K-Query:基于关键点查询的全景分割方法

姚治成 王 卅 包云岗
(中国科学院计算技术研究所 北京 100190)
(中国科学院大学 北京 100190)

摘 要 全景分割是图像语义分割与实例分割的融合,在自动驾驶、机器人应用等领域有重要应用价值.在当前基于深度学习的全景分割方法中,基于"查询"的方法在分割流程上统一了语义分割任务和实例分割任务,取得了当前最优全景分割效果.该类方法将自然语言处理中的注意力机制应用到了图像分割领域,然而由于输入图片数据量远大于文本句子数据量,该方法无法直接采用输入数据作为查询向量,为此构建了固定数量的静态向量作为"查询".但是,该静态查询设计存在查询向量个数不好确定,容易出现实例表示混淆等问题.在基于静态查询的设计中,需要人为地根据经验去设定实例查询向量的个数,但是在实际情况中,输入图片中实例的个数不是固定的,在动态变化.如果把需要的查询向量个数设置的太少,少于图片中的实例数,则多的实例就无法表示.且由于每一个查询向量在解析过程中都会生成一张对应的掩码图片,多一倍的查询向量就会多一倍的资源开销,因此如果设置了太多的查询向量,在一些图片输入下就可能导致大量的资源浪费.另一方面,由于静态设置的查询向量和需要解析的输入图片不相关,在某些情况下,一个静态查询向量可能会得到多个事物的掩码,或者多个静态查询向量得到相同物体的掩码,导致查询向量在事物表示上发生混淆.为了解决该问题,我们期望查询向量是动态的,和输入图片中待查询的事物相关,且每个查询向量之间都具有一定的可区分性,为此本文提出了一种基于目标物体关键点的动态查询全景分割方法,称之为 K-Query.为了让实例查询向量与图片中的实例直接关联,并在它们之间具有一定的区分距离,本方法首先将图片中的实例通过深度神经网络映射为可区分的高维嵌入编码,并保证同一个物体对应像素点的编码距离足够近,不同物体间像素点的编码距离足够远,然后基于快速"行列式"聚类方法为每一个物体都挑选一个对应的高维嵌入编码和对应的位置编码作为最终的实例查询向量.K-Query 方法中的查询向量,动态地来自于输入图片中目标物体自身的高维嵌入编码,能避免静态查询面临的问题,进一步提升了全景分割性能.本文基于 detectron2 框架对 K-Query 进行了实现,并在多个数据集上进行了验证.测试结果表示,在 Res50 的骨干网络配置下,K-Query 在 Cityscapes val 数据集上的全景分割结果为 63.2% *PQ*,在 COCO panoptic 2017 val 数据集上的 *PQ* 值为 52.9%,相比当前最优全景分割方法,它在 *PQ* 值上分别提升了 1.1 和 1.0 个点(points).

关键词 深度学习;图像分割;聚类;实例分割;全景分割
中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2023.01693

K-Query: Panoptic Segmentation with Keypoint-Based Query

YAO Zhi-Cheng WANG Sa BAO Yun-Gang
(*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)
(*University of Chinese Academy of Sciences, Beijing* 100190)

Abstract Panoptic segmentation is the fusion of image semantic segmentation and instance segmentation. It has significant application value in automatic driving, robotic applications, etc. The query-based method unifies semantic and instance segmentation tasks in the current deep learning-based panoptic segmentation methods. This method applies the attention mechanism in natural language processing (NLP) to image segmentation and achieves the recent best panoptic segmentation results. However, because the amount of input image data is far more significant

