ST++: Make Self-training Work Better for Semi-supervised Semantic Segmentation

Lihe Yang1 Wei Zhuo3 Lei Qi4,1 Yinghuan Shi1,2* Yang Gao1
State Key Laboratory for Novel Software Technology, Nanjing University
National Institute of Healthcare Data Science, Nanjing University
Tencent Southeast University

摘要

通过伪标记进行自我训练是利用未标记数据的一种传统的、模拟的、流行的管道。在这项工作中，我们首先通过在未标记的图像上注入强数据增强(SDA)来构建半监督语义分割的强自我训练基线(即ST)，以缓解过度拟合的噪声标签以及教师和学生之间的de - couple相似预测。通过这种简单的机制，我们的ST优于所有现有的方法，没有任何花哨的东西，例如迭代重新训练。受到这些令人印象深刻的结果的启发，我们深入研究了SDA，并提供了一些实证分析。为此，我们提出了一种先进的自我训练框架(即ST++)，该框架基于整体预测级稳定性，通过优先选择可靠的未标记图像来进行选择性再训练。

关键词：自我训练；SDA（强数据增强）

1 引言

语义分割一直是CV领域的一个重要研究方向，但是对于图像的标注，需要耗费极大的人工成本，因此就提出了半监督语义分割。半监督语义分割旨在用标记图像集D1和未标记图像集Du共同训练网络，来达到较好的分割性能。常用的半监督语义分割训练方法主要有伪标签和一致性正则化两种方式，然而对于伪标签这个方向来说确有比较大的局限性。该文章主要提出了一种无需任何花哨的功能，就能够高效的利用伪标签来训练，进而获得最先进的性能，并且在此基础上提出了一种更为先进的训练框架。

2 相关工作

2.1 半监督学习

近年来提出了两个主要分支，即一致性正则化(consistency reg)和熵最小化(entropy minimiztion)。一致性正则化强制当前优化模型在不同扰动(例如形状和颜色)下对相同的未标记数据产生稳定和一致的预测。早期的工作还节省了几个检查点或维护了一个教师，其参数为更新的学生的指数移动平均值，以便为学生模型生成更可靠的人工标签。另一方面，由自我训练管道推广的熵最小化以一种显式的自举方式利用无标签数据，其中未标记的数据被分配为伪标签，与手动标记的数据联合训练。与以往的研究不同，MixMatch综合了两种方法的优点，并提出了一种混合框架，从两种角度对未标签数据进行利用，这与本文的研究有些类似。

2.2 半监督语义分割

与半监督学习有略微的不同，半监督语义分割倾向于利用生成对抗网络(GANs)作为对未标记数据

的辅助监督信号。但是，GANs不容易优化，可能会出现模式崩溃的问题。因此，同样受到SSL中的success的启发，后续的方法试图用更简单的机制来解决这一任务，例如在多个摄动嵌入下，在两个不同的上下文收成下，以及在两个不同的初始化模型之间强制类似的预测。作为FixMatch的扩展，pseudoSeg将弱到强一致性适配于分段(seg)隔离(seg)场景，并进一步应用校准模块来细化伪掩码。

2.3 自我训练

伪标记自我训练是一种显式的经典方法，起源于十多年前。近年来，它越来越受到多个领域的关注，如全监督图像年龄识别、半监督学习和领域适应。特别是在半监督设置中，它已经在几个任务中被重新讨论，包括图像分类，目标检测和语义分割。然而，本文的工作与之有根本的不同，因为我们证明了在无标记数据上适当的SDA（强数据增强）对半监督学习者非常有益，而设计他们的方法是基于过度的数据增强会破坏干净的数据分布的假设。另一项工作通过手动设计与任务相关的增强来解决目标检测任务，而我们的SDA在图像识别中很常见，但在半监督语义分割中被忽略了。此外，上述两种方法都采用了普通的训练管道，而本篇文章进一步提出了ST++，以课程学习的方式安全地挖掘未标记的图像。

2.4 不确定的预测

之前的方法用贝叶斯分析估计模型的不确定性。然而，受贝叶斯推理计算量的限制，其他一些方法采用Dropout和数据aug来测量不确定度。在半监管的设置中，FixMatc只是设置了一个置信度阈值来过滤不确定的样本，而DMT主要设置了两个不同初始化的网络来突出显示不一致的区域。与它们相比，本文的方法通过测量进化面具的整体预测稳定性来估计图像级的不确定性，而不需要训练额外的网络或手动选择阈值，使其适用于其他场景。此外，该模型还学习了高置信度图像中的整体的上下文区域，使其更稳定，更适合于分割任务。

3 本文方法

3.1 本文方法概述

半监督语义分割旨在从像素级标记图像和未标记图像的组合集中生成标签，在大多数工作中，总体优化目标形式为$L = L_s + \lambda L_u$。作者首先提出了一个简单的自我训练方案，它包括三个步骤：

