用于语义分割的全卷积网络

Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley
{jonlong,shelhamer,trevor}@cs.berkeley.edu

Abstract

卷积网络是强大的视觉模型，可以产生特征的层次结构。本文表明，卷积网络本身，以端到端、像素到像素的方式训练，在语义分割方面超过了最先进的技术。本文的关键见解是建立"全卷积"网络，接受任意大小的输入，并产生相应大小的输出，具有高效的推理和学习。定义并详细介绍了*fully convolutional networks*的空间，解释了它们在空间密集预测任务中的应用，并与之前的模型建立了联系。将当代分类网络*(AlexNet [19]*、*VGG net [31]*和*GoogLeNet [32])*改编为*fully convolutional networks*，并通过将*[4]*微调到分割任务来迁移其学习到的表示。定义了一种新的架构，将来自深层粗层的语义信息与浅层细层的外观信息相结合，以产生准确和详细的分割。*fully convolutional network*实现了*PASCAL VOC(2012*年*62.2*%平均*IU*的相对改进*20*%*)*、*NYUDv2*和*SIFT*流的最先进分割，而对典型图像的推理耗时不到五分之一秒。

1. 简介

卷积网络正在推动识别领域的进步。卷积网络不仅在全图像分类方面有所改进[19, 31, 32]，而且在具有结构化输出的局部任务上也取得了进展。这些包括边界框目标检测[29, 12, 17]，部分和关键点预测[39, 24]和局部通信[24, 9]的进展。

从粗略到精细推理的自然下一步是对每个像素进行预测。之前的方法使用卷积网络进行语义分割[27, 2, 8, 28, 16, 14, 11]，其中每个像素都被标记为其封闭对象或区域的类，但这项工作解决了缺点。

本文表明，fully convolutional network(FCN)在语义分割上经过端到端、像素到像素的训练，超过了最先进的技术，而无需进一步的机器。据我们所知，这是第一个训练FCNs端到端(1)像素预测和(2)监督预训练的工作。现有网络的全卷积版本从任意大小的输入预测密集输出。学习和推理都是通过密集的前馈计算和反向传播一次全图像进行的。网络内上采样层在具有次采样池化的网络中实现像素级预测和学习。

该方法是一种有效的、渐进的和绝对的方法，并且

*作者贡献均等

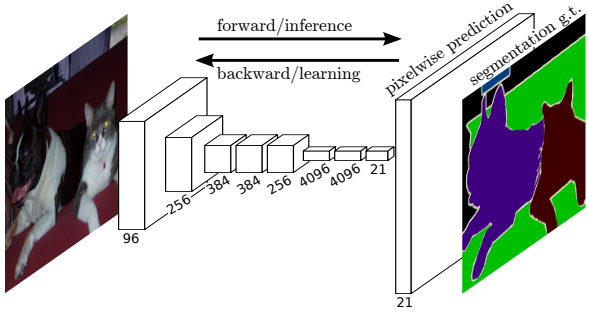

Figure 1. 全卷积网络可以有效地学习对语义分割等逐像素任务进行密集预测。

排除了其他工作中的复杂性。Patchwise训练很常见[27, 2, 8, 28, 11]，但缺乏全卷积训练的效率。该方法没有利用预处理和后处理的复杂性，包括超像素[8, 16]，建议[16, 14]，或随机字段或局部分类器[8, 16]的事后细化。该模型通过将分类网络重新解释为完全卷积和对其学习到的表示进行微调，将最近在分类[19, 31, 32]方面的成功转移到密集预测上。相比之下，之前的工作在没有监督的情况下应用了小型卷积网络[8, 28, 27]。

语义分割面临语义和位置之间的内在紧张关系:全局信息解决什么问题，而局部信息解决哪里问题。深度特征层次在局部到全局金字塔中联合编码位置和语义。我们定义了一个新的"跳过"架构，以结合深度、粗粒度、语义信息和浅层、细粒度、外观信息在4.2部分(见图3)。

在下一节中，我们回顾了深度分类网络FCNs的相关工作，以及最近使用卷积网络进行语义分割的方法。以下部分解释FCN设计和密集预测权衡，介绍我们具有网络内上采样和多层组合的架构，并描述我们的实验框架。在PASCAL VOC 2011-2、NYUDv2和SIFT Flow上展示了最先进的结果。

2. 相关工作

该方法借鉴了深度网络在图像分类[19, 31, 32]和迁移学习[4, 38]方面的最新成功。迁移首先在各种视觉识别任务[4, 38]上进行演示，然后在检测上，以及在混合建议-分类器模型[12, 16, 14]中的实例和语义分割上进行演示。现在，我们重新构建和微调分类网络，以直接、密集地预测语义分割。在这个框架中，我们绘制

了FCNs的空间，并放置了之前的模型，包括历史的和最近的。

Fully convolutional networks 据我们所知，将卷积网络扩展到任意大小输入的想法首次出现在Matan *et al.* [25]中，它将经典的LeNet [21]扩展到识别数字串。因为他们的网络仅限于一维输入字符串，Matan *et al.*使用Viterbi解码来获得他们的输出。Wolf和Platt [37]将convnet输出扩展为邮政地址块的四个角的检测分数的二维地图。这两项历史工作都进行了完全卷积的推理和学习以进行检测。Ning *et al.* [27]定义一个卷积网络，用全卷积推理对秀丽隐杆线虫组织进行粗分类分割。