收稿日期:2022-11-14;在线发布日期:2023-04-06.本课题得到国家自然科学基金项目(62090020,61672499)、中国科学院青年促进创新会项目(2013073)、中国科学院先导项目(XDC05030200)资助.**姚治成**,博士研究生,工程师,中国计算机学会(CCF)会员,主要研究方向为计算机体系结构、计算机视觉、机器人系统.E-mail:yaozhicheng@ict.ac.cn.**王 卅**(通信作者),博士,副研究员,硕士生导师,中国计算机学会(CCF)会员,主要研究领域为计算机体系结构、云计算、操作系统.E-mail:wangsa@ict.ac.cn.**包云岗**,博士,研究员,中国计算机学会(CCF)会员,主要研究领域为计算机体系结构、云计算、操作系统、开源芯片.

than the amount of text sentence data in NLP, this method cannot directly use the input data as the query vector. So current works construct a fixed number of static vectors as queries. However, this static query design has problems such as the uncertain number of queries and the confusion in instance representation. On the one hand, in this static query-based design, it is necessary to manually set the number of queries based on experience. But, in practice, the number of instances in the input image is not fixed and varies dynamically. If the number of required query vectors is too small to be less than the number of objects in the picture, then the queries cannot represent more instances. And because each query vector generates a corresponding mask image during the parsing process, doubling the query vector will double the resource overhead. Therefore, if too many query vectors are set, many resources may be wasted under some image inputs. On the other hand, because the static query vectors are not related to the input image, in some cases, a static query vector may generate a mask for multiple objects, or multiple static query vectors may obtain masks for the same thing, which is a confusion problem in the representation of the queries. To solve this problem, we expect the query vector to be dynamic and related to the things queried in the input image. Each query vector has a certain distinguishability. Therefore, this paper proposes K-Query, an active query panoptic segmentation method based on the objects' key points. It aims to associate the instance's query vector with the instances directly in the picture and make a certain distance between them. K-Query first maps the instances into distinguishable high-dimensional embedded through the depth neural network. It ensures the embedding distance of the corresponding pixel points of the same object is close enough and the distance between different things is far enough. Then, based on a fast clustering method, K-Query selects a corresponding high-dimensional embedded and its position encoding for each object to compose the final instance queries. The query vectors in this method are dynamically generated from the high-dimensional embeddings of the instance itself in the input image, which can avoid problems in static query approaches and improve the performance of panoptic segmentation results. We implement K-Query based on detectron2 and evaluate it on two datasets. The test results show that K-Query achieved 63.2% *PQ* and 52.9% *PQ* on the Cityscapes val and COCO panoptic 2017 val dataset. It has increased *PQ* by 1.1 and 1.0 points compared to the current state-of-the-art method.

Keywords deep learning; image segmentation; clustering; instance segmentation; panoptic segmentation

1 引 言

全景分割[1]的目标是从输入图片中解析出像素点级别的全景信息,具体包括语义信息和实例信息.语义信息即图片中物体的类别,实例信息用于区分相同类别下不同实例的个体.图片中不可数物体,例如天空、草地等背景称之为 Stuff,仅有语义信息没有实例信息;可数物体称之为 Thing,既有语义信息也有实例信息,例如车辆、行人等.因此全景分割结果包含了传统图像语义分割[2]和实例分割[3]的结果,可以用来从图片和视频中提取关键信息,在自动驾驶、工业机器人等领域有重要应用价值[4-5].当前基于深度学习的全景分割方法[6-12],按照实例掩码的生成方式,可分为自顶而下、自底而上和基于查询的三大类方法,它们在不同数据集上取得了很好的结果.其中基于查询的方法克服了前两类方法的部分缺点,是最新的解决方案,分割结果表现最好[9-10],且有运行速度快、后处理简单等诸多优点.但当前基于查询的方法利用"静态"查询进行实例掩码生成和类别提取,导致其在实例生成上依旧存在不足.如图 1 所示,在一些场景下无识别某些物体导致实例丢失,或者把多个物体识别成同一个物体导致实例混淆.

