计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(12):203−210 [doi: 10.15888/j.cnki.csa.008830]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于语义对齐的小样本语义分割模型①

张　珉, 杨　娟, 汪荣贵

(合肥工业大学 计算机与信息学院, 合肥 230009)
通信作者: 张　珉, E-mail: 2019170994@mail.hfut.edu.cn

摘　要: 现实世界的物体图像往往存在较大的类内变化, 使用单一原型描述整个类别会导致语义模糊问题, 为此提出一种基于超像素的多原型生成模块, 利用多个原型分别表示物体的不同语义区域, 通过图神经网络在生成的多个原型间利用上下文信息执行原型校正以保证子原型的正交性. 为了获取到更准确的原型表示, 设计了一种基于Transformer 的语义对齐模块, 以挖掘查询图像特征和支持图像的背景特征中蕴含的语义信息, 此外还提出了一种多尺度特征融合结构, 引导模型关注同时出现在支持图像和查询图像中的特征, 提高对物体尺度变化的鲁棒性. 所提出的模型在 PASCAL-5^i 数据集上进行了实验, 与基线模型相比平均交并比提高了 6%.

关键词: 小样本语义分割; 度量学习; 原型学习; Transformer; 注意力机制; 语义对齐

引用格式: 张珉,杨娟,汪荣贵.基于语义对齐的小样本语义分割模型.计算机系统应用,2022,31(12):203–210. http://www.c-s-a.org.cn/1003-3254/8830.html

Few-shot Semantic Segmentation Model Based on Semantic Alignment

ZHANG Min, YANG Juan, WANG Rong-Gui

(School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China)

Abstract: Object images in the real world often have large intra-class variations, and thus using a single prototype to describe an entire category will lead to semantic ambiguity. Considering this, a multi-prototype generation module based on superpixels is proposed, which uses multiple prototypes to represent different semantic regions of objects and employs the context to correct prototypes among the generated prototypes by a graph neural network to ensure the orthogonality of the sub-prototypes. To obtain a more accurate prototype representation, a Transformer-based semantic alignment module is designed to mine the semantic information contained in the features of the query images and the background features of the supporting images. In addition, a multi-scale feature fusion structure is proposed to instruct the model to focus on features that appear in both the supporting images and the query images, which can improve the robustness to changes in object scales. The proposed model is tested on the PASCAL-5^i dataset, and the mean intersection over union (mIoU) is improved by 6% compared with that of the baseline model.

Key words: few-shot semantic segmentation; metric learning; prototype learning; Transformer; attention mechanism; semantic alignment

语义分割[1] 作为计算机视觉的基本任务之一, 有着广泛的应用场景[2,3], 然而全监督语义分割模型的训练依赖于带有人工标注的大规模数据集, 这些数据集的像素级标注获取成本高昂, 且训练出的模型的分割能力难以拓展到从未见过的新类, 这些缺陷严重制约了语义分割模型的实际应用. 为了应对这一挑战, 小样本语义分割应运而生. 小样本语义分割的目标是在仅有少量带注释的支持图像可用的条件下, 通过分割已

① 基金项目: 国家自然科学基金联合基金 (U20B2044)
收稿时间: 2022-03-20; 修改时间: 2022-04-14; 采用时间: 2022-04-29; csa 在线出版时间: 2022-08-19

知类别的对象来学习可迁移的知识, 从而将分割能力推广到未见过的新类别. Shaban 等人[4] 首先将小样本学习引入语义分割领域, 在仅有一张带有像素级注释的参考图像可用的条件下, 实现了对未知图像的分割, 提出的模型包括支持分支和查询分支, 分别用于处理带有像素级注释的参考图像和查询图像, 这种双分支结构也成为了后续小样本分割模型的主流架构. Dong 等人[5] 基于这种双分支架构在模型中引入了原型学习的思想, 利用支持图像生成图像类别的原型表示, 然后通过最近邻度量的方式对查询图像的每个像素执行分类, 最终得到分割结果. 为了更充分地利用支持图像中的语义信息, Wang 等人[6] 在原型学习的基础上提出了原型对齐正则化, 通过引入额外的监督信息提高模型的分割能力.

上述模型都采用了全局池化的方式从支持图像中提取类别原型表示, 这种方法简单易用, 且易于拓展到多个支持样本对的情形, 但是由于现实场景下物体的外观多变, 因此提取到的原型往往并不准确, 难以生成完整的类别表示. 为了解决这一问题, 本文提出一种基于语义对齐的小样本语义分割模型, 利用超像素在保留空间结构信息的前提下生成多个子原型, 随后使用图神经网络对生成的多个原型执行语义校正, 从而消除外观相似导致的语义混淆. 为了进一步利用查询图像和支持图像上的语义信息, 引入了基于 Transformer 的语义对齐模块, 对任意支持原型和查询原型之间的关系进行建模, 利用上下文信息对原型进行迭代优化, 从而生成更完整、精确的类的语义表示. 为了将得到的类别原型传播到查询特征中, 本文设计了一种基于相似度度量的原型分配模块, 以一种类别无关的方式执行原型分配. 此外, 还提出了一种基于注意力的多尺度特征融合模块, 引导模型关注重点特征, 提高对不同尺度物体的适应能力.

