Fully Convolutional Networks for Semantic Segmentation

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，能够生成特征的层次结构。我们展示了单独训练的端到端的像素到像素的卷积网络在语义分割方面超过了最先进的水平。我们的关键洞察是构建“完全卷积”网络，该网络接受任意大小的输入并产生相同大小的输出，具有高效的推理和学习能力。我们定义和详细说明了*fully convolutional network*空间，解释了它们在空间密集预测任务中的应用，并与先前的模型进行了联系。我们将当代分类网络（*AlexNet [19]*、*VGG net [31]* 和 *GoogLeNet [32]*）调整为*fully convolutional network*，并通过微调*[4]*转移它们学到的表示到分割任务。然后，我们定义了一种新颖的架构，将来自深层粗糙层的语义信息与来自浅层细粒度层的外观信息相结合，以产生准确和详细的分割。我们的*fully convolutional network*在*PASCAL VOC*（*2012*年相对改进*20%*至*62.2%*平均*IU*）、*NYUDv2* 和*SIFT Flow* 的分割任务中取得了最先进的结果，同时对于典型图像，推理时间不到一秒的五分之一。

1. Introduction

卷积网络正推动识别技术的发展。卷积网络不仅在整体图像分类方面有所改进（[19, 31, 32]），还在具有结构化输出的局部任务上取得进展。这些包括边界框物体检测（[29, 12, 17]）、部位和关键点预测（[39, 24]）以及局部对应（[24, 9]）的进展。

从粗糙到精细推断的递进自然下一步是在每个像素点进行预测。以往的方法已经将卷积网络用于语义分割（[27, 2, 8, 28, 16, 14, 11]），其中每个像素被标记为其所属对象或区域的类别，但存在本文所解决的缺陷。

我们展示了一个名为（FCN）的端到端训练像素到像素的语义分割模型超越了最先进技术而无需额外的机器。据我们所知，这是第一个端到端训练 FCN模型（1）用于像素级预测且（2）从监督预训练中。现有网络的全卷积版本能够从任意大小的输入中预测密集输出。训练和推断都通过密集前馈计算和反向传播一次处理整个图像。网络内的上采样层实现了像素级预测和在具有子采样池化的网络中进行学习。

这种方法在性能上高效，既在渐近意义上也绝对如此，并且避免了其他作品中的复杂性。基于块的训练很常见（[27, 2, 8, 28, 11]），但缺乏全卷积训练的效率。我们的方法不采用预处理和后处理的复杂步骤，包括超像素（[8, 16]）、提议（[16, 14]）或基于随机场或局部分类器的事后精炼处理（[8, 16]）。我们的模型通过重新将分类网络解释为全卷积网络并从其学到的表示微调，将最近在分类方面取得的成功转移到密集预测中。相比之下，以前的作品应用小型无监督预训练的卷积网络进行。

语义分割面临语义和位置之间固有的张力：全局信息解决“什么”，而局部信息解决“在哪里”。深度特征层次结构联合编码位置和语义，形成一个从局部到全局的金字塔。我们在第 5.2 节中定义了一种新颖的“跳跃”架构，将深层、粗糙的语义信息和浅层、细致的外观信息结合在一起（见图 3）。

在下一节中，我们将回顾深度分类网络、FCNs 以

*Authors contributed equally

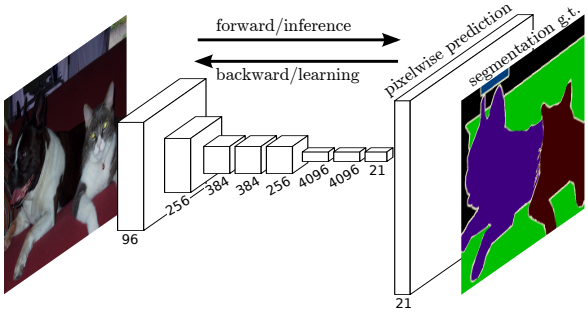

图 1. 全卷积网络可以高效地学习为像语义分割这样的每像素任务做出密集预测。

及最近使用 convnets 进行语义分割的相关工作。接下来的章节将解释 FCN 的设计和密集预测折衷，介绍我们的架构与网络内上采样和多层组合，并描述我们的实验框架。最后，我们展示了在 PASCAL VOC 2011-2、NYUDv2 和 SIFT Flow 上的最新成果。

2. Related work

我们的方法借鉴了深度神经网络在图像分类 [19, 31, 32] 和迁移学习 [4, 38] 领域的最新成功。迁移学习首先在各种视觉识别任务 [4, 38] 上展示出成功，然后扩展到检测以及在混合提案-分类器模型中的实例和语义分割 [12, 16, 14]。