在当今的多层网络时代，全卷积计算也得到了利用。Sermanet的滑动窗口检测*et al.* [29]，Pinheiro和Collobert的语义分割[28]，Eigen的图像恢复*et al.* [5]进行全卷积推理。完全卷积训练很少见，但Tompson有效地使用*et al.* [35]来学习端到端的部件检测器和用于姿态估计的空间模型，尽管他们没有暴露或分析这种方法。

或者，他*et al.* [17]放弃分类网络的非卷积部分，以制作一个特征提取器。它们将建议和空间金字塔池化相结合，以产生用于分类的本地化、定长特征。虽然快速有效，但这种混合模型不能进行端到端学习。

Dense prediction with convnets 最近的几项工作将卷积网络应用于密集预测问题，包括由Ning *et al.* [27], Farabet *et al.* [8]，以及Pinheiro和Collobert [28]进行的语义分割; Ciresan的电子显微镜边界预测*et al.* [2]和Ganin和Lempitsky的混合神经网络/最近邻模型的自然图像边界预测[11]; 以及图像复原和深度估计的Eigen *et al.* [5, 6]。这些方法的共同要素包括

- 小模型限制了容量和感受野;
- 分段训练[27, 2, 8, 28, 11];
- 通过超像素投影、随机场正则化、滤波或局部分类进行后处理[8, 2, 11];
- 输入移位和输出交错密集输出[28, 11]介绍了OverFeat [29];
- 多尺度金字塔处理[8, 28, 11];
- 饱和tanh非线性[8, 5, 28];而且
- 套装[2, 11],

而我们的方法没有这种机制。然而，我们从FCNs的角度研究了patchwise训练3.4和"shift-and-stitch"密集输出3.2。我们还讨论了网络内上采样3.3，其中Eigen *et al.* [6]的全连接预测是一个特例。

与这些现有方法不同，自适应和扩展了深度分类架构，将图像分类作为监督预训练，并进行完全卷积微调，以简单有效地从整个图像输入和整个图像地面传输中学习。

Hariharan *et al.* [16]和Gupta *et al.* [14]同样使深度分类网络适应语义分割，但在混合建议-分类器模型中是这样做的。这些方法通过采样边界框和/或区域建议来微调R-CNN系统[12]，用于检测、语义分割和实例分割。这两种方法都不是端到端的学习。

他们分别在PASCAL VOC分割和NYUDv2分割方面取得了最先进的结果，因此我们直接将我们独立的端到端FCN与5部分中的语义分割结果进行比较。

3. Fully convolutional networks

卷积网络中的每一层数据都是一个大小为$h \times w \times d$的三维数组，其中h和w是空间维度，d是特征或通道维度。第一层是图像，像素大小为$h \times w$，颜色通道为d。较高层中的位置对应于它们路径连接到的图像中的位置，这些位置称为它们的感受野。

卷积网络是基于平移不变性建立的。它们的基本组件(卷积、池化和激活函数)在局部输入区域上运行，并且仅依赖于相对空间坐标。为特定层中位置(i, j)的数据向量编写$\mathbf{x}_{ij}$，为下一层编写$\mathbf{y}_{ij}$，这些函数计算输出$\mathbf{y}_{ij}$ by

$$\mathbf{y}_{ij} = f_{ks}\left(\{\mathbf{x}_{si+\delta i, sj+\delta j}\}_{0 \le \delta i, \delta j \le k}\right)$$

其中k称为内核大小，s是步幅或子采样因子，f_{ks}确定层类型:用于卷积或平均池化的矩阵乘法，用于最大池化的空间最大值，或用于激活函数的elementwise非线性，等等。

这种函数式形式在composition中保持，内核大小和步长服从转换规则

$$f_{ks} \circ g_{k's'} = (f \circ g)_{k'+(k-1)s', ss'}.$$

虽然一般的深度网络计算一般的非线性函数，但只有这种形式的层的网络计算非线性滤波器，我们称之为深度滤波器或全卷积网络。FCN自然地对任何大小的输入进行操作，并产生相应的(可能重新采样的)空间维度的输出。

由FCN组成的实值损失函数定义了一个任务。如果损失函数是最后一层($\ell(\mathbf{x}; \theta) = \sum_{ij} \ell'(\mathbf{x}_{ij}; \theta)$)的空间维度的总和，则其梯度将是其每个空间分量的梯度的总和。因此，在ℓ上计算整个图像的随机梯度下降将与在ℓ'上计算的随机梯度下降相同，将所有最终层感受野作为一个小批量。

当这些感受野显著重叠时，在整个图像上逐层计算而不是逐块独立计算时，前馈计算和反向传播都要高效得多。

接下来，我们将解释如何将分类网络转换为产生粗输出图的全卷积网络。对于逐像素预测，我们需要将这些粗输出连接回像素。3.2节描述了OverFeat [29]为此引入的一个技巧。我们通过将其重新解释为等效的网络修改来了解这一技巧。作为一种有效的替代方案，我们在3.3节中介绍了用于上采样的反卷积层。在3.4节中，我们考虑通过分段采样进行训练，并在4.3节中给出证据，表明我们的整个图像训练更快且同样有效。