1 问题定义

小样本语义分割模型普遍将数据集分解为若干个子任务, 每个子任务即为一个场景, 以场景为单位进行训练和测试. 数据集被切分为两个图像集合 D_{train} 和 D_{test}, 分别用于模型训练和测试, 且二者的图片类别之间不存在交叉. 以 k-shot 下的训练过程为例, 每次从 D_{train} 中选择 k 张图像 x^s 及其对应的标注 m^s 组成支持集合 $S=\left\{x_i^s, m_i^s\right\}_{i=1}^k$, 随后选择和 S 中的图像不重复的一张图像 x^q 及其对应的标注 m^q 组成查询集 $Q=\{x^q, m^q\}$, 支持集 S 和查询集 Q 共同组成一个场景, 其中标注 m^s 和 m^q 是分辨率与图像相同的二值矩阵, 图像中像素点对应的标注值为 1 时表示该像素为前景, 值为 0 时表示该像素为背景. 注意同一个场景里的支持图像和查询图像都属于同一类别. 在进行训练时, 首先将支持集 S 和查询集 Q 中的图像 x^q 输入到模型中, 模型通过支持集 S 中的图像及其对应的标注, 学习需要分割的区域, 然后对 x^q 进行分割, 最终将预测结果 $\tilde{m}^q$ 和 m^q 进行比较, 计算损失 $\mathcal{L}_{\text{seg}}$ 后进行梯度更新. 模型训练完成后, 测试过程与训练过程类似, 唯一不同的是支持集 S 和查询集 Q 都来自 D_{test}.

2 网络结构

本文提出的基于语义对齐的小样本语义分割模型框架如图 1 所示, 框架由以下模块构成: 原型生成模块、语义对齐模块、原型分配模块和解码器模块. 首先模型的骨干网络将输入的支持图像 $x^s \in \mathbb{R}^{H\times W\times 3}$ 和查询图像 $x^s \in \mathbb{R}^{H\times W\times 3}$ 分别映射为深度特征 $f^s \in \mathbb{R}^{H_f\times W_f\times C}$ 和 $f^q \in \mathbb{R}^{H_f\times W_f\times C}$, 随后将 f^s 与其对应的标注 $m^s \in \mathbb{R}^{H\times W}$ 相乘, 以得到支持图像的前景特征 $f^{s\text{-fore}} \in \mathbb{R}^{H_f\times W_f\times C}$. 将特征 $f^{s\text{-fore}}$ 和 f^q 分别输入原型生成模块, 模块利用超像素对图像中的相似区域进行合并, 得到多个表示物体不同语义区域的子原型 $p_i^s \in \mathbb{R}^C$ 和 $p_i^q \in \mathbb{R}^C$. 随后原型对齐模块对原型 p_i^{s+} 执行迭代优化, 获取到最终的原型 $\tilde{p}_i^{s+}$. 原型分配模块利用余弦相似度将原型 $\tilde{p}_i^{s+}$ 拼接为特征 $f^m \in \mathbb{R}^{H_f\times W_f\times C}$, 最终解码器融合特征 f^q 和 f^m, 执行多次卷积和上采样, 得到分割结果 $\tilde{m}^q$.

2.1 原型生成模块

原型生成模块的作用是生成对象的多个原型, 不同的原型分别表示对象不同的语义区域. 例如输入犬类图像特征, 则生成的多个原型分别描述犬类的头部、四肢、尾巴和身体. 此外由于本文使用了 Transformer 进行原型迭代优化, 之前的视觉 Transformer 模型中往往将图像切分为多个大小相同的图像块, 这种切分方式并没有考虑到图像中的结构关系, 而原型生成模块可以按照图像的语义关系生成多个 Token 输入到 Transformer 中, 每个 Token 专注于描述类别的一个子语义区域, 从而避免语义混淆的问题. 原型生成模块的结构如图 2 所示. 超像素可以基于图像中像素的相似性和空间邻近性将图像分割为多个区域. 基于对泛化能力和性能的考虑, 原型生成模块中使用无可学习参数的高效且无监督的线性迭代聚类 (simple linear iterative clustering, SLIC)[7] 算法对图像执行超像素分割.