我们现在重新设计和微调分类网络，以直接、密集地预测语义分割。我们对 FCNs 的空间进行了总结，并将之前的模型，无论是历史上的还是最近的，放置在这个框架中。

Fully convolutional networks 据我们所知，将 convnet 扩展到任意大小输入的想法最早出现在 Matan 等人的工作中 [25]，他们将经典的 LeNet [21] 扩展为识别数字字符串。由于他们的网络受限于一维输入字符串，Matan 等人使用维特比译码来获得他们的输出。Wolf 和 Platt [37] 将 convnet 输出扩展为检测邮政地址块四个角的二维地图上的检测得分。这两个历史性的工作均在检测中完全卷积进行推理和学习。 Ning 等人 [27] 定义了一个用于粗略多类别分割 *C. elegans* 组织的 convnet，具有全卷积推理。

在当前时代的多层网络中，全卷积运算也被广泛利用。 Sermanet 等人的滑动窗口检测 [29]、Pinheiro 和 Collobert 的语义分割 [28]，以及Eigen 等人的图像恢复 [5] 都使用全卷积推断。全卷积训练很少见，但Tompson 等人 [35] 使用它有效地学习端到端的部件检测器和姿态估计的空间模型，尽管他们没有详细阐述或分析这种方法。

另外，He 等人 [17] 抛弃分类网络的非卷积部分以构建特征提取器。他们结合提议和空间金字塔池化来产生一个定位的固定长度特征进行分类。虽然快速而有效，这种混合模型无法端到端学习。

使用 convnets 进行密集预测 最近的几项工作将 convnets 应用于密集预测问题，包括 Ning 等人的语义分割 [27]，Farabet 等人 [8] 和 Pinheiro 和 Collobert 的语义分割；Ciresan 等人 [2] 的电子显微镜边界预测，以及 Ganin 和 Lempitsky 的混合神经网络/最近邻模型在自然图像边界预测中的应用；以及 Eigen 等人 [5, 6] 的图像恢复和深度估计。这些方法的共同要素包括

3. Introduction

Our study aims to investigate the relationship between socioeconomic status and health outcomes in urban populations. Previous research has shown that individuals of higher socioeconomic status tend to have better health outcomes compared to those of lower socioeconomic status. However, the mechanisms underlying this relationship are complex and multifaceted. By exploring various factors such as access to healthcare, living environment, and lifestyle choices, we aim to provide a comprehensive understanding of how socioeconomic status influences health in urban settings. Moreover, we seek to identify potential interventions that can help mitigate health disparities related to socioeconomic status.限制容量和感受野较小的模型; 基于块训练 [27, 2, 8, 28, 11]; 通过超像素投影、随机场正则化、滤波或局部分类的后处理 [8, 2, 11]； 输入偏移和输出交织是由OverFeat [29] 引入的密集输出 [28, 11] 的技术。 多尺度金字塔处理[8, 28, 11]; 饱和的 tanh 非线性变换 [8, 5, 28]; 和 ensembles [2, 11],

然而，我们的方法不使用这种机制。然而，我们从FCN生成的角度研究了基于补丁的训练4.4和“移位和拼接”密集输出4.2。我们还讨论了网络内部的上采样4.3，其中由Eigen等人[6]提出的全连接预测是一种特

殊情况。

与这些现有方法不同，我们改编和扩展了深度分类架构，使用图像分类作为监督预训练，并通过全卷积微调，以从整个图像输入和整个图像地面真相中简单高效地学习。

Hariharan等人[16]和Gupta等人[14]同样将深度分类网络适应语义分割，但是他们在混合提议-分类器模型中这样做。这些方法通过对检测、语义分割和实例分割进行边界框和/或区域提议的采样，对R-CNN系统[12]进行微调。这两种方法均不是端到端学习。

他们在PASCAL VOC分割和NYUDv2分割上获得了最先进的结果，因此我们直接将我们独立的端到端FCN与他们的语义分割结果在第6节中进行了比较。

4. Fully convolutional networks

在convnet中，每个数据层都是大小为$h \times w \times d$的三维数组，这里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$h \times w$，具有d个彩色通道。更高层中的位置对应于它们与图像连接的位置，这些位置被称为它们的接受域。

Convnet建立在平移不变性上。它们的基本组件（卷积、池化和激活函数）在本地输入区域上操作，并且仅依赖于相对空间坐标。用$\mathbf{x}_{ij}$表示特定图层中位置(i, j)处的数据向量，用$\mathbf{y}_{ij}$表示接下来的图层，这些函数计算输出$\mathbf{y}_{ij}$通过

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

其中k被称为核大小，s是步幅或下采样因子，f_{ks}确定了层类型：卷积或平均池化的矩阵乘法，最大池化的空间最大值，激活函数的逐元素非线性等，其他类型的层类似。这个函数形式在组合下是保持不变的，核大小和步幅遵循变换规则。

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

虽然一个一般的深度网络计算了一个一般的非线性函数，但一个只含有这种形式层的网络计算了一个非线性滤波器，我们称之为深度滤波器或完全卷积网络。一个FCN自然地作用于任何尺寸的输入，并产生相应维度的（可能重新采样的）输出。