3.1. 适应密集预测的分类器

典型的识别网络，包括LeNet [21], AlexNet [19]及其更深层次的后继者[31, 32]，表面上接受固定大小的输

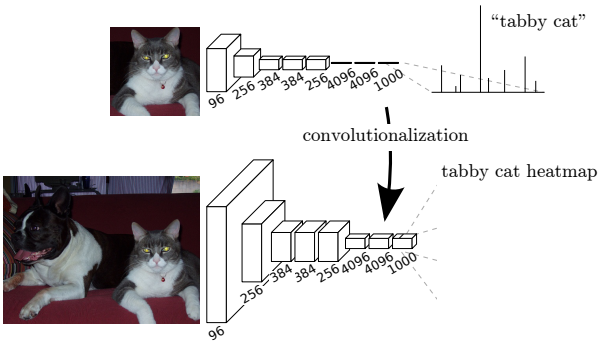

Figure 2. 将全连接层转换为卷积层使分类网络能够输出热图。添加层和空间损失(如图1所示)可产生用于端到端密集学习的高效机器。

入并产生非空间输出。这些网络的全连接层具有固定的尺寸，并且没有空间坐标。然而，这些全连接层也可以被视为卷积，其核覆盖了整个输入区域。这样做将它们强制转换到fully convolutional networks中，该s接受任何大小的输入并输出分类图。这种转换如图2所示。(相比之下，非卷积网络，如Le *et al.* [20]的网络，缺乏这种能力。)

此外，虽然产生的映射等效于对特定输入块的原始网络的评估，但计算在这些块的重叠区域上高度摊销。例如，AlexNet使用1.2 ms(在典型的GPU上)来生成227×227图像的分类分数，而全卷积版本使用22 ms来从500×500图像生成输出的10×10网格，这比naïve方法[1]快5倍以上。

这些卷积模型的空间输出图使它们成为处理语义分割等密集问题的自然选择。由于每个输出单元都有真实值，前向和后向传递都是直接的，都利用了卷积的固有计算效率(和积极优化)。

对于AlexNet示例，对应的向后时间是单个图像的2.4 ms和全卷积10×10输出映射的37 ms，从而产生与正向传递类似的加速。这种密集的反向传播如图1所示。

虽然我们将分类网络重新解释为完全卷积，但对于任何大小的输入，都可以产生输出映射，但输出维度通常通过降采样来减少。分类网子样本保持过滤器小，计算要求合理。这粗化了这些网络的全卷积版本的输出，将其从输入的大小减少到等于输出单元感受野的像素步幅的因子。

3.2. Shift-and-stitch是过滤稀疏

输入移位和输出交错是一种技巧，它可以从粗输出中产生密集预测，而无需插值，由OverFeat [29]引入。如果输出按f因子进行下采样，则输入(按左内边距和上内边距)向右移动x像素，向下移动y像素，$(x, y) \in \{0, \ldots, f-1\} \times \{0, \ldots, f-1\}$的每个值移动一次。这些$f^2$输入都通过卷积网络运行，并且输出是交错的，因此预测对应于其感受野中心的像素。

[1]假设对单个图像输入进行高效批处理。生成单幅图像的分类分数需要5.4 ms，比完全卷积的版本慢了近25倍。

仅更改卷积网络的滤波器和层步长就可以产生与此shift-and-stitch技巧相同的输出。考虑一个输入步幅s的层(卷积或池化)，以及一个滤波器权重f_{ij}的后续卷积层(排除特征维度，在这里不相关)。将下层的输入步幅设置为1对其输出进行s因子的上采样，就像shift-和-stitch一样。然而，将原始滤波器与上采样输出卷积并不能产生与技巧相同的结果，因为原始滤波器只看到其输入的减少部分(现在是上采样)。为了重现这个技巧，将滤镜放大为

$$f'_{ij} = \begin{cases} f_{i/s,j/s} & \text{if } s \text{ divides both } i \text{ and } j; \\ 0 & \text{otherwise,} \end{cases}$$

(i和j从0开始计数)。要再现该技巧的全部净输出，需要一层一层地重复这个滤波器放大，直到所有的子采样都被删除。

简单地减少网络内的次采样是一种权衡:滤波器看到更精细的信息，但具有更小的感受野，并需要更长的计算时间。我们已经看到，移位缝合技巧是另一种权衡:在不减少滤波器的感受野大小的情况下使输出更密集，但是滤波器被禁止以比原始设计更精细的尺度访问信息。

虽然我们已经用移位-缝合做了初步实验，但我们没有在我们的模型中使用它。我们发现，通过上采样进行学习(如下一节所述)更有效、更高效，特别是与稍后介绍的跳跃层融合相结合时。

3.3. 上采样是向后跨步卷积

另一种将粗输出连接到密集像素的方法是插值。例如，简单的双线性插值通过仅依赖于输入和输出单元的相对位置的线性映射从最近的四个输入计算每个输出y_{ij}。

从某种意义上说，具有f因子的上采样是具有$1/f$分数输入步幅的卷积。只要f是积分，因此上采样的自然方法是反向卷积(有时称为反卷积)，输出步幅为f。这样的操作实现起来很简单，因为它只是反转卷积的前向和后向传递。因此，通过像素级损失的反向传播，在网络内进行上采样以进行端到端学习。

请注意，该层中的反卷积滤波器不需要固定(例如，双线性上采样)，但可以学习。反卷积层和激活函数的堆栈甚至可以学习非线性上采样。

实验发现，网络内上采样对于学习密集预测是快速有效的。我们最好的分割架构使用这些层来学习上采样以进行细化预测，参见4.2部分。

3.4. 分段训练是损失采样

在随机优化中，梯度计算由训练分布驱动。拼接训练和全卷积训练都可以产生任何分布，尽管它们的相对计算效率取决于重叠和小批量大小。整幅图像全卷积训练与patchwise训练相同，其中每个批次由图像(或图像集合)损失以下单元的所有感受野组成。虽然这比均匀采样更有效，但它减少了可能的批次数量。然而，随机选择图像块可能简单地恢复。将损失限制为其空间项的随机采样子集(或，等效地在输出和损失