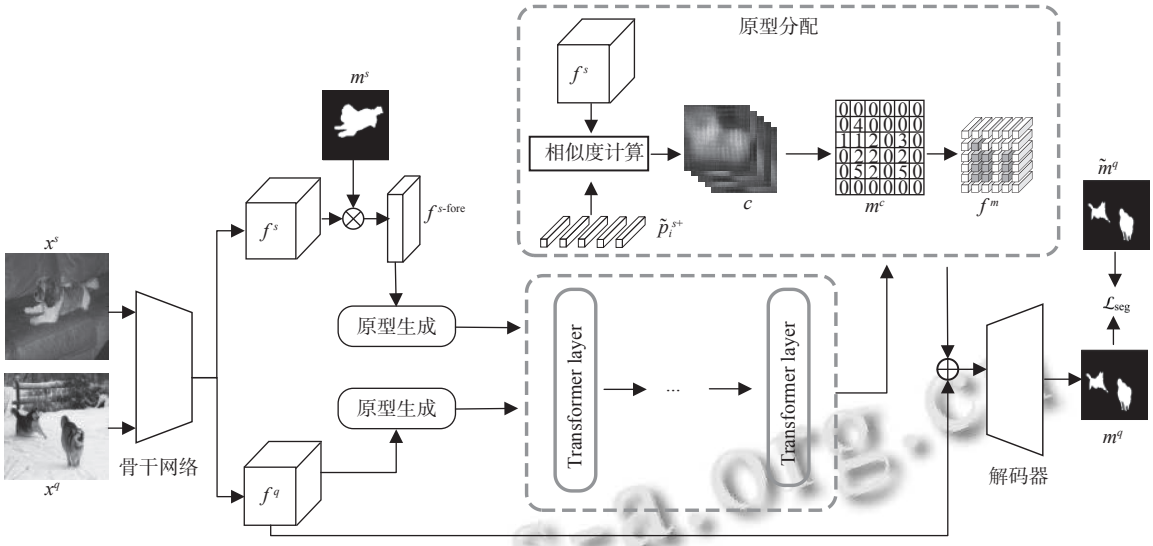

图 1 本文提出的模型

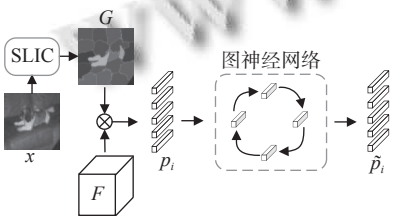

图 2 原型生成模块

对于支持图像$x^s \in \mathbb{R}^{H\times W\times 3}$, 由于相应的标注信息$m^s \in \mathbb{R}^{H\times W}$在训练和测试阶段都是可用的, 因此利用$m^s$分离出前景图像, 只对前景区域执行超像素分割, SLIC 算法将前景区域的所有像素划分到不同的集合中, 得到分割结果$G^s=\{G_1^s,\cdots,G_n^s\}$, 其中$n$表示支持图像前景区域的超像素个数, 即语义类别的数量. 分别对每个集合内的像素按照以下方式执行平均池化:

$$p_k^s=\frac{\sum_{i=1}^{H_f}\sum_{j=1}^{W_f} f_{i,j}^s\cdot g_k^s(i,j)}{\sum_{i=1}^{H_f}\sum_{j=1}^{W_f} g_k^s(i,j)} \tag{1}$$

其中, p_k^s表示第k组集合G_k^s的原型, g_k^s为分组结果指示器, 当坐标为i,j的像素属于集合G_k^s为 1, 否则为 0, 最终得到支持前景的初始原型$p^{s+}=\left\{p_i^{s+}\in\mathbb{R}^C\right\}_{i=1}^n$.

对于查询图像x^q, 由于相应的标注信息m^q在测试阶段是不可用的, 因此直接对x^q执行超像素, 生成的超像素分割结果为$G^q=\{G_1^q,\cdots,G_m^q\}$, 按照分割结果执行池化, 得到查询图像原型$p^q$, 其中即包括前景部分, 也包括背景部分.

超像素生成的分割结果并不总是与物体的语义结构完全对应, 因此模块中使用图神经网络利用上下文信息对生成的初始原型执行校正. 对于p^s和p^q, 分别以两个集合中的原型为节点建立全连通图, 使用以下方式在图中任意节点i和j间传播消息:

$$\tilde{p}_i=p_i+ReLU\left(\frac{1}{Z_i}\sum_{j=1\wedge j\neq i}^{n} d\left(p_i,p_j\right)Wp_j\right) \tag{2}$$

其中, p_i和$\tilde{p}_i$分别表示节点i更新前和更新后的值, Z_i表示节点i的正则化因子, $W\in\mathbb{R}^{C\times C}$是定义线性映射的权重矩阵, 对来自节点$j$的消息进行编码. d是用于节点间关系编码的相似度函数, 用以下方式计算:

$$d\left(p_i,p_j\right)=\frac{p_i^{\mathrm{T}}p_j}{\|p_i\|\left\|p_j\right\|} \tag{3}$$