由一个FCN成的实值损失函数定义了一个任务。如果损失函数是在最终层的空间维度上的求和，

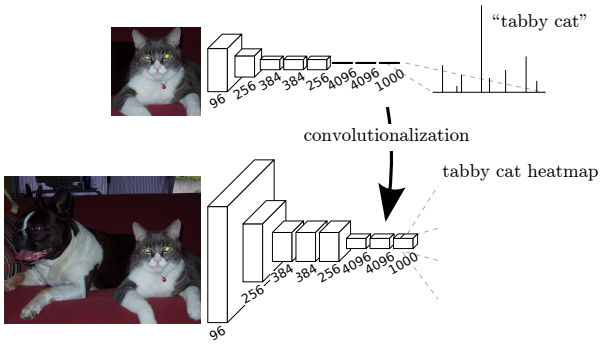

图 2. 将全连接层转换为卷积层能够使分类网络输出热力图。在模型中添加层和空间损失（如图1所示）会产生一个高效的端到端密集学习机制。

$\ell(\mathbf{x}; \theta) = \sum_{ij} \ell'(\mathbf{x}_{ij}; \theta)$，它的梯度将是对每个空间分量的梯度之和。因此，在整个图像上计算的ℓ的随机梯度下降将和在ℓ'上的随机梯度下降相同，将最终层的所有接收域视为一个小批量。

当这些接收域有重叠时，前向计算和反向传播在整个图像上逐层计算要比独立地逐块地计算更有效率。

接下来我们将解释如何将分类网络转换为产生粗糙输出地图的完全卷积网络。对于像素级预测，我们需要将这些粗糙输出与像素连接起来。第 4.2 节描述了 OverFeat [29] 为此引入的一个技巧。通过重新解释为一个等效的网络修改，我们深入理解了这个技巧。作为一个高效、有效的替代方法，我们在第 4.3 节介绍了上采样的反卷积层。在第 4.4 节中，我们考虑通过分块采样进行训练，并在第 5.3 节中提供证据表明我们的整个图像训练更快且同样有效。

4.1. Adapting classifiers for dense prediction

典型的识别网络，包括 LeNet[21]、AlexNet[19] 以及其更深层的后继者[31, 32]，表面上接受固定大小的输入并产生非空间化的输出。这些网络的全连接层具有固定的维度，并丢弃空间坐标。然而，这些全连接层也可以被视为卷积，其核覆盖整个输入区域。这样做将它们转变为可接受任何大小输入并输出分类地图的 fully convolutional network。该转换在图 2 中有所展示。（相比之下，例如 Le 等人的非卷积网络 [20] 不具备这种能力。）

此外，虽然得到的地图等同于在特定输入补丁上评估原网络的结果，但计算高度分摊在这些补丁的重

叠区域。例如，虽然 AlexNet 在典型 GPU 上需要 1.2 毫秒来生成一个 227×227 图像的分类分数，但全卷积版本在一个 500×500 图像上需要 22 毫秒来生成一个 10×10 网格的输出，比朴素方法快了超过 5 倍[1]。

这些卷积网络模型的空间输出地图使其成为密集问题如语义分割的自然选择。由于每个输出单元的地面真实 (ground truth) 可用，前向和后向传播都是直截了当的，并且都利用了卷积的固有计算效率（以及积极优化）。

以 AlexNet 为例，对于单个图像，相应的反向传播时间为 2.4 毫秒，对于一个全卷积 10×10 输出地图需要 37 毫秒，结果速度提升类似于前向传播。该密集反向传播在图 1 中有所展示。

尽管我们将分类网络重新解释为全卷积，可为任何大小输入生成输出地图，但输出维度通常会通过子采样减小。分类网络进行子采样以保持滤波器的小尺寸和合理的计算要求。这会使得完全卷积版本的输出变得更粗糙，将其输出从输入大小缩减的程度等同于输出单元感受野的像素步长的因子。

4.2. Shift-and-stitch is filter rarefaction

输入迁移和输出交错是一种技巧，它在没有插值的情况下从粗糙的输出中产生密集的预测，由OverFeat [29]引入。如果输出被缩减了一个因子f，则输入被移动（通过左侧和顶部填充）向右移动x个像素，向下移动y个像素，对于每个$(x, y) \in \{0, \ldots, f-1\} \times \{0, \ldots, f-1\}$的值进行一次移动。这$f^2$个输入分别通过卷积网络运行，输出交错以使预测对应于它们感受野中心的像素。

仅更改convnet的滤波器和层步长可以产生与这种移位和拼接技巧相同的输出。考虑一个具有输入步长s的层（卷积或池化），以及具有滤波器权重f_{ij}的后续卷积层（这里省略了特征维度，这里不相关）。将较低层的输入步幅设置为1会使其输出上采样s倍，就像移位和拼接一样。然而，将原始滤波器与上采样输出进行卷积不会产生与技巧相同的结果，因为原始滤波器只看到其（现在上采样的）输入的缩小部分。要复制这个技巧，通过扩大滤波器稀疏。