之间应用DropConnect掩码[36])排除了梯度计算中的补丁。

如果保持的补丁仍然有很大的重叠，全卷积计算仍然会加快训练。如果梯度是在多个反向通道中累积的，则批次可以包括来自多个图像的补丁。[2]

分块训练中的采样可以纠正类不平衡[27, 8, 2]并缓解密集块的空间相关性[28, 16]。在全卷积训练中，类别平衡也可以通过对损失进行加权来实现，损失采样可以用来解决空间相关性。

我们在4.3节中探索了采样训练，并没有发现它对密集预测产生更快或更好的收敛。全图像训练是有效且高效的。

4. 分割架构

我们将ILSVRC分类器转换为FCNs，并通过网络内上采样和像素级损失来增强它们以进行密集预测。通过微调进行分割训练。构建了一种新的跳跃结构，结合粗、语义和局部、外观信息来细化预测。

在PASCAL VOC 2011分割挑战赛[7]上进行训练和验证。用逐像素多项logistic损失进行训练，用平均像素交并比的标准度量进行验证，均值接管所有类别，包括背景。训练会忽略在真实情况中被掩盖(模糊或困难)的像素。

4.1. 从分类器到稠密FCN

我们首先对经过验证的分类架构进行卷积，如3节所示。我们考虑了赢得ILSVRC12的AlexNet [3]架构[19]，以及VGG nets [31]和GoogLeNet[4] [32] 这在ILSVRC14中表现得特别好。我们选择VGG的16层网络[5]，我们发现它在这个任务上等同于19层网络。对于GoogLeNet，

[2]请注意，并不是每个可能的patch都以这种方式包含，因为最终层单元的感受野位于一个固定的、跨步的网格上。然而，通过将图像向左和向下移动一个随机值以达到步幅，可以从所有可能的块中恢复随机选择。

[3]使用公开可用的`CaffeNet`参考模型。

[4]由于GoogLeNet没有公开可用的版本，我们使用自己的重新实现。该版本在较少的数据增强下进行了训练，并获得了68.5%的top-1和88.4%的top-5 ILSVRC精度。

[5]使用Caffe模型库中的公开版本。

Table 1. 本文自适应并扩展了三种分类卷积网络以进行分割。我们通过PASCAL VOC 2011验证集上的平均交并比和推理时间(在NVIDIA Tesla K40c上对500 × 500输入平均进行20次试验)来比较性能。详细介绍了自适应网络在密集预测方面的架构:参数层的数量、输出单元的感受野大小以及网络内的最粗步长。(这些数字给出了在固定学习率下获得的最佳性能，但不是可能的最佳性能。)

	FCN-AlexNet	FCN-VGG16	FCN-GoogLeNet[4]
mean IU	39.8	**56.0**	42.5
forward time	50 ms	210 ms	59 ms
conv. layers	8	16	22
parameters	57M	134M	6M
rf size	355	404	907
max stride	32	32	32

我们只使用最终的损失层，并通过丢弃最终的平均池化层来提高性能。我们通过丢弃最终的分类器层来砍掉每个网络，并将所有全连接层转换为卷积。我们附加一个带通道维度21的1 × 1卷积来预测每个PASCAL类(包括背景)在每个粗输出位置的分数，然后是一个反卷积层，将粗输出双线性上采样到像素密集输出，如3.3节所述。表1比较了初步验证结果以及每个网络的基本特征。本文报告了以固定的学习率(至少175次)收敛后取得的最佳结果。

从分类到分割的微调为每个网络提供了合理的预测。即使是最差的模型也达到了最先进的性能~75%。配备分割的VGG网络(FCN-VGG16)似乎已经是最先进的，平均val为56.0 IU，而测试[16]为52.6 IU。对额外的数据进行训练，将性能提高到在val [7]的子集上的59.4平均IU。培训细节见4.3部分。

尽管分类精度类似，但我们的GoogLeNet实现与此分割结果不匹配。

4.2. 结合what和where

我们定义了一个新的全卷积网络(FCN)用于分割，它结合了特征层次结构的层，并改进了输出的空间精度。参见图3。

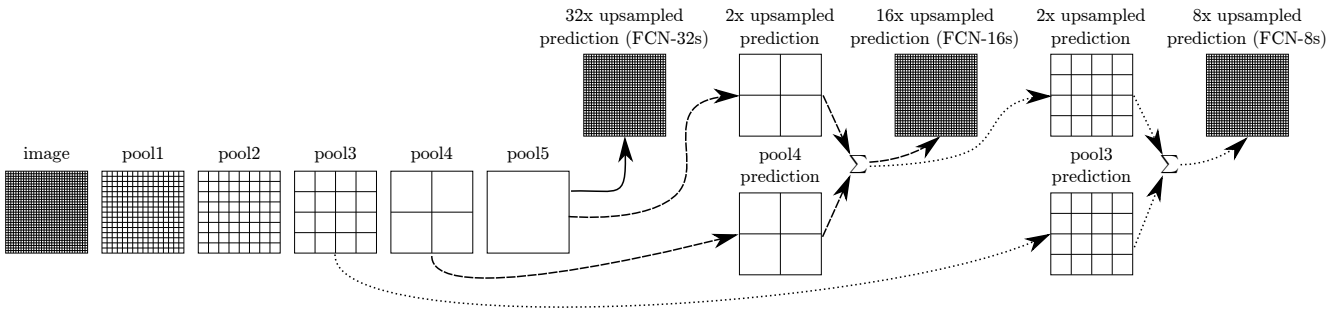

Figure 3. 我们的DAG网络学习将粗、高层信息与细、低层信息结合起来。图层显示为网格，显示相对空间粗糙。只显示了池化层和预测层;中间卷积层(包括转换后的全连接层)被省略。实线(FCN-32s):我们的单流网络，在4.1节中描述，上样本在一个步骤中将32个预测步幅回像素。虚线(FCN-16s):结合来自最终层和池层的预测，在stride 16时，让我们的网络预测更精细的细节，同时保留高层语义信息。虚线(FCN-8s): `pool3`的额外预测，步幅8，提供了进一步的精度。