2.2 Transformer 层

支持图像和查询图像都包含了同一类别的物体, 因此可以借助查询图像中的特征对提取出的原型进行完善. 此外, 之前的模型选择直接将查询图像中的背景特征抛弃, 但是前景和背景物体间往往存在语义关联, 例如船总是在水上. 为了充分挖掘查询特征和支持背景特征中蕴含的语义信息, 本文设计了一种基于 Transformer 的原型丰富模块, 通过多头自注意力机制对原型和特征之间的关系进行建模, 利用 Transformer 从特征中抽取局部信息和远程上下文信息对原型进行迭代优化, 以生成更准确的类别描述, Transformer 层结构如图 3 所示.

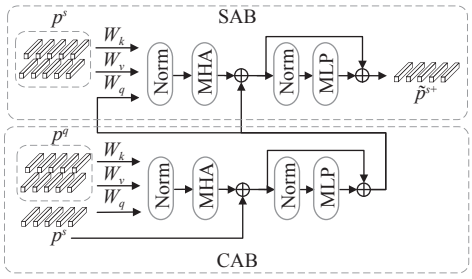

图 3 Transformer 层结构

Transformer 层由两个串联的 Transformer 块组成, 分别是交叉对齐模块 (cross alignment module, CAM) 和自对齐模块 (self alignment module, SAM). 在 CAM 中, 支持前景原型p^{s+}首先与查询原型p^q执行多头自注意力计算, 从而在任意支持原型和查询原型间建立联系, 自适应地挑选具有语义关联的特征对原型进行优化, 随后经过正则化输入到多层感知机中, 最终得到 CAM 的输出结果. p^{s+}和p^q按照以下方式进行自注意力计算:

$$\begin{cases} K = p^q W_k, \quad V = p^q W_v, \quad Q = p^{s+} W_q \\ Att\left(z^{l-1}\right) = z^{l-1} + Softmax\left(\dfrac{QK^{\mathrm{T}}}{\sqrt{c}}\right)V \end{cases} \tag{4}$$

其中, W_k、W_v和W_q为线性映射矩阵, 矩阵中均为可学习参数, Z^{l-1}为该层输入, 即p^{s+}, c为原型的维度. 自注意力运算过程中的维度变化如图 4 所示.

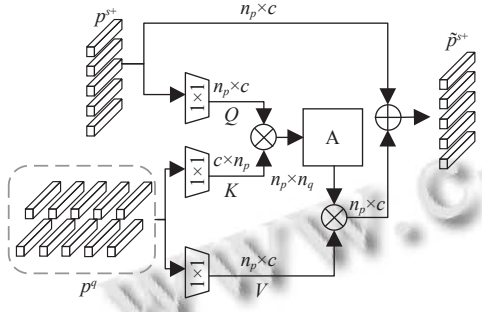

图 4 自注意力运算过程

模块中使用由多个自注意力组成的多头注意力, 多个自注意力的输出结果在通道方向上进行拼接, 即:

$$\begin{cases} Mha\left(z^{l-1}\right) = \left\{Att\left(z_1^{l-1}\right) \oplus \cdots \oplus Att\left(z_n^{l-1}\right)\right\} \\ z^l = Mha\left(z^{l-1}\right) + MLP\left(Att\left(z^{l-1}\right)\right) \end{cases} \tag{5}$$

其中, MLP为多层感知机, Z^l即为 CAM 的输出结果, 即经过查询原型优化后的支持原型p^{s+}. SAM 的作用是进一步挖掘支持图像中背景部分与前景部分的语义关联信息, 对支持图像中的上下文信息进行建模, SAM 的结构与 CAM 类似, 只是输入变成了p^{s+}和p^s, 使用转换矩阵对p^{s+}进行线性变换后再次将其作为Q, 而 K和V则是支持图像上的全体原型p^s. 最终, 在利用支持图像特征对p^{s+}进行优化后, 得到了 Transformer 层的输出$\tilde{p}^{s+}$. 本文在模型中使用了 4 个结构相同的串联的 Transformer 层, 上一层的输出即为下一层的输入, 从而对原型进行迭代优化. 关于 Transformer 层数量的取值在第 2.4 节进行了讨论.