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

用i和j为零基准。复现这一技巧的完整网络输出需要层层重复这种滤波器扩大，直到所有的下采样被去除。

简单地减少网络中的下采样是一种权衡：滤波器看到更精细的信息，但具有更小的感受野且计算时间更长。我们已经看到，位移拼接技巧是另一种权衡：输出变得更密集，而不减小滤波器的感受野大小，但这些滤波器被禁止访问比它们原始设计更细的信息。

尽管我们已经进行了位移拼接的初步实验，但我们没有在我们的模型中使用它。我们发现通过上采样学习，特别是与后面描述的跳层融合相结合，更加有效和高效。

4.3. Upsampling is backwards strided convolution

另一种将粗糙的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线性映射从最近的四个输入计算每个输出 y_{ij}，该映射仅取决于输入和输出单元的相对位置。

在某种意义上，具有因子 f 的上采样是具有 分数输入步幅 $1/f$ 的卷积。只要 f 是整数，因此一种自然的上采样方法是使用*反卷积*（有时称为*反卷积*）以 输出 步幅为 f。这样的操作很容易实现，因为它只是颠倒了卷积的正向和反向传递。因此，通过从像素级损失反向传播实现端到端学习的网络中执行上采样。

请注意，在该层中的反卷积滤波器不需要固定（如双线性上采样），而可以进行学习。甚至一堆反卷积层和激活函数可以学习非线性上采样。

在我们的实验中，我们发现在网络内进行上采样对于学习密集预测是快速和有效的。我们最佳的分割架构使用这些层学习用于在第 5.2 节中精炼预测的上采样。

4.4. Patchwise training is loss sampling

在随机优化中，梯度计算受训练分布的驱动。无论是分块训练还是完全卷积训练都可以被配置为产生任意的分布，但它们的相对计算效率取决于重叠和小批量大小。整个图像完全卷积训练与分块训练相同，其中每个批次包含图像（或图像集合）下方单元的所有感知域。虽然这比均匀采样补丁更有效，但会减

[1]假设单图像输入的高效批处理。单独处理一个图像的分类分数需要 5.4 毫秒,几乎比全卷积版本慢了近 25 倍。

少可能的批次数。然而，可以轻松恢复图像内的补丁的随机选择。将损失限制在其空间项的随机抽样子集上（或者等效地在输出和损失之间应用DropConnect掩码[36]），会在梯度计算中排除补丁。

如果保留的补丁仍然有显着的重叠，完全卷积计算仍然会加快训练速度。如果梯度在多次后向传播中被累积，批次可以包含来自多个图像的补丁。[2]

在分块训练中的采样可以纠正类别不平衡[27, 8, 2]并减轻密集补丁的空间相关性[28, 16]。在完全卷积训练中，通过调整损失权重可以实现类别平衡，并且损失采样可用于解决空间相关性。

我们在第5.3节中探讨了采样训练，并没有发现它能够为密集预测提供更快或更好的收敛。整个图像训练既有效又高效。

5. Segmentation Architecture

我们将ILSVRC分类器转化为FCN，并通过网络内插法和逐像素损失来增强它们，以用于稠密预测。我们通过微调来进行分割训练。接下来，我们构建了一种新颖的跳跃架构，将粗糙的语义信息和局部的外观信息相结合，以改进预测。

对于这项研究，我们在PASCAL VOC 2011分割挑战赛上进行训练和验证[7]。我们采用逐像素多项式逻辑损失进行训练，并使用平均像素交集联合的标准指标进行验证，将平均值计算在所有类别上，包括背景。训练时忽略地面实况中被屏蔽出来的像素（作为模棱或困难的像素）。

5.1. From classifier to dense FCN

我们首先按照第 4 节的方法将经过验证的分类架构转换为卷积模式。我们考虑了赢得ILSVRC12竞赛的AlexNet架构（使用公开可获得的`CaffeNet`参考模型）[19]，以及在ILSVRC14竞赛中表现出色的VGG网络 [31] 和 GoogLeNet（由于没有公开可获得的GoogLeNet版本，我们使用自己重新实现的版本。我们的版本经过较少的数据增强训练，获得了68.5%的ILSVRC top-1准确率和88.4%的top-5准确率） [32]。我们选择了VGG 16层网络（使用来自Caffe模型库的公开可获得版本），我们发现在这项任务中，它等效于19层网络。对于GoogLeNet，我们仅使用最终的损失层，并通过丢弃最终的平均池化层来提高性能。我们通过舍弃最终的分类器层来截断每个网络，并将所有全连接层转换为卷积层。我们在每个粗输出位置追加一个具有21个通道维度的1 × 1卷积，用于预测每个PASCAL类（包括背景）在每个粗输出位置的分数，之后通过反卷积层将粗输出双线性上采样为像素密集输出，如第 4.3 节所述。表 1 比较了初步验证结果以及每个网络的基本特性。我们报告了在固定学习率（至少175个时期）下达到收敛后的最佳结果。

表 1. 我们将三个分类卷积网络调整并扩展到分割任务中。我们通过在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输入进行20次试验取平均）来比较性能。我们详细介绍了适应的网络体系结构，包括密集预测的参数层数、输出单元的感受野大小以及网络内最粗的步长。（这些数字代表在固定学习率下获得的最佳性能，并非可能的最佳性能。）