上述三大类别的全景分割方法按照技术发展顺序存在一定的迭代演进关系.自顶而下的实例生成

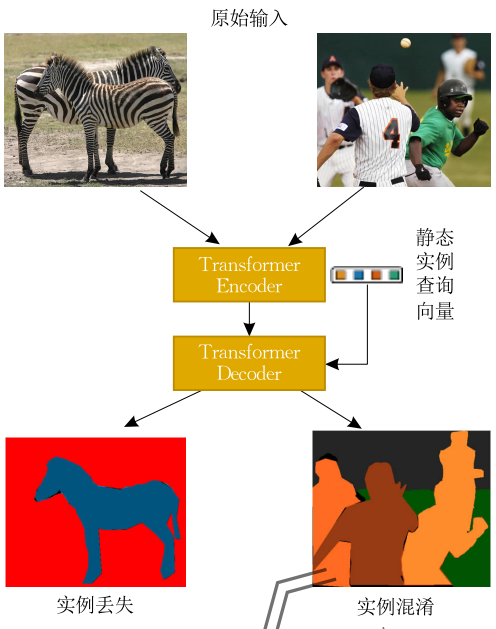


图 1　“静态”查询分割结果面临的问题

方法通常是基于边界框的两阶段方法，它首先利用目标检测[13]技术得到目标实例的边界框和类别，然后基于边界框，利用深度神经网络生成对应实例的掩码. 该方法的代表性的工作有 Panoptic FPN[12]、AUNet[7]、UPSNET[6]、Unifying[14]、EfficientPS[15]等. 其中 Panoptic FPN[12]最为典型，它在实例分割网络 Mask RCNN[16]基础之上增加了特征金字塔网络，让语义分割分支和实例分割分支共享特征网络，在后处理过程中基于置信度消除两个分支输出的重叠部分. 其他方法在结构上与 Panoptic FPN 类似，主要在骨干网络[8,15]、融合方式[17]、解码器[6,14]等方面进行创新改进. 自顶而下的技术分割效果好，但在生成结果过程中存在多个阶段，有复杂的预处理和后处理过程，导致其运行速度慢.

为了提升自顶而下全景分割技术的运行速度，Yang 等人提出了“一阶段”的、“自底而上”的图像分割技术[18-20]，他们利用深度神经网络直接一次性输出图片中实例的关键点信息、实例像素点偏移等，然后在后处理过程中“聚合”出实例掩码、边界框、类别等信息. 该类方法在处理速度上有所提升，例如 Panoptic-DeepLab[11]利用质量心和实例像素点偏移聚合出实例掩码，然后利用 Majority Vote[19]进行语义信息和实例信息的融合，在 2048×1024 的高清图片上可以做到近实时的处理速度. 自底而上的方法模型简单，简化了实例生成的流程，虽然处理速度快，但通常在全景分割性能上有一定的下降.

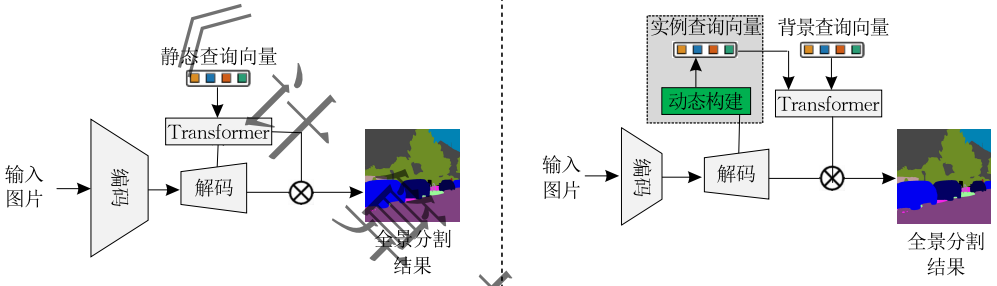
基于查询的方法是最新提出的全景分割方法，它把 NLP[21]领域的注意力机制和 Transformer 结构应用到了图像分割领域. 该方法无论是在速度上，还是分割效果上，都全面超越了前两种方法，例如在相同骨干网络下，基于查询的 Mask2Former[10]方法相对经典的、自顶而下的 Panoptic FPN 在 Cityscapes val[22]数据集上有 4.0 个点的全景分割质量(Panoptic Quality，PQ)提升. 基于查询的方法是注意力机制在图像处理领域的应用. 注意力机制往往采用“查询-键-值”(Query-Key-Value)[21]的模式，其中 Query 通常表示目标输入，Key 和 Value 是数据特征的“隐层”表示. 基于注意力机制可以构建 Transformer 网络结构，增强网络的泛化性能. 基于查询的全景分割方法中代表性的有 DETR[23]、K-Net[9]和 Mask2Former[10]等.