2.3 原型分配模块

Zhang 等人[8] 指出, 深层特征具有更强的类特异性, 因此使用深层特征会对模型的泛化性能造成损害, 然而深层特征中具有更加丰富的语义信息, 这些语义信息对于指导分割过程有很大帮助, 为了同时利用中层特征和深层特征, Tian 等人[9] 使用中层特征作为主干特征, 使用深层特征进行相似度度量生成先验相似度图, 从而向模型提供先验知识, 但是由于模型中采用全局平均池化生成的单一原型进行相似度度量, 依旧存在语义模糊的问题, 由此生成的先验相似度图可能会对模型产生误导. 为了解决这一问题, 本文设计了原型分配模块, 利用多个原型分别与查询特征进行相似度图的计算, 结合多个相似度图的结果生成更加准确的先验相似度图, 将原型特征以一种类别无关的方式传播到查询特征中, 提高模型对新类的适应能力.

模块中首先利用生成的多个原型$\tilde{p}^s$分别对查询特征f^q执行相似度计算, 得到一组相似度图$c = \left\{c_i \in \mathbb{R}^{H_f \times W_f}\right\}_{i=1}^n$, c_i在坐标(x,y)处的相似度计算方式如下:

$$c_i^{x,y} = d\left(\tilde{p}_i^{s\text{-fore}}, f_{x,y}^q\right) \tag{6}$$

为了综合多个相似度图的结果, 将所有的相似度图相加后得到m^{pre}, 对其按照如下方式进行归一化, 最终得到先验相似度图$\tilde{m}^{\text{pre}}$, 其中η设置为 1E–7:

$$\begin{cases} m_{x,y}^{\text{pre}} = \displaystyle\sum_{i=1}^{n} c_i^{x,y} \\ \tilde{m}^{\text{pre}} = \dfrac{m^{\text{pre}} - \min\left(m^{\text{pre}}\right)}{\max\left(m^{\text{pre}}\right) - \min\left(m^{\text{pre}}\right) + \eta} \end{cases} \tag{7}$$

对每个像素位置, 按照c中的相似度图统计在该位置的激活值之和, 如果所有原型在该位置的相似度都很小, 说明该位置很可能是背景区域, 因此当激活值之和小于阈值θ时, 该位置不进行原型分配, 即赋值 0, 如果激活值之和高于阈值, 则选择相似度最高的原型进行分配, 即按照以下方式生成最终的分配结果:

$$m^c=\begin{cases}\arg\max\limits_{i\in\{1,\cdots,n\}} c_i^{x,y}, \sum\limits_{i=1}^{n} c_i^{x,y}>\theta\\ 0, \sum\limits_{i=1}^{n} c_i^{x,y}<\theta\end{cases}\\ \forall x\in\{1,2,\cdots,W_f\}, y\in\{1,2,\cdots,H_f\}\tag{8}$$

其中, H_f和W_f表示特征图的分辨率, 按照m^c中给出的分配结果对原型进行拼接, 最终得到原型特征f^m, 方法如下:

$$f^m=\begin{cases}\tilde{p}^s_{m^c_{x,y}}, m^c_{x,y}\neq 0\\ 0, m^c_{x,y}=0\end{cases}\\ \forall x\in\{1,2,\cdots,W_f\}, y\in\{1,2,\cdots,H_f\}\tag{9}$$

2.4 解码器

同一物体可能在不同的背景下出现, 作为分割目标的查询图像中包含背景部分, 背景的变化会对模型的分割过程造成干扰. 同时出现在查询图像和支持图像上的物体更有可能是分割目标, 基于这种朴素的假设, 本文设计了一种基于通道注意力和空间注意力的特征融合模块 (feature merge module, FMM), 以引导模型关注原型特征和查询特征中具有较高相似度的特征, 以捕获具有不变性的物体特征, 减少背景杂波对于分割的影响.

FMM 的结构如图 5 右侧所示, 对于输入的原型特征f^m和查询特征f^q, 经过 1×1 卷积后在空间方向上进行压缩, 分别得到向量$v^m\in\mathbb{R}^C$和$v^q\in\mathbb{R}^C$, 其中c为通道数. 对这两条向量计算 Hadamard 积, 两者在某个通道上的激活值越高则乘积越大, 随后向量经过全连接层和 Sigmoid 激活后分别对f^m和f^q进行通道加权. 经过通道维度的增强, 随后特征f^m和f^q在空间方向上再次进行压缩, 分别得到矩阵$M^m\in\mathbb{R}^{h\times w}$和$M^q\in\mathbb{R}^{h\times w}$, 同样对矩阵$M^m$和$M^q$计算 Hadamard 积, 以得到$M^{m-q}\in\mathbb{R}^{h\times w}$, 对$M^{m-q}$进行 3×3 卷积后分别对特征$f^m$和$f^q$在空间方向上进行加权, 从而进一步引导模型关注相似度较高的空间区域. 最终f^m和f^q经过空间方向和通道方向增强后与先验相似度图$\tilde{m}^{\text{pre}}$进行拼接, 经过 1×1 卷积运算后得到融合后的特征.