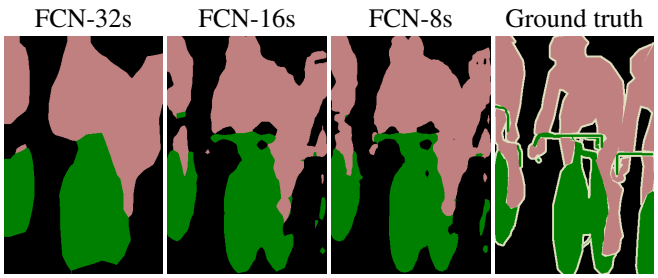

Figure 4. 通过融合来自不同步幅的层的信息来细化全卷积网络，可提高分割细节。前三个图像显示了我们的32、16和8像素步幅网络的输出(参见图3)。

虽然完全卷积的分类器可以微调到分割，如4.1所示，甚至在标准指标上得分很高，但它们的输出是粗糙的，令人失望(参见图4)。最终预测层的32像素步长限制了上采样输出中的细节尺度。

我们通过添加链接来解决这个问题，这些链接将最终的预测层与步幅更细的较低层相结合。这将线拓扑转换为DAG，其中的边从较低的层跳到较高的层(图3)。由于它们看到的像素更少，更精细的尺度预测应该需要更少的层，因此从较浅的净输出中制作它们是有意义的。结合精细层和粗层可以让模型做出尊重全局结构的局部预测。通过类比Florack的多尺度本地喷射*et al.* [10]，我们称我们的非线性局部特征层次为深喷射。

首先通过从16像素步幅层进行预测，将输出步幅分成两半。我们在`pool4`上添加1×1卷积层以产生额外的类预测。我们通过添加2×上采样层，将此输出与在`conv7`(卷积`fc7`)之上计算的预测融合在步幅32上和[6]两个预测的总和。(参见图3)。我们初始化2×上采样为双线性插值，但允许参数学习，如3.3节所述。最后，将stride 16的预测结果上采样回图像。我们称这个网络为FCN-16s。FCN-16s是端到端的学习，用最后一个更粗的网络的参数初始化，我们现在称之为FCN-32s。作用于`pool4`的新参数是零初始化的，因此网络以未修改的预测开始。学习率降低了100倍。

学习这个skip net可以将验证集的性能提高3.0 IU到62.4。图4显示了输出精细结构的改进。我们将这种融合与仅从池层学习(这会导致较差的性能)进行了比较，并只是降低了学习率，而没有添加额外的链接(这会导致不显著的性能改进，而没有提高输出质量)。

我们继续以这种方式融合来自`pool3`的预测与来自`pool4`和`conv7`融合的预测的2×上采样，构建网络FCN-8s。我们获得了62.7平均IU的轻微额外改进，并发现输出的平滑性和细节略有改善。在这一点上，我们的融合改进的回报越来越小，无论是就强调大规模正确性的IU指标而言，还是就可见的改进而言，例如在图4中，因此我们不再继续融合更低的层。

Refinement by other means 减少池化层的步长是获得更精细预测的最直接方法。然而，对于我们基于vgg16的网络来说，这样做是有问题的。将`pool5`层设置为步幅1需要我们的卷积`fc6`具有14×14的内核大小，以保持其感受野大小。除了它们的计算成本之外，我们还难以学习如此大的滤波器。我们尝试用较小的过滤器重新构建`pool5`之上的层，但未能成功实现可比的性能;一种可能的解释是，来自上层imagenet训练权重的初始化很重要。

获得更精细预测的另一种方法是使用3.2节中描述的移位缝合技巧。在有限的实验中，我们发现该方法的改进成本比低于层融合。

[6]由于梯度切换，最大融合使得学习变得困难。

Table 2. skip FCNs在PASCAL VOC2011验证子集上的比较[7]。学习是端到端的，除了FCN-32s-fixed，其中只有最后一层是微调的。请注意，FCN-32s是FCN-VGG16，重命名以突出步幅。

	pixel acc.	mean acc.	mean IU	f.w. IU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	**90.3**	**75.9**	**62.7**	**83.2**

4.3. 实验框架

Optimization 我们通过带动量的SGD进行训练。我们通过直线搜索选择FCN-AlexNet，FCN-VGG16和FCN-GoogLeNet，分别使用20张图像的小批量大小和10^{-3}, 10^{-4}和5^{-5}的固定学习率。我们使用动量0.9, 5^{-4}或2^{-4}的权重衰减，并将偏差的学习率提高一倍，尽管我们发现训练对这些参数不敏感(但对学习率敏感)。我们对类评分卷积层进行零初始化，找到随机初始化，既没有产生更好的性能，也没有更快的收敛。Dropout包含在原始分类器网络中使用的地方。