基于查询的全景分割方法通常需要手动构建初始查询变量作为输入进行语义分割和实例分割，最终分割结果中的每一个“实例掩码”和“语义掩码”信息都需要对应一个单独的查询向量，因此如何设置初始查询变量的数量成为一个关键挑战. 通常情况下，输入图片中的实例数是预先不可知的，查询变量数量设置太少会导致实例丢失，太多会导致开销过大. 如图 1 所示，因查询向量个数不足会导致分割结果中实例丢失，或者一个查询得到多个实例掩码造成实例混淆. 然而，当前基于查询的全景分割方法都是根据经验设置固定数量(例如 100 个)的静态查询向量进行全景分割，在部分场景中会出现实例混淆的问题.

为解决这一挑战，我们思考，是否可以通过分析输入图像本身来“动态”生成所需的查询向量？先通过简单的方法得到图像中实例的个数、大致位置等基本信息，基于这些信息构建查询向量，然后再通过 Transformer 结构得到更精细、更准确的全景分割结果. 由于注意力机制“内在”是通过计算查询向量与图片中实例高维嵌入的相似度来进行对应掩码的生成[23]. 为了达到高质量分的割结果，需要动态生成的查询向量与实例嵌入有一定的相似性，而各个查询向量之间具有足够的区分度.

为解决这一挑战，我们思考，是否可以通过分析输入图像本身来“动态”生成所需的查询向量？先通过简单的方法得到图像中实例的个数、大致位置等基本信息，基于这些信息构建查询向量，然后再通过 Transformer 结构得到更精细、更准确的全景分割结果. 由于注意力机制“内在”是通过计算查询向量与图片中实例高维嵌入的相似度来进行对应掩码的生成[23]. 为了达到高质量分的割结果，需要动态生成的查询向量与实例嵌入有一定的相似性，而各个查询向量之间具有足够的区分度.

基于上述目标，本文提出一种如图 2(b)所示，名为 K-Query 的基于关键点构建“动态”查询的全景分割方法. 为了动态地构建更准确的查询向量，K-Query 需要解决两方面的挑战：首先需要对图片中所有实例的像素点进行可区分编码，让不同实例的“编码”之间有足够的区分距离. 然后需要快速地从“编码数据”中构建出一组与所有实例都有一一对应关系的查询向量. 为此，K-Query 针对性地进行了两方面的工作解决对应挑战：(1) 利用实例嵌入技术把图片中的物体对应的像素点进行高维嵌入(Embedding)编码[24]，让来自同一个物体的像素点对应的嵌入编码之间的距离足够近，让来自不同实例的像素点对应的嵌入编码之间的距离足够远. 该过程的目标是保证每个图片中待分割的对象在粗粒度上都有一定的区分度，其中每一个像素点对应的高维嵌入编码都是候选查询向量；(2) 利用快速聚类的方式从上一步生成的候选查询向量中，根据向量之间的距离为每一个对象选择一个高维嵌入向量和对应的位置编码作为最终的查询向量. 基于上述过程，K-Query 可以从图片中根据实例“动态”地构建查询，克服了当前静态查询全景分割方法面临的查询个数不好确定的问题. 经过测试，K-Query 在相同骨干网络、相同 Transformer 结构配置下，相对当前最优静态查询方法在 Cityscapes val 数据集和 MS COCO panoptic 2017 val[25] 数据集上分别取得了全景分割质量 1.1 和 1.0 个点的提升.