	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

从分类到分割的微调为每个网络提供了合理的预测。即使最差的模型也达到了最优性能的约75%。分割配备的VGG网络（FCN-VGG16）在验证集上的均值IU已经达到56.0，在测试集上为52.6 [16]，似乎已经达到了最先进水平。在额外数据集上训练使性能提升至验证集上的一个子集的59.4均值IU（见脚注[7]）。有关训练细节请参见第 5.3 节。

尽管具有相似的分类准确度，我们的GoogLeNet实现并不符合这一分割结果。

5.2. Combining what and where

我们为分割定义了一个新的全卷积网络（FCN），它结合了特征层次结构并提升了输出的空间精度。见图3。

尽管完全卷积的分类器可以通过微调实现分割，

[2]请注意,并非以这种方式包含每个可能的补丁,因为最终层单元的感知域位于一个固定的步进网格上。然而,通过将图像向左和向下移动一个随机值,直到步进,可以恢复对所有可能补丁的随机选择。

如5.1所示，并且在标准度量上得分很高，但它们的输出粗糙（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规模。

为了解决这个问题，我们添加了连接，将最终预测层与细粒度步长的下层相结合。这将线型拓扑结构转变为DAG，边缘跳过从较低层到高层（见图3）。由于它们看到的像素更少，较细粒度的预测应该需要更少的层，因此从更浅的网络输出中制作它们是合理的。结合细层和粗层让模型做出尊重全局结构的局部预测。类比于Florack等人的多尺度局部射线[10]，我们将非线性局部特征层次称为深射线。

我们首先将输出步长减半，通过从16像素步长层进行预测。我们在pool4上方添加一个1 × 1卷积层以产生额外的类预测。我们将此输出与在conv7（卷积化的fc7）顶部计算的预测相结合，步长为32，方法是添加一个2×上采样层，并将两个预测相加（见图3）。我们初始化2×上采样为双线性插值，但允许参数像在第4.3节中描述的那样学习。最后，步长16的预测被上采样回图像。我们将这个网络称为FCN-16s。FCN-16s是端到端学习的，初始化为最后一个更粗糙的网络的参数，我们现在称之为FCN-32s。作用在pool4上的新参数是零初始化的，以使网络从未修改的预测开始。学习率减小了100倍。

学习这个跳跃网络提高了验证集上的表现，平均IU达到62.4，比提高了3.0。图4显示了输出中细微结构的改善。我们将这种融合与仅从pool4层学习（导致性能不佳）以及仅降低学习速率而不添加额外连接（结果是微不足道的性能提升，且不会改善输出质量）进行了比较。

表 2. 在PASCAL VOC2011验证集的子集上比较了跳跃FCNs[7]。学习是端到端的，除了FCN-32s-fixed外，仅对最后一层进行微调。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**

我们继续这种方式，通过将pool3的预测与从pool4和conv7融合预测的2×上采样融合而来，构建网络FCN-8s。我们对平均IU进行轻微额外改进，达到了62.7，并发现输出的平滑度和细节略有改进。此时，我们的融合改进已经达到了收益递减点，无论是从强调大尺度正确性的IU度量的角度，还是从图4中可见的改进程度，因此我们不再融合更低层。

其他手段的细化 减少池化层的步幅是获得更精细预测的最直接的方法。然而，对于我们基于VGG16的网络来说，这样做存在问题。将pool5层的步幅设置为1需要我们的卷积化fc6层的核大小为14 × 14，以保持其感受野大小。除了它们的计算成本外，我们发现很难学习这么大的滤波器。我们尝试重新设计pool5上方的层，使用更小的滤波器，但未能取得可比性能；一个可能的解释是在上层使用ImageNet训练的权重进行初始化很重要。