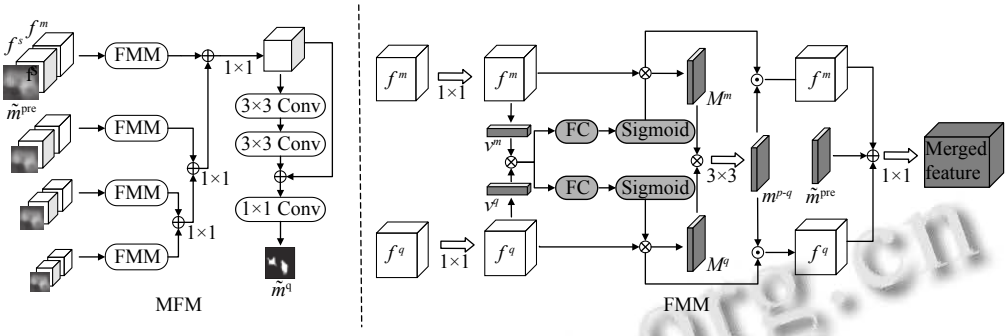

图 5 多尺度融合模块和特征融合模块

由于拍摄距离的变化, 即使是同一类别的物体, 在图像中的大小往往也有较大不同, 为了适应物体尺度的变化, 本文设计了一种多尺度模块, 可以融合多个尺度的特征从而生成更鲁棒的物体特征表示, 从而优化分割结果. 原始特征f^m和f^q首先经过空洞率为{1, 3, 5, 7}的空洞卷积得到 4 个尺度的特征, 分别利用 FMM 在通道和空间方向进行增强后, 对相邻尺度的输出结果进行连接和 1×1 卷积, 随后继续与更大尺度的输出结果进行连接和 1×1 卷积, 最终经过两个 3×3 卷积和残差连接后, 使用 1×1 卷积得到分割结果$\tilde{m}^q$, 对其计算交叉熵损失:

$$\mathcal{L}_{\text{seg}}=-\frac{1}{hw}\sum_{x=1}^{w}\sum_{y=1}^{h} m^q_{x,y}\log\left(\tilde{m}^q_{x,y}\right)\tag{10}$$

为了充分利用数据集, 同样也对支持图像进行预测, 使用支持图像的预测结果 m^s 作为监督信息, 计算交叉熵损失:

$$\mathcal{L}_{\text{aux}}=-\frac{1}{hw}\sum_{x=1}^{w}\sum_{y=1}^{h} m^s_{x,y}\log\left(\tilde{m}^s_{x,y}\right)\tag{11}$$

最终模型训练时的总体损失为:

$$\mathcal{L}_{\text{total}}=\mathcal{L}_{\text{seg}}+\mathcal{L}_{\text{aux}}\tag{12}$$

3 实验分析

3.1 实验数据及评价准则

为验证模型及各个模块的有效性, 本文在小样本语义分割两大公开数据集 PASCAL-5^i[10] 和 COCO[11]

进行实验, 采用平均交并比 (mIoU) 作为性能评估指标. PASCAL-5^i 数据集中包含 20 个图像类别, 这些类别被平均分为 4 个部分, 每个部分包含 5 个类别. 每次训练时取其中 3 个部分作为训练集, 剩下的 1 个部分用作测试集, 一共有 4 种分配方案和对应的 4 种测试结果, 取这 4 种分配方案的平均精度作为评估结果. 为了在公平的条件下与之前的模型进行性能对比, 测试条件的设置与之前的工作[12] 保持一致, 在测试时随机从测试集中抽取 5 000 个样本对进行测试, 对每个样本对上的测试结果取平均.

3.2 实验环境

模型基于 PyTorch 1.4 实现, 分别使用 VGG-16[13]、ResNet-50[14] 和 ResNet-101[14] 作为骨干网络在 ImageNet[15] 数据集上进行预训练, 其他参数使用默认的 PyTorch 设置进行初始化. 实验在 Linux 环境下进行, 模型在 NVIDIA GeForce RTX 2080Ti GPU 上训练 150 个轮回数 (epoch). 超像素生成的支持图像的原型数量为 10, 查询图像的原型数量为 25, 原型分配的阈值θ设置为 1, 原型匹配损失的权重λ设置为 0.2. 所有输入图像被随机裁剪为 473×473 大小, 以 0.5 的概率进行随机水平翻转和−10 度到 10 度之间的旋转. 算法采用随机梯度下降优化器 (SGD), 学习率为 0.1, 动量因子为 0.9.