Fine-tuning 我们通过在整个网络中反向传播来微调所有层。与表2相比，仅对输出分类器进行微调只产生全部微调性能的70%。考虑到学习基本分类网络所需的时间，从头开始训练是不可行的。(请注意，VGG网络是分阶段训练的，而我们从完整的16层版本开始初始化。) 对于粗略的FCN-32s版本，单个GPU上的微调需要三天，升级到FCN-16s和FCN-8s版本各需要一天左右。

Patch Sampling 正如在3.4节中解释的那样，我们的完整图像训练有效地将每个图像批量处理为大的、重叠的补丁的规则网格。相比之下，之前的工作在完整数据集[27, 2, 8, 28, 11]上随机采样补丁，可能导致更高的方差批次，可能加快收敛[22]。本文通过以前面描述的方式对损失进行空间采样来研究这种权衡，做出独立选择以某种概率忽略每个最终层单元$1-p$。为了避免改变有效的批大小，我们同时将每个批的图像数量增加一个因子$1/p$。请注意，由于卷积的效率，对于足够大的p值，这种形式的拒绝采样仍然比patchwise训练更快(例如，根据3.1节中的数字，至少对于$p > 0.2$)。图5显示了这种抽样形式对收敛性的影响。与整幅图像训练相比，采样对收敛速度没有显著影响，但由于每批需要考虑的图像数量更多，采样所需的时间明显更

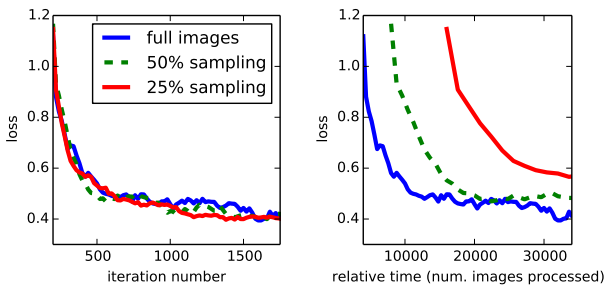

Figure 5. 对整个图像进行训练与对图像块进行采样一样有效，但通过更有效地利用数据，可以获得更快的收敛速度(墙时间)。左图显示了在预期批量大小固定的情况下，采样对收敛速度的影响，而右图则显示了相对壁时间的影响。

长。因此，在其他实验中，我们选择无采样的整幅图像训练。

Class Balancing 全卷积训练可以通过对损失进行加权或采样来平衡类别。虽然我们的标签略有不平衡(关于3/4是背景)，但我们发现类平衡是不必要的。

Dense Prediction 通过网络内的反卷积层将分数上采样到输入维度。最后一层反卷积滤波器固定为双线性插值，而中间上采样层初始化为双线性上采样，然后进行学习。移位和缝合(3.2节)，或过滤器稀疏等效，不使用。

Augmentation 我们尝试通过随机镜像和"抖动"图像来增加训练数据，将图像在每个方向上平移到32像素(最粗略的预测尺度)。这没有产生明显的改善。

More Training Data PASCAL VOC 2011分割挑战训练集，我们将其用于表1，标记了1112张图像。Hariharan *et al.* [15]收集了更大的8498个PASCAL训练图像集的标签，这些图像被用于训练之前最先进的系统SDS [16]。该训练数据提高了FCN-VGG16验证分数[7] 平均IU为59.4，下降3.4点。

Implementation 所有模型都在单个NVIDIA Tesla K40c上使用Caffe [18]进行训练和测试。模型和代码将在发布时开源。

5. 结果

在语义分割和场景解析上测试了FCN，探索了PASCAL VOC、NYUDv2和SIFT流。虽然这些任务在历史上区分了物体和区域，但我们将两者都视为像素预测。我们评估了FCN skip架构[8] 然后将其扩展到NYUDv2的多模态输入和SIFT流的语义和几何标签的多任务预测。

Metrics 本文报告了来自常见语义分割和场景解析评估的四个指标，它们是像素精度和区域并交点(IU)的变化。设n_{ij}为预测类别i属于类别j的像素个数，其中有n_{cl}个不同的类别，设$t_i = \sum_j n_{ij}$为类别i的像素总数。我们计算:

- 像素精度: $\sum_i n_{ii} / \sum_i t_i$
- 平均精度: $(1/n_{\text{cl}}) \sum_i n_{ii}/t_i$
- 平均IU: $(1/n_{\text{cl}}) \sum_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$
- 频率加权IU: $(\sum_k t_k)^{-1} \sum_i t_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$

PASCAL VOC 表3给出了我们的FCN-8s在PASCAL VOC 2011和2012测试集上的性能，并将其与之前的最先进的技术SDS [16]和著名的R-CNN [12]进行了比较。我们以20%的相对优势在平均IU [9]上取得了最好的结果。推理时间减少114×(仅convnet，忽略建议和改进)或286×(总体)。

Table 3. 所提出的全卷积网络在PASCAL VOC 2011和2012测试集上比最先进的水平有20%的相对改进，并减少了推理时间。

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [12]	47.9	-	-
SDS [16]	52.6	51.6	~ 50 s
FCN-8s	**62.7**	**62.2**	~ **175 ms**