(a) 静态查询方法 (b) 动态查询方法

图 2 现有静态查询方法和目标动态查询方法

综上所述，本文的主要贡献如下：

(1) 对当前最先进的，基于查询的图像全景分割方法进行了分析，发现“静态查询”方式是限制其分割效果的一大主要因素；

(2) 针对静态查询问题，提出了一种基于关键点的动态查询全景分割方法，它能有效地缓解静态查询带来的如查询个数不好确定、部分场景性能不够等问题；

(3) 提出了一种简单、快速且张量计算框架友好的“行列式”聚类方法. 该方法利用向量的并行计算特征，通过简单地计算“行(列)”最大值的方法快速完成聚类；

(4) 对所提出的方法进行了实现和验证，其测试结果表明本文提出的 K-Query 全景分割方法相对静态方法有更好的分割效果，在典型数据集上全景分割效果最大提升 1.1 个点.

2 相关工作

2.1 语义分割

语义分割[2] 是计算机图像处理领域中的基础任务，它给输入图片中的每个像素点都标记出“语义信息”，即类别标签，例如汽车、房屋、行人等，最终得到的结果为带语义标记同输入图片同等大小的掩码图片. 不同的数据集在语义分类上有很大不同，例如针对自动驾驶的 Cityscapes 数据集[22]，共有 19 个语义分类，而 MS COCO 数据集[25] 有 133 个语义分类. 最常用的语义分割结果的质量评价标准为平均类别 IoU(Intersection of Union)，简称 $mIoU$. 其计算公式如式(1)所示，其中 M_{Pred} 表示预测的语义掩码，M_{true} 表示真实的语义掩码.

$$IoU=\frac{M_{\text{pred}}\bigcap M_{\text{true}}}{M_{\text{pred}}\bigcup M_{\text{true}}} \tag{1}$$

当前基于深度学习的语义分割方法通常采用 Encoder-Decoder[26] 的深度卷积结构，模型输出利用 Softmax[27] 函数进行处理，使其值为每个像素点对应类别的概率，然后使用 argmax 函数快速得到每个像素点所预测的分类，进而得到整个输入图片的语义分割掩码. 该分割过程简单直观，被大多数研究工作采用. 在语义分割工作中，研究点侧重在骨干网络、解码结构、损失函数、处理速度等方面. 例如 He

等人提出残差网络 ResNet[28]，利用残差结构缓解梯度消失问题；Sun 等人[29]提出 HRNet 高分辨率网络，利用融合高低分辨率特征来增强模型表达能力；Liu、Dosovitskiy 等人将 Transformer 应用到骨干网络，构建 Swin[30]、ViT[31]等表达能力更强的网络. 在解码结构方面，Lin 等人提出 FPN[32]解码结构，将高分辨率和低分辨率特征进行融合；Chen 等人提出一种 ASPP[33]池化结构进行特征融合. 在损失函数方面，OHEM[34]、RMI[35]是语义分割等领域常用的损失函数. 在提升模型速度方面，代表性的工作有 MobileNets[36]、DDRNet[37]、ShuffleNet[38]等.

从语义分割的定义可知，该工作只能对图片中对象的像素点类别进行标记，无法对多个实例进行区分，例如无法区分道路上的多个车辆或者行人. 仅仅知道语义信息无法完成自动驾驶和工业机器人等领域的相关任务，因此还需要知道图片中的实例信息.