获得更细粒度预测的另一种方法是使用第4.2节中描述的移位和拼接技巧。在有限的实验中，我们发现

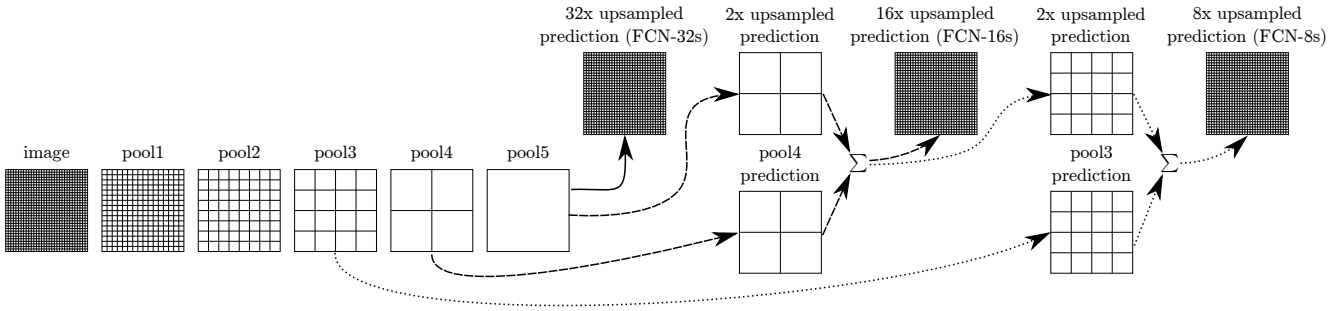

图 3. 我们的DAG网络学习将粗糙的高层信息与细致的低层信息相结合。层被显示为网格，展现相对空间粗细度。仅显示池化和预测层；中间的卷积层（包括我们转换的全连接层）被省略。实线（FCN-32s）：我们单流网络，在第5.1节中描述，将步长为32的预测一次性放大到像素级别。虚线（FCN-16s）：结合来自最后一层和pool4层的预测，在步长为16时，让我们的网络预测更精细的细节，同时保留高级语义信息。虚线（FCN-8s）：从pool3获得额外的预测，在步长为8时，提供更进一步的精度。

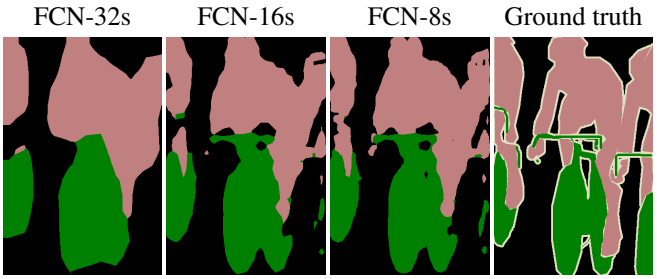

图 4. 通过融合具有不同步幅层的信息，改进完全卷积网络有助于提升细分的细节。前三幅图展示了我们32、16和8像素步幅网络的输出（详见图3）。

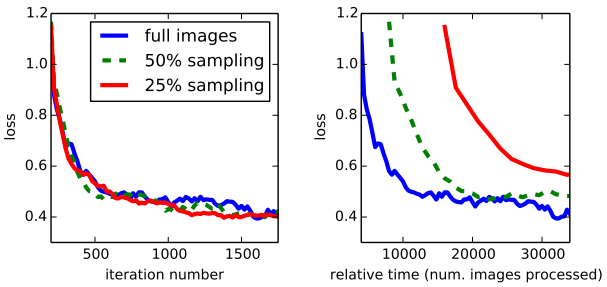

图 5. 在整张图像上训练与采样补丁一样有效，但由于更有效地利用数据，导致更快（墙上时间）的收敛。左边显示了在固定预期批处理大小的情况下，采样对收敛速率的影响，而右边则根据相对墙上时间绘制了相同的情况。

该方法的成本与改进比例较采用层融合要差。

5.3. Experimental framework

优化 我们使用带动量的SGD进行训练。对于FCN-AlexNet、FCN-VGG16和FCN-GoogLeNet，我们选择了迭代搜索确定的固定学习率10^{-3}、10^{-4}和5^{-5}，并使用20张图像的小批量大小。我们使用动量0.9，权重衰减分别为5^{-4}或2^{-4}，并且将偏置的学习率加倍，尽管我们发现对于这些参数训练不敏感（但对学习率敏感）。我们将类别评分卷积层初始化为零，发现随机初始化既没有提高性能也没有加快收敛速度。我们在原始分类器网络中用到了Dropout。

微调 我们通过整个网络进行反向传播对所有层进行微调。仅微调输出分类器仅能得到完全微调性能的70%，如表2所示。从头开始训练考虑到所需学习基础分类网络的时间不切实际。（请注意，VGG网络是分阶段训练的，而我们是从完整的16层版本初始化的。）对于粗糙的FCN-32s版本，微调需要单个GPU三天时间，升级到FCN-16s和FCN-8s版本各需要一天左右。

补丁采样 如4.4节所述，我们的全图像训练有效地将每个图像批量化为大型重叠补丁的规则网格。相比之下，先前的工作在整个数据集上随机采样补丁，可能导致更高方差的批次，有可能加快收敛速度。我们通过空间采样损失以前述方式进行独立选择，以某种概率$1-p$忽略每个最终层单元。为避免改变有效批量大小，我们同时将每批图像的数量增加p倍。需要注意的是，由于卷积的高效性，这种拒绝采样方式对于足够大的p值仍比基于补丁的训练更快（例如，根据4.1节中的数据，至少对于$p > 0.2$）。图5展示了这种采样方式对收敛的影响。我们发现与整图像训练相比，采样对收敛速度没有显著影响，但由于需要考虑更多图像数量，所需时间显著增加。因此，在其他实验中，我们选择没有采样的整个图像训练。

类别平衡 全卷积训练可以通过加权或采样损失来平衡类别。虽然我们的标签稍微不平衡（大约3/4为背景），但我们发现类别平衡是不必要的。

密集预测 得分通过网络内的反卷积层上采样到输入尺寸。最终层的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，然后学习。不使用Shift-and-stitch（第4.2节）或相应的滤波稀疏技术。