NYUDv2[30]是使用Microsoft Kinect收集的RGB-D数据集。它有1449张RGB-D图像，具有像素化标签，这些标签已被Gupta合并为40类语义分割任务*et al.* [13]。我们报告了795张训练图像和654张测试图像的标准分割结果。(注意:所有模型选择都是在PASCAL 2011 val上执行的。) 表4给出了我们的模型在几种变体中的性能。首先，我们在RGB图像上训练未修改的粗模型(FCN-32s)。为了添加深度信息，我们在一个升级为四通道RGB-D输入的模型上进行训练(早期融合)。这几乎没有提供什么好处，可能是因为很难在模型中一直传播有意义的梯度。在Gupta *et al.* [14]的成功之后，我们尝试深度的三维HHA编码，仅在此信息上训练网络，以及RGB和HHA的"后期融合"，其中来自两个网络的预测在最后一层进行求和，由此产生的双流网络是端到端的学习。最后，我们将这个后期融合网络升级到16步版本。

SIFT Flow是一个包含2688张图像的数据集，具有33个语义类别("桥"、"山"、"太阳")的像素标签，以及三个几何类别("水平"、"垂直"和"天空")。一个FCN可以自然地学习一个联合表示，同时预测两种类型的标签。学习具有语义和几何预测层和损失的双头版本FCN-16s。学习到的模型在两个任务上的表现与两个独立训练的模型一样好，而学习和推理本质上与每个独立模型本身一样快。将标准分割为2488张训练图像和200张测试图像，计算结果如表5所示。[10] 在这两项任务上展示最先进的性能。

[7] PASCAL VOC 2011 val集包含了来自[15]的训练图像，因此我们在736张图像的不相交集上进行验证。本文的一个早期版本错误地对整个val集进行了计算。

[8] 我们的模型和代码可以在https://github.com/BVLC/caffe/wiki/Model-Zoo中公开。

[9] 这是测试服务器提供的唯一指标。

[10] 其中三个SIFT流类别在测试集中不存在。我们对所有33个类别进行了预测，但在评估中只包括测试集中实际存在的类别。(本文早期版本报告了较低的平均IU，包括评估中现有或预测的所有类别。)

Table 4. NYUDv2测试结果。*RGBD*是RGB通道和深度通道在输入端的早期融合。*HHA*是[14]作为水平视差的深度嵌入，与地面的高度，以及局部表面法线与推断的重力方向的夹角。*RGB-HHA*是联合训练的后期融合模型，对RGB和HHA的预测进行求和。

	pixel acc.	mean acc.	mean IU	f.w. IU
Gupta *et al.* [14]	60.3	-	28.6	47.0
FCN-32s RGB	60.0	42.2	29.2	43.9
FCN-32s RGBD	61.5	42.4	30.5	45.5
FCN-32s HHA	57.1	35.2	24.2	40.4
FCN-32s RGB-HHA	64.3	44.9	32.8	48.0
FCN-16s RGB-HHA	**65.4**	**46.1**	**34.0**	**49.5**

Table 5. SIFT Flow [10]具有类别分割(中)和几何分割(右)的结果。Tighe [33]是一种非参数迁移方法。Tighe 1是一个范例SVM，而2是SVM + MRF。Farabet是一个在类别平衡样本(1)或固有频率样本(2)上训练的多尺度卷积网络。Pinheiro是一个多尺度的循环卷积网络，记为$_3(\circ^3)$。几何形状的度量是像素精度。

	pixel acc.	mean acc.	mean IU	f.w. IU	geom. acc.
Liu *et al.* [23]	76.7	-	-	-	-
Tighe *et al.* [33]	-	-	-	-	90.8
Tighe *et al.* [34] 1	75.6	41.1	-	-	-
Tighe *et al.* [34] 2	78.6	39.2	-	-	-
Farabet *et al.* [8] 1	72.3	50.8	-	-	-
Farabet *et al.* [8] 2	78.5	29.6	-	-	-
Pinheiro *et al.* [28]	77.7	29.8	-	-	-
FCN-16s	**85.2**	**51.7**	39.5	76.1	**94.3**

6. 结论

Fully convolutional network卷积网络是一类非常丰富的模型，现代分类卷积网络是其中的一个特例。认识到这一点，将这些分类网络扩展到分割，并使用多分辨率层组合改进架构，大大提高了最先进的技术，同时简化和加快了学习和推理。

Acknowledgements 这项工作得到了美国国防部高级研究计划局MSEE和SMISC项目的部分支持，NSF奖项IIS-1427425，IIS-1212798，IIS-1116411，以及NSF GRFP，丰田和伯克利视觉与学习中心的支持。我们非常感谢NVIDIA对GPU的捐赠。我们感谢Bharath Hariharan和Saurabh Gupta的建议和数据集工具。我们感谢Sergio guadarama在Caffe再现GoogLeNet。我们感谢马利克的有益评论。感谢Wei Liu指出了SIFT流平均IU计算的一个问题，以及我们的频率加权平均IU公式的错误。

A. IU上限

本文在平均IU分割度量上取得了很好的性能，即使是在粗语义预测的情况下。为了更好地理解这个指标以及这种方法对它的限制，我们计算了不同规模预测的性能的近似上界。我们通过对真实图像进行降采样来实现这一点，然后再次对它们进行上采样，以模拟使用特定的降采样因子可获得的最佳结果。下表给出了不同下采样因子下PASCAL 2011 val子集上的平均IU。