1、用已标签训练数据，采用全监督的方式来训练teacher model，用的是交叉熵损失函数。

2、用步骤一训练好的teacher model来预测未标记数据，以此生成伪标签。

3、将已标记图像和已产生伪标签的未标记图像合并，然后采用全监督的方式来对student model进行训练。

上述自我训练方案长期以来收到批评，因为伪标签中的错误会累积，并大大降低student model的效果。此外，在这样一个自我引导的过程中，第二次全监督训练过程中引入了不充分的信息，导致teacher model和student model出现了严重的耦合问题。

为了改进前面提到的两个问题，即过拟合伪标签，和teacher model和student model之间的预测耦合，文章提出了在训练student model的过程中，注入SDA（强数据增强）在未标记图像上。SDA在这

里被命名为正则型全监督语义分割中采用的弱或基本扩充，包括正则型调整大小、随机裁剪和随机翻转。至于SDA的具体选择，文章设法在不同的数据集或设置之间保持统一策略。

ST中，我们通过直接过度采样Dl到与Du相同的尺度来模拟这一选择，然后从合并的数据集中统一采样。通过这样的方式，在半监督的定义下，采用全监督的方式进行优化。具体伪代码如下：

```
Algorithm 1: ST Pseudocode
  Input:  Labeled training set Dl = {(xi , yi)}M i=1,
          Unlabeled training set Du = {ui} N i=1,
          Weak/strong augmentations Aw/As ,
          Teacher/student model T /S
  Output: Fully trained student model S
  Train T on Dl with cross-entropy loss Lce
  Obtain pseudo labeled D^u = {(ui , T(ui))} N i=1
  Over-sample Dl to around the size of D^u
  for minibatch {(xk, yk)} B k=1 ⊂ (Dl ∪ D^u ) do
      for k ∈ {1, . . . , B} do
          if xk ∈ Du then
              xk, yk ← As (Aw((xk, yk))
          else
              xk, yk ← Aw(xk, yk)
          y^k = S(xk)
      Update S to minimize Lce of {(^yk, yk)} B k=1
return S
```

然而，尽管直向前ST框架取得了令人印象深刻的结果，但它平等地对待每个未标记的样本，并考虑到个别样本的固有可靠性和难度，以同样的方式利用它们。某些硬样例中的错误预测可能会对训练过程产生负面影响。因此，在目前先进的ST++框架下，文章提出了一种选择性再训练方案，通过优先选择可靠的未标记样本，以一种容易-困难的课程学习方式安全地利用整个unlabeled集。

以往的工作从不同的角度估计图像或像素的可靠性或不确定性，例如将最终的softmax输出作为置信度分布，并通过预定义的阈值过滤低置信度像素。在ST++中，希望用单一的训练模型来测量可靠性，而不需要手动选择置信度阈值。为了更稳定地评估可靠性，根据图像级信息(而不是广泛采用的像素级信息)过滤掉不可靠的样本。图像级的选择也使模型能够在训练期间学习更全面的上下文模式。具体伪代码如下：

```
Algorithm 2: ST++ Pseudocode
  Input:  Labeled training set Dl = {(xi , yi)}M i=1,
          Unlabeled training set Du = {ui} N i=1,
          Weak/strong augmentations Aw/As ,
          Teacher/student model T /S
  Output: Fully trained student model S
          Train T on Dl and save K checkpoints {Tj} K j=1
          for ui ∈ Du do
              for Tj ∈ {Tj} K j=1 do
                  Pseudo mask Mij = Tj (ui)
              Compute si with Equation 4 and {Mij} K j=1
          Select R highest scored images to compose Dul
```

```
            Du2 = Du − Du1
            Du1 = {(uk, T(uk))}uk∈Du1
            Train S on (Dl ∪ Du1 ) with ST re-training
            Du2 = {(uk, S(uk))}uk∈Du2
            Re-initialize S
            Train S on (Dl ∪ Du1 ∪ Du2 ) with ST retraining
return S
```

3.2 特征提取模块

文章采用了最常用的backbone ResNet-50以及ResNet-101，作为特征提取器，以及使用全监督语义分割近几年最常用的网络DeepLab作为主网络。

3.3 损失函数定义

损失函数由两部分组成，一部分是已标记数据，一部分是未标记数据。因此损失函数如下：

$$L = L_s + \lambda L_u$$

4 复现细节

4.1 与已有开源代码对比

主要差异有一下几个部分：

1、特征提取模块，文章采用的是pytoch官方提供的ResNet代码，而我将其简化，在将代码简便的同时，以不影响性能。但是相对于官方的源码，可能通用性性没那么高。

2、分割网络，对于Deeplab模型中的ASPP模块，我根据原论文的思路，在自己搜寻pytorch官方给出的文档，将其复现。

3、训练函数模块，由于作者已近给出了伪代码，所以我顺着伪代码的思路，将其复现，并且通过调试最终将其成功运行。而关于训练的参数如学习率和batchsize等，则按照原论文给出的参数使用。

