了在迁移的过程中，待迁移场景没有提供任何的可以添加到训练数据的人工标记数据。

450 Performance evaluation

451 在表格x中，我们跟6个最近发布的在有监督和无监督设定下的工作的结果。
 452 我们的方法在两种设定下都超过了其他方法的结果。这更加证实了我们的迭代
 453 算法以及无监督MMD正则化函数加到最后一层全连通层转化的元素点乘层在不
 454 同运用下的效果。

	$A \rightarrow W$	
	Supervised	Unsupervised
GFK(PLS,PCA)[19]	46.4	15.0
SA [20]	45.0	15.3
DA-NBNN [21]	52.8	23.3
DLID [22]	51.9	26.1
DeCAF ₆ S [23]	80.7	52.2
DaNN [11]	53.6	35.0
Ours	84.3	66.3

468 5 Conclusions

470 The paper ends with a conclusion.

472 References

1. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Computer Vision–ECCV 2010. Springer (2010) 213–226
2. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1785–1792
3. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 999–1006
4. Huang, J., Gretton, A., Borgwardt, K.M., Schölkopf, B., Smola, A.J.: Correcting sample selection bias by unlabeled data. In: Advances in neural information processing systems. (2006) 601–608
5. Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.: Covariate shift by kernel mean matching. Dataset shift in machine learning **3**(4) (2009) 5
6. Wang, X., Wang, M., Li, W.: Scene-specific pedestrian detection for static video surveillance. Pattern Analysis and Machine Intelligence, IEEE Transactions on **36**(2) (2014) 361–374
7. Zeng, X., Ouyang, W., Wang, M., Wang, X.: Deep learning of scene-specific classifier for pedestrian detection. In: Computer Vision–ECCV 2014. Springer (2014) 472–487
8. Hattori, H., Naresh Boddeti, V., Kitani, K.M., Kanade, T.: Learning scene-specific pedestrian detectors without real data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3819–3827



Fig. 7. One kernel at x_s (dotted kernel) or two kernels at x_i and x_j (left and right)

- 540 9. Mesnil, G., Dauphin, Y., Glorot, X., Rifai, S., Bengio, Y., Goodfellow, I.J., Lavoie,
541 E., Muller, X., Desjardins, G., Warde-Farley, D., et al.: Unsupervised and transfer
542 learning challenge: a deep learning approach. ICML Unsupervised and Transfer
543 Learning **27** (2012) 97–110
- 544 10. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discriminatively
545 learning domain-invariant features for unsupervised domain adaptation.
546 In: Proceedings of The 30th International Conference on Machine Learning. (2013)
547 222–230
- 548 11. Ghifary, M., Kleijn, W.B., Zhang, M.: Domain adaptive neural networks for object
549 recognition. In: PRICAI 2014: Trends in Artificial Intelligence. Springer (2014)
550 898–904
- 551 12. Pishchulin, L., Jain, A., Wojek, C., Andriluka, M., Thormählen, T., Schiele, B.:
552 Learning people detection models from few training samples. In: Computer Vision
553 and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1473–
554 1480
- 555 13. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D.,
556 Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings
557 of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1–9
- 558 14. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on
559 Computer Vision. (2015) 1440–1448
- 560 15. Vu, T.H., Osokin, A., Laptev, I.: Context-aware cnns for person head detection.
561 In: Proceedings of the IEEE International Conference on Computer Vision. (2015)
562 2893–2901
- 563 16. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman,
564 A.: The pascal visual object classes challenge: A retrospective. International
565 Journal of Computer Vision **111**(1) (2015) 98–136
- 566 17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama,
567 S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding.
568 arXiv preprint arXiv:1408.5093 (2014)
- 569 18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep
570 convolutional neural networks. In: Advances in neural information processing systems.
571 (2012) 1097–1105
- 572 19. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised
573 domain adaptation. In: Computer Vision and Pattern Recognition (CVPR), 2012
574 IEEE Conference on, IEEE (2012) 2066–2073
- 575 20. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual do-
576 main adaptation using subspace alignment. In: Proceedings of the IEEE Interna-
577 tional Conference on Computer Vision. (2013) 2960–2967
- 578 21. Tommasi, T., Caputo, B.: Frustratingly easy nbnn domain adaptation. In: Proceed-
579 ings of the IEEE International Conference on Computer Vision. (2013) 897–904
- 580 22. Chopra, S., Balakrishnan, S., Gopalan, R.: Dlid: Deep learning for domain adap-
581 tation by interpolating between domains. In: ICML workshop on challenges in
582 representation learning. Volume 2. (2013)
- 583 23. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.:
584 Decaf: A deep convolutional activation feature for generic visual recognition. arXiv
585 preprint arXiv:1310.1531 (2013)