3.3 与其他先进算法比较

实验在两种不同的骨干网络上分别对比了本文提出的模型与近年来其他先进的小样本语义分割模型的性能, 如表 1 所示, 本文提出的模型在 1-shot 和 5-shot 设置上都具有性能上的优势. 具体而言, 在使用 ResNet-50 作为骨干网络时, 本文提出的模型在 1-shot 和 5-shot 实验设置下分别超过目前先进模型 0.6% 和 1%. 在 5-shot 设置下, 由于有 5 对样本作为参照, 模型可以提取到更接近真实特征分布的多个原型 , 对查询图像的分割结果也更准确, 进一步放大了模型的优势.

表 1 模型在 PASCAL-5^i 数据集上的性能比较 (%)

骨干网络	模型	1-shot					5-shot				
		Fold-1	Fold-2	Fold-3	Fold-4	mean	Fold-1	Fold-2	Fold-3	Fold-4	mean
ResNet-50 Backbone	CANet[8]	52.5	65.9	51.3	51.9	55.4	55.5	67.8	51.9	53.2	57.1
	PGNet[16]	56.0	66.9	50.6	50.4	56.0	54.9	67.4	51.8	53.0	56.8
	PMM[17]	55.2	66.9	52.6	50.7	56.4	56.3	67.4	54.5	51.0	57.3
	PPNet[18]	48.6	60.6	55.7	46.5	52.3	58.9	68.3	66.8	58.0	63.0
	PFEnet[9]	61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9
	ASR[19]	55.23	70.36	53.38	53.66	58.16	59.38	71.85	56.87	55.72	60.96
	本文	61.9	69.7	54.8	54.3	61.4	66.3	70.6	60.3	58.8	64.0
ResNet-101 Backbone	DAN[20]	57.7	68.6	57.8	51.6	58.2	57.9	69.0	60.1	54.9	60.5
	PFEnet[9]	60.5	69.4	54.4	55.9	60.1	62.8	70.4	54.9	57.6	61.4
	本文	63.3	70.3	54.5	55.1	60.8	65.9	72.0	57.7	59.6	63.8

3.4 消融实验

为了进一步评估模型中各个模块的有效性, 本文在 PASCAL-5^i 数据集上以 ResNet-50 作为骨干网络进行了消融实验, 实验结果如表 2 所示. 不添加任何模块的基线模型在 1-shot 和 5-shot 设置下的性能分别是 55.2% 和 57.1%, 在使用原型生成模块生成多原型之后, 性能提升了 0.6% 和 1.3%, 这表明了多原型能针对图像类别生成多个语义区域的原型表示, 从而激活查询图像上更大的区域. 进一步引入 Transformer 对原型进行迭代优化后, 性能分别提升了 1.9% 和 2.2%, 这说明 Transformer 可以挖掘支持特征和查询特征, 利用上下文信息进一步优化原型表示. 加入 FMM 模块后模型的性能显著提升, 分别是 1.4% 和 1.6%, 这说明了 FMM 通过注意力机制可以引导模型关注高相似特征. MFM 模块可以生成不同尺度的特征表示, 提高模型对于物体尺度变化的适应能力, 加入后模型的性能分别提升了 2.1% 和 1.8%.

表 2 消融实验结果 (%)

原型生成	Transformer	FMM	MFM	1-shot	5-shot
—	—	—	—	55.2	57.1
√	—	—	—	55.8	58.4
√	√	—	—	57.7	60.6
√	√	√	—	59.1	62.2
√	√	√	√	61.2	64

为了寻找合适的超参数设置, 本文在 PASCAL-5^i 数据集上以 ResNet-50 作为骨干网络进行了实验. 对于

支持图像的前景超像素个数n, 实验结果如表 3 所示. 在n为 2 的时候, 由于此时前景只会生成 2 个原型, 依旧存在着严重的语义模糊问题. 随着n的取值逐渐增大, 模型性能也随之上升, 但是当取值超出 10 以后, 性能不再上升, 这是由于当产生的原型过多时, 原本属于同一种语义类别下的区域也被进一步切分, 较小的区域下生成的原型特征更容易偏离真实的语义原型成为离群点, 从而导致错误的分割结果, 因此本文将超参数n设置为 10.

表 3 不同 n 取值的实验结果

n	2	4	6	8	10	12	14	16
1-shot	58.3	59.6	60.5	60.8	61.3	60.6	61	60.5
5-shot	62.3	61.9.	62.6	63.2	63.7	62.8	62.5	62.3

本文对查询图像的超像素个数m也进行了实验, 由于查询图像额外包含背景, 而背景往往比较杂乱, 因此相比n而言, m需要设置成更大的值, 从而对前景和背景都能生成准确的超像素分割结果. 表 4 中不同m取值下的实验结果与表 3 中出现了相似的趋势, 即取值太小或者太大都会引起性能下降, 因此本文将m设置为 25.