数据增强 我们尝试通过随机镜像和将图像随机平移最多32像素（预测的最粗粒度尺度）来增强训练数据。但这并没有明显改善。

更多训练数据 我们在表1中使用了PASCAL VOC 2011分割挑战训练集，其中包含1112张图像的标签。Hariharan等人[15]为8498张PASCAL训练图像收集了标签，这是用于训练先前最先进系统SDS的数据集[16]。这些训练数据改进了FCN-VGG16的验证分数[3]，将平均IU提高了3.4个百分点至59.4。

实现 所有模型均使用Caffe[18]在单个NVIDIA Tesla K40c上进行训练和测试。模型和代码将在发表时开源发布。

[3]在PASCAL VOC 2011验证集中包含了来自[15]的训练图像，因此我们在不相交的736张图像集上验证。本文早期版本错误地在整个验证集上评估。

6. Results

我们在语义分割和场景解析方面测试了我们的FCN，探索了PASCAL VOC、NYUDv2和SIFT Flow。尽管这些任务在历史上区分了对象和区域，但我们将它们统一视为像素预测。我们评估了我们的FCN跳过连接架构[4]在每个数据集上的表现，然后将其扩展到NYUDv2的多模态输入以及SIFT Flow的语义和几何标签的多任务预测。

评估指标 我们报告了普通语义分割和场景解析评估中的四个指标，这些指标是关于像素准确度和区域交并比的变量。设n_{ij}是预测为属于类别j的类别i像素数，其中有n_{cl}个不同类别，$t_i = \sum_j n_{ij}$表示类别i的像素总数。我们计算：

- 像素精确度：$\sum_i n_{ii} / \sum_i t_i$
- 平均准确率：$(1/n_{\text{cl}}) \sum_i n_{ii}/t_i$
- 均值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，并将其与先前的最新技术SDS [16]以及著名的R-CNN [12]进行了比较。我们在平均IU上取得了相对较大的20%优势。推理时间降低了114×（仅考虑convnet，不考虑提案和精炼）或286×（总体）。

表 3. 我们的全卷积网络在PASCAL VOC 2011和2012测试集上相对于最先进技术有20%的改进，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。它包含1449个RGB-D图像，像素级标签已由Gupta等人 [13] 合并为40类语义分割任务。我们报告了在795个训练图像和654个测试图像的标准拆分上的结果。（注意：所有模型选择均在PASCAL 2011 val上执行。）表 4 给出了我们模型在几个变种中的性能。

[4]我们的模型和代码公开可用于https://github.com/BVLC/caffe/wiki/Model-Zoo#fcn。

表 4. 在NYUDv2数据集上的结果。 *RGBD* 是RGB和深度通道在输入端的早期融合。 *HHA* 是[14]中深度嵌入的水平视差、离地高度和局部表面法线与推断重力方向之间的角度。 *RGB-HHA* 是联合训练的后期融合模型，将RGB和HHA的预测相加。

	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**

首先我们在RGB图像上训练我们未经修改的粗糙模型（FCN-32s）。为了添加深度信息，我们训练了一个升级为四通道RGB-D输入的模型（早期融合）。这几乎没有提供任何好处，可能是由于难以将有意义的梯度一直传播到模型的原因。在Gupta等人的成功之后 [14]，我们尝试了深度的三维 HHA 编码，仅在这些信息上训练网络，以及一个RGB和HHA的“后期融合”，其中两个网络的预测在最终层相加，并学习结果的双流网络。最后我们将这个后期融合网络升级为16步长版本。

SIFT Flow 是一个包含2688张图像的数据集，其中包含33个语义类别（“桥梁”，“山脉”，“太阳”），以及三个几何类别（“水平”，“垂直”和“天空”）的像素级标签。 FCN可以自然地学习同时预测两种标签的联合表示。我们学习了一个具有语义和几何预测层和损失的 FCN-16s 的双头版。通过在标准拆分为2488个训练图像和200个测试图像上计算的结果报告（见表5），[5] 显示了在两项任务上的最新成果。

7. Conclusion

模型是一类丰富的模型，其中现代分类卷积网络是一个特殊情况。认识到这一点，将这些分类网络扩展到分割，并通过多分辨率层组合改进架构，可以显著改善最新技术，同时简化和加快学习和推理过程。

致 谢 本工作部分得到

[5]三个SIFT Flow类别不在测试集中。我们跨越所有33个类别进行预测,但仅在我们的评估中包含实际出现在测试集中的类别。（本文的早期版本报告了一个更低的平均IU,其中包括所有出现或预测在评估中的类别。）

表 5. 使用类别分割（中）和几何分割（右）的SIFT Flow[10]结果。 Tighe [33] 是一种非参数化的传递方法。 Tighe 1是样本SVM，而2是SVM + MRF。 Farabet是在类别平衡样本（1）或自然频率样本（2）上训练的多尺度卷积网络。 Pinheiro是一个多尺度、循环的卷积网络，表示为RCNN$_3$ ($\circ^3$)。几何的度量标准是像素精度。