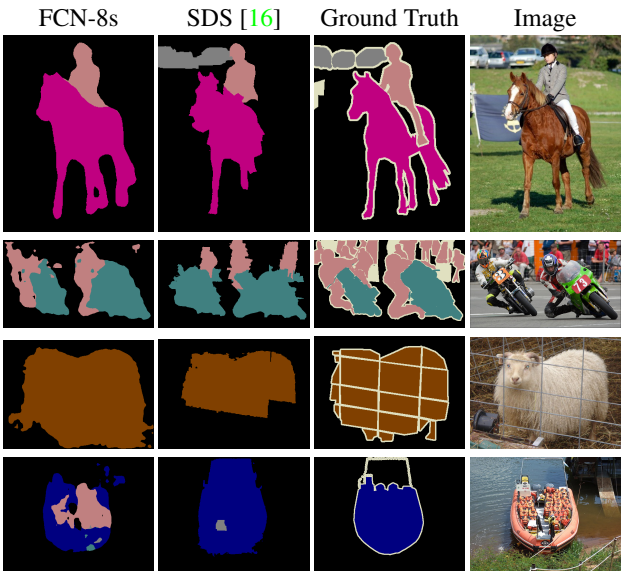

Figure 6. 全卷积分割网络在PASCAL上产生最先进的性能。左边一列显示了我们性能最高的网络FCN-8s的输出。第二个展示了Hariharan之前最先进的系统产生的分割*et al.* [16]。请注意恢复的精细结构(第一行)，分离密切相互作用的对象的能力(第二行)，以及对遮挡的鲁棒性(第三行)。第四行展示了一个失败案例:网络将船上的救生衣视为人。

factor	mean IU
128	50.9
64	73.3
32	86.1
16	92.8
8	96.4
4	98.5

像素完美预测显然不是实现平均IU远高于最先进水平的必要条件，相反，平均IU并不是精细尺度精度的良好度量。

B. 更多结果

进一步评估了FCN的语义分割。

PASCAL- context[26]提供PASCAL VOC 2010的全场景注释。虽然有400多个不同的类，但我们遵循[26]定义的59个类任务，选择最频繁的类。我们分别在训练集和val集上进行训练和评估。在表6中，我们将卷积特征屏蔽的联合对象+内容变化[3]进行了比较，这是这项任务上的最新技术。FCN-8s得分35.1平均IU，相对改善11%。

Table 6. PASCAL-Context的结果。*CFM*是[3]通过VGG网络进行卷积特征掩盖和分割追踪的最佳结果。O_2P为二阶池化方法[1]，如[26]勘误表所述。59类任务包括59个最常见的类，而33类任务由一个更容易识别的子集[26]组成。

59 class	pixel acc.	mean acc.	mean IU	f.w. IU
O_2P	-	-	18.1	-
CFM	-	-	31.5	-
FCN-32s	63.8	42.7	31.8	48.3
FCN-16s	65.7	46.2	34.8	50.7
FCN-8s	**65.9**	**46.5**	**35.1**	**51.0**
33 class				
O_2P	-	-	29.2	-
CFM	-	-	46.1	-
FCN-32s	69.8	65.1	50.4	54.9
FCN-16s	**71.8**	**68.0**	53.4	57.5
FCN-8s	**71.8**	67.6	**53.5**	**57.7**

变更日志

本文的arXiv版本保持更新，并附有更正和其他相关材料。下面给出了变化的简要历史。

v2 添加附录A给出平均IU的上限，添加附录B给出PASCAL-Context的结果。正确的PASCAL验证数(以前，一些val图像被包含在训练中)，SIFT流平均IU(使用了不恰当的严格度量)，以及频率加权平均IU公式中的错误。添加到模型的链接并更新时间编号，以反映改进的实现(这是公开可用的)。

References

[1] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 8

[2] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *NIPS*, pages 2852–2860, 2012. 1, 2, 4, 5

[3] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. *arXiv preprint arXiv:1412.1283*, 2014. 7, 8

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 1

[5] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 633–640. IEEE, 2013. 2

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014. 2

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html. 4

[8] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2013. 1, 2, 4, 5, 7

[9] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to SIFT. *CoRR*, abs/1405.5769, 2014. 1

[10] L. Florack, B. T. H. Romeny, M. Viergever, and J. Koenderink. The gaussian scale-space paradigm and the multiscale local jet. *International Journal of Computer Vision*, 18(1):61–75, 1996. 5

[11] Y. Ganin and V. Lempitsky. N^4-fields: Neural network nearest neighbor fields for image transforms. In *ACCV*, 2014. 1, 2, 5

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014. 1, 2, 6

[13] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *CVPR*, 2013. 6

[14] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*. Springer, 2014. 1, 2, 6, 7

[15] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011. 6

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 2, 4, 6, 7

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 1, 2

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 6

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 4

[20] Q. V. Le, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012. 3

[21] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to hand-written zip code recognition. In *Neural Computation*, 1989. 2

[22] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 1998. 5

[23] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):978–994, 2011. 7

[24] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *NIPS*, 2014. 1

[25] O. Matan, C. J. Burges, Y. LeCun, and J. S. Denker. Multi-digit recognition using a space displacement neural network. In *NIPS*, pages 488–495. Citeseer, 1991. 2

[26] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 891–898. IEEE, 2014. 7, 8

[27] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano. Toward automatic phenotyping of developing embryos from videos. *Image Processing, IEEE Transactions on*, 14(9):1360–1371, 2005. 1, 2, 4, 5

[28] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014. 1, 2, 4, 5, 7

[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 1, 2, 3

[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 6

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1, 2, 4

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 1, 2, 4

[33] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, pages 352–365. Springer, 2010. 7

[34] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 7

[35] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *CoRR*, abs/1406.2984, 2014. 2

[36] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1058–1066, 2013. 3

[37] R. Wolf and J. C. Platt. Postal address block location using a convolutional locator network. *Advances in Neural Information Processing Systems*, pages 745–745, 1994. 2

[38] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014. 1

[39] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014. 1