2.2　实例分割

实例分割[16]同样也是计算机图像处理领域的基础任务，目标是解析出输入图中的“实例”掩码. 实例即图片中可数目标的具体实体，例如标记多个车辆中的每个车辆对应的像素点掩码. 相对语义分割，实例分割仅仅关注有实例的物体类别，而不包含对背景像素点的处理. 在 Cityscapes[22]数据集中，有实例的语义类别仅有 8 个，在 MS COCO[25]数据集中有 80 个语义类别拥有实例. 实例分割通常采用平均识别精度(Average Precision)作为评价指标，其定义如式(2)所示，其中 R 和 P 表示在 PR 曲线中对应的召回率和精度.

$$AP=\sum_{n}(R_n-R_{n-1})P_n \tag{2}$$

相比语义分割，传统实例分割的分割流程要复杂很多，按照分割流程不同可以分为“自顶而下”和“自底而上”等类别. “自顶而下”类方法也称为“基于边界框”的方法，它首先利用目标检测方法[13]得到目标物体的类别和具体位置，然后基于位置生成对应的掩码. 该过程通常包括多个预处理，和后处理阶段，例如“提议”的选取、基于非极大值抑制(Non Maximum Suppression，NMS)[16]的边界框后处理等. 该类方法代表性工作有 Mask RCNN[16]、YOLACT[39]、Center Mask[40]等工作. Mask RCNN 是在 Faster RCNN[41]基础之上增加掩码分支完成实例分割任务；YOLCAT 方法提出了一种实时网络和快速 NMS 方法；Center Mask 是基于 FCOS 和 SAM 结构的实例分割方法，并提出了一种改进型骨干网络 VoVNetV2. 该类“自顶而下”的方法分割精度高，但处理流程长，时间代价大. 在该模式之外，De Brabandere 等人提出了“自底而上”的方法 Discriminative Loss[42]，该方法的实例掩码处理过程与前一类方法相反，先得到目标物体的“像素点级”编码信息，然后利用聚类等方式聚合出实例掩码，再利用启发式规则得到对应的实例类别. 该类方法如文献[3，20，24，42]通常采用“单阶段”模型，有效地减少了处理时间，但识别精度有所下降.

实例分割仅仅关注可数实例的类别、位置、掩码等信息，不关注背景等不可数事物. 因此在自动驾驶等场景中并不适用，例如重要的“道路”信息无法利用常见的实例分割方法得到.

2.3　全景分割

全景分割[1]不仅需要按像素点为图片中所有的物体提供语义分类结果，也需要生成实例掩码区分出不同的实例. 在全景分割任务中，背景类不可数对象称为 Stuff，前景类有实例的对象叫做 Thing. 全景分割是语义分割和实例分割的有机融合，在自动驾驶，机器人视觉等领域有重要应用价值. 在语义信息上，全景分割和语义分割完全相同，但是在实例表示上和实例分割存在一定差异. 因为在全景分割中，一个像素点只能属于一个实例，不存在重叠，而在实例分割任务中一个像素点可以属于多个实例的掩码，具体如图 3 所示.

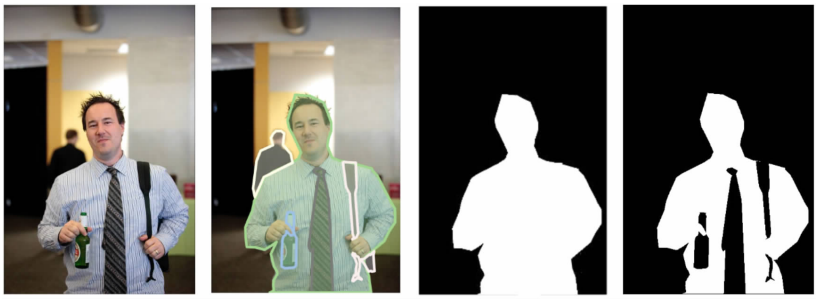


图 3　传统实例分割与全景分割中的实例分割[11]