最终模型跑出来的效果，和作者原论文给出的结果基本一致。

4.2 实验环境搭建

本次复现是在linux上运行，使用的是以下的深度学习框架版本：

torch	1.10.2+cu113
torchaudio	0.10.2+cu113
torchvision	0.11.3+cu113
tqdm	4.64.1

除此之外，还有其他使用的库这里就不一一列举。

4.3 使用说明

文件目录如下所示：

```
├── ./pretrained
    ├── resnet50.pth
    └── resnet101.pth
```

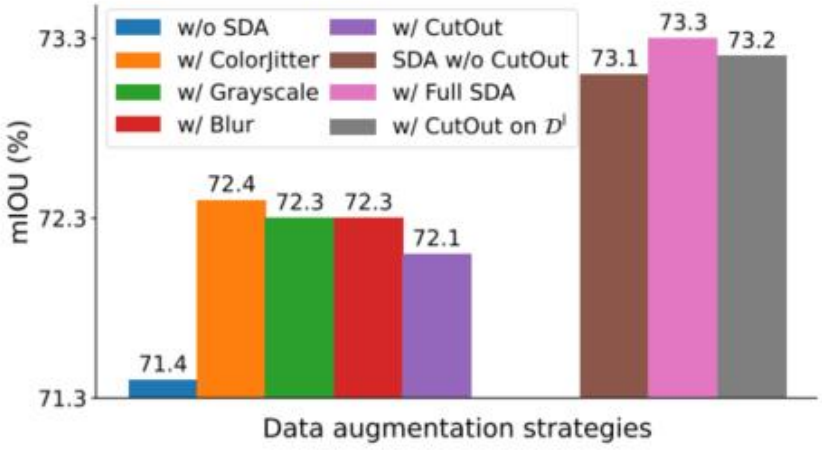
```
├── [PascalVOC Path]
    ├── JPEGImages
    └── SegmentationClass

├── [Cityscapes Path]
    ├── leftImg8bit
    └── gtFine
```

最后运行main.py即可。

5 实验结果分析

关于数据增强的使用：

文章中提到了使用多种数据增强：



根据作者给出的这个表格，我做了些实验，结果和图所示不变。

关于最终性能：

我做了一些实验，其结果和作者论文中给出的结果差距不大。

Network	Method	1/16 (662)	1/8 (1323)	1/4 (2645)	Network	Method	1/16 (662)	1/8 (1323)	1/4 (2645)
PSPNet ResNet-50	SupOnly	63.8	67.2	69.6	DeepLabv3+ ResNet-50	SupOnly	64.8	68.3	70.5
	CCT [41]	62.2	68.8	71.2		ECS [37]	-	70.2	72.6
	DCC [31]	67.1	71.3	72.5		DCC [31]	70.1	72.4	74.0
	ST	***69.1***	***73.0***	***73.2***		**ST**	***71.6***	***73.3***	***75.0***
	ST++	**69.9**	**73.2**	**73.4**		**ST++**	**72.6**	**74.4**	**75.4**
DeepLabv2 ResNet-101	SupOnly	64.3	67.6	69.5	DeepLabv3+ ResNet-101	SupOnly	66.3	70.6	73.1
	AdvSSL [26]	62.6	68.4	69.9		S4GAN [38]	69.1	72.4	74.5
	S4L [57]	61.8	67.2	68.4		GCT [28]	67.2	72.5	75.1
	GCT [28]	65.2	70.6	71.5		DCC [31]	72.4	74.6	76.3
	ST	***68.6***	***71.6***	***72.5***		**ST**	***72.9***	***75.7***	***76.4***
	ST++	**69.3**	**72.0**	**72.8**		**ST++**	**74.5**	**76.3**	**76.6**

上面表格是文章中作者给出的实验结果，下边是我的实验结果

Network	Method	1/16	1/8	1/4
PSPNet ResNet-50	ST	69.2	71.8	73.3
	ST++	69.8	72.4	74.6
DeepLabv3+ ResNet-50	ST	71.5	73.5	74.3
	ST++	72.6	73.7	75.1
DeepLabv2 ResNet-101	ST	67.7	72.9	72.1
	ST++	68.3	72.3	72.4
DeepLabv3+ ResNet-101	ST	72.9	75.6	74.7
	ST++	74.6	76.1	76.8

6 总结与展望

本次选的这篇半监督语义分割，与之前提出的模型框架不同，是在训练的baseline上进行改变，可以适用于大多数半监督语义分割场景。提出了一个能够高效使用unlabel数据的训练方法，并且在此基础上进行了一些改进。

目前实现过程中能够基本达到论文水平，但是代码复用性不高。以及没有提出一些有效的创新点。未来的话希望能够在在训练策略上进行改进。

参考文献

[1] Yang L , Zhuo W , Qi L , et al. ST++: Make Self-training Work Better for Semi-supervised Semantic Segmentation:, 10.48550/arXiv.2106.05095[P]. 2021.