表 4 不同 m 取值的实验结果

m	16	19	22	25	28	31	34	37
1-shot	60.4	60.2	61.3	61.7	61.4	60.9	61.1	59.6
5-shot	62.4	62.1.	62.3	63.6	63.1	62.9	62.8	60.6

本文对于 Transformer 层的数量l进行了实验, 在表 5 中可以看到, 随着层数的增加, 模型的性能逐渐上升, 但是当l大于 4 之后, 性能提升进入了瓶颈. 对于小样本学习而言, 过多的参数会导致模型在训练集上过拟合, 并且也会影响运算性能, 而过少的层数也无法发挥出 Transformer 强大的上下文建模能力, 因此本文将层数l设置为 4.

表 5 不同 l 取值的实验结果

l	0	1	2	3	4	5	6	7
1-shot	56.5	57.3	57.8	59.2	61.1	60.6	61.0	60.8
5-shot	59.4	60.4	60.9	61.7	64.1	63.9	63.2	63.4

4 结论与展望

本文提出了一种基于语义匹配的小样本语义分割模型, 利用超像素在保留空间结构的前提下提取多个原型, 利用图神经网络对原型执行校正, 从而提高模型对类别的表达能力. 模型通过 Transformer 进一步利用来自查询图像和支持图像上的语义信息和上下文对原型进行迭代优化, 从而得到完整准确的原型特征, 随后利用原型分配模块将得到的原型嵌入到特征中, 以细粒度的方式传播相似性. 实验结果表明了本文提出的模型能够生成精确完整的类的原型表示, 并且能够利用原型对查询图像执行分割, 通过解码器得出更好的分割结果. 下一步将继续对原型生成和优化过程进行研究, 在不损害泛化能力的前提下提高模型对类别的描述能力和分割能力, 从而达到更好的分割效果.

参考文献

1 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2015. 3431–3440.

2 丁才富, 杨晨, 纪秋浪, 等. MCA-Net: 多尺度综合注意力 CNN 在医学图像分割中的应用. 微电子学与计算机, 2022, 39(3): 71–77.

3 张勋晖, 周勇, 赵佳琦, 等. 基于熵增强的无监督域适应遥感图像语义分割. 计算机应用研究, 2021, 38(9): 2852–2856. [doi: 10.19734/j.issn.1001-3695.2020.11.0431]

4 Shaban A, Bansal S, Liu Z, *et al*. One-shot learning for semantic segmentation. British Machine Vision Conference. London: BMVA, 2017. 4–7.

5 Dong NQ, Xing EP. Few-shot semantic segmentation with prototype learning. British Machine Vision Conference. Newcastle: BMVA, 2018. 3–6.

6 Wang KX, Liew JH, Zou YT, *et al*. PANet: Few-shot image semantic segmentation with prototype alignment. 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 9196–9205.

7 Achanta R, Shaji A, Smith K, *et al*. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(11): 2274–2282. [doi: 10.1109/TPAMI.2012.120]

8 Zhang C, Lin GS, Liu FY, *et al*. CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 5217–5226.

9 Tian ZT, Zhao HS, Shu M, *et al*. Prior guided feature enrichment network for few-shot segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(2): 1050–1065. [doi: 10.1109/TPAMI.2020.3013717]

10 Everingham M, van Gool L, Williams CKI, *et al*. The

PASCAL visual object classes (VOC) challenge. International Journal of Computer Vision, 2010, 88(2): 303–338. [doi: 10.1007/s11263-009-0275-4]

11 Lin TY, Maire M, Belongie S, *et al*. Microsoft COCO: Common objects in context. 2014 13th European Conference on Computer Vision. Zurich: Springer, 2014. 740–755.

12 Siam M, Oreshkin B, Jagersand M. AMP: Adaptive masked proxies for few-shot segmentation. 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 5248–5257.

13 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations. San Diego: IEEE, 2015. 7–9.

14 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778.

15 Deng J, Dong W, Socher R, *et al*. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 248–255.

16 Zhang C, Lin GS, Liu FY, *et al*. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 9586–9596.

17 Yang BY, Liu C, Li BH, *et al*. Prototype mixture models for few-shot semantic segmentation. 2020 16th European Conference on Computer Vision. Glasgow: Springer, 2020. 763–778.

18 Liu YF, Zhang XY, Zhang SY, *et al*. Part-aware prototype network for few-shot semantic segmentation. 2020 16th European Conference on Computer Vision. Glasgow: Springer, 2020. 142–158.

19 Liu BH, Ding Y, Jiao JB, *et al*. Anti-aliasing semantic reconstruction for few-shot semantic segmentation. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 9747–9756.

20 Wang HC, Zhang XD, Hu YT, *et al*. Few-shot semantic segmentation with democratic attention networks. 2020 European Conference on Computer Vision. Glasgow: Springer, 2020. 730–746.

(校对责编: 牛欣悦)