	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**

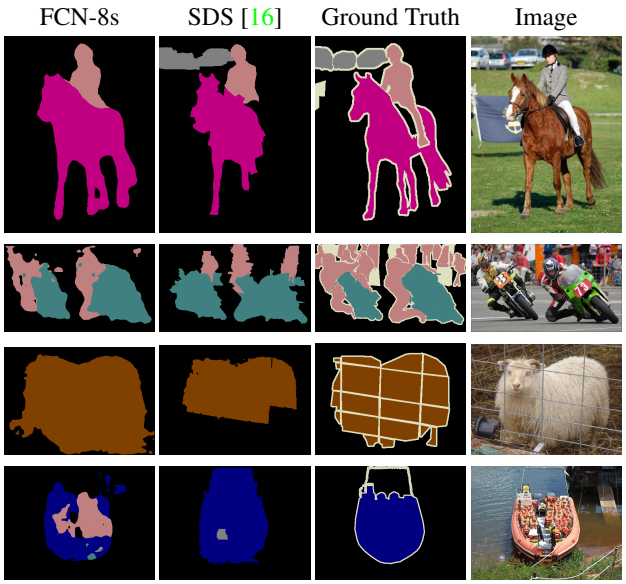

图 6. 完全卷积分割网络在PASCAL数据集上实现了最先进的性能。左列展示了我们性能最好的网络FCN-8s的输出。第二列展示了由Hariharan等人[16]提出的先前最先进系统生成的分割结果。注意细微结构的恢复（第一行）、分开紧密交互的对象的能力（第二行）以及对遮挡物的稳健性（第三行）。第四行展示了一个失败案例：网络将船上的救生衣误认为是人。

了DARPA的MSEE和SMISC计划的支持， NSF的奖励IIS-1427425， IIS-1212798， IIS-1116411， 以及NSF的GRFP， 丰田公司和伯克利视觉与学习中心。我们衷心感谢NVIDIA提供的GPU捐赠。我们感谢Bharath Hariharan和Saurabh Gupta提供的建议和数据集工具。我们感谢Sergio Guadarrama在Caffe中重新生成GoogLeNet。我们感谢Jitendra Malik提供的宝贵意见。感谢Wei Liu指出我们SIFT Flow均值IU计算中的问题和频率加权均值IU公式中的错误。

A. Upper Bounds on IU

在本文中，即使进行了粗略的语义预测，我们在平均IU分割指标上取得了良好的性能。为了更好地理解这一指标及其与该方法的限制，我们通过以不同比例进行预测来计算性能的近似上限。我们通过对地面真实图像进行降采样，然后再次上采样以模拟通过特定降采样因子获得的最佳结果。以下表格列出了在Pascal 2011 val子集上对不同降采样因子计算的平均IU。

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

像素级的完美预测显然并非必要，才能实现远高于最先进水平的平均IU值；相反，平均IU值并不是衡量细微尺度准确性的良好指标。

B. More Results

我们进一步评估我们的FCN用于语义分割。

PASCAL-Context [26] 提供了对PASCAL VOC 2010整个场景的注释。虽然有超过400个不同的类别，但我们遵循[26]定义的包括最常见类别的59类任务。我们分别在训练集和验证集上进行训练和评估。在表6中，我们与Convolutional Feature Masking的对象+材料联合变种进行比较[3]，后者是该任务的先前最先进技术。 FCN-8s获得了35.1的均值IU得分，相对改善了11%。

Changelog

这篇论文的arXiv版本会随时更新，包括纠正和添加额外相关材料。以下简要介绍了修改历史。

表 6. PASCAL-Context的结果。 *CFM*是[3]中通过卷积特征遮罩和VGG网络进行的分割追踪得到的最佳结果。 O_2P是报告在[26]的勘误中的第二阶池化方法[1]。 59类任务包括59个最常见的类别，而33类任务则是由[26]确定的更易于处理的子集。

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**

v2 添加附录 A，给出IU均值的上界，以及包含PASCAL-Context结果的附录 B with PASCAL-Context results. Correct PASCAL validation numbers (previously, some val images were included in train), SIFT Flow mean IU (which used an inappropriately strict metric), and an error in the frequency weighted mean IU formula. Add link to models and update timing numbers to reflect improved implementation (which is publicly available).

参考文献

[1] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 10

[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, 5

[3] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. *arXiv preprint arXiv:1412.1283*, 2014. 9, 10

[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, 2

[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. 5

[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, 5, 9

[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. 6

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

[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, 3, 8

[13] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *CVPR*, 2013. 8

[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, 3, 8

[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. 7

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 2, 3, 5, 7, 8, 9

[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. 7

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 3, 5

[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, 3

[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.

[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. 9

[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. 9, 10

[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, 5

[28] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014. 1, 2, 5, 9

[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, 4

[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 8

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1, 2, 3, 5

[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, 3, 5

[33] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, pages 352–365. Springer, 2010. 9

[34] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 9

[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. 5

[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. 2

[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