全景分割任务的评价指标为 PQ(Panoptic Quality)[1],其定义如式(3)～(5)所示,它由 SQ(Segmentation Quality)和 RQ(recognition Quality)组成. SQ 表示分割的质量,是所有物体掩码的 IoU 大于 0.5[4]且识别正确的 IoU 平均值. RQ 代表目标语义类别识别的 $F1$ 分数. 在公式中,TP 表示正确的预测识别,FP 表示错误的预测识别,FN 表示类别正确但实例错误的预测. PQ 取值范围为 0～1,通常使用百分数表示.

$$SQ=\frac{\sum_{p,g\in TP} IoU(p,g)}{|TP|} \tag{3}$$

$$RQ=\frac{|TP|}{|TP|+\frac{1}{2}|FP|+\frac{1}{2}|FN|} \tag{4}$$

$$PQ=SQ\times RQ \tag{5}$$

由于全景分割的指标只有 PQ,因此多数情况下会对 PQ 进行细化,如对无实例类(背景)的全景分割质量用 PQ^{st} 表示,对有实例类(前景)的全景分割质量用 PQ^{th} 表示.

传统的全景分割方法按照其实例生成方式的不同,同实例分割一样,也有"自顶而下"和"自底而上"的分类. 例如 Panoptic FPN[12]、TASCNet[17]、UPSNet[6]、AUNet[7]等网络是典型的自顶而下方法. 它们的侧重点各不相同,Panoptic FPN 在 Mask RCNN 上增加语义分支,然后解决重叠问题进行实例和语义的融合;TASCNet 利用二值掩码加强语义掩码和实例掩码的一致性;UPSNet 利用无分类类别解决语义和实例融合时的冲突;AUNet 利用注意力模块主导全景融合. 在"自底而上"的分类中,DeeperLab[19]利用边界框的四个角以及中心点进行实例掩码的生成;SSAP[18]利用像素点间的亲和性进行掩码聚类;Panoptic-DeepLab[11]预测实例的质心和像素点相对质心的偏移进行实例聚合,然后通过选举的方法得到实例的类别信息.

在传统全景分割方法中,由于实例分割和语义分割在方法上不兼容,需要分别采用独立的 Decoder 分支进行预测,该过程导致相关模型结构复杂,并在后处理时需要对实例掩码和语义掩码进行融合,带来了更多的时间开销.

2.4 基于查询的通用图像分割

为了解决语义分割和实例分割不兼容的问题,Carion 等人提出 DETR[23],将基于查询的方法应用到图像分割领域,该方法使得语义分割、实例分割、全景分割在模型结构、处理流程、数据表示上变得一致,并且在处理速度和分割结果上也有了大幅提升. K-Net[9]、Mask2Former[10]等工作在 DETR 基础上进行了改进,进一步提高了性能.

基于查询的图像分割方法的核心逻辑来源于自然语言处理(Natural Language Processing,NLP)[21]任务中的注意力机制(Attention). 注意力机制一般采用"查询-键-值"(Query-Key-Value)的模式进行表示,具体计算公式如式(6)所示. 其中 $\boldsymbol{Q}$(Query)代表输入的查询向量,例如在翻译任务中可以认为是需要翻译的原始语句,K(Key)和 V(Value)表示数据的高维表示,式(6)可理解为计算输入数据 $\boldsymbol{Q}$ 和全体特征数据 K 之间的相似度(注意力),然后根据相似度获取 V 中的目标值. 基于注意力集机制,叠加"注意力模块"构建的"编码-解码"神经网络称之为 Transformer 网络[21]. 在 NLP 任务中,通常采用"自注意力机制",其 QKV 值都来源于输入数据.

$$Attention(\boldsymbol{Q},K,V)=\mathrm{Softmax}\left(\frac{\boldsymbol{Q}K^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{6}$$

不同于 NLP 任务,在计算机视觉任务中,一次输入图片的数据量远超过 NLP 任务中句子数据量的大小,若采用输入图片的所有数据作为查询向量会导致当前 GPU 设备内存不够用. 因此在图像分割任务中,如图 2(a)所示,首先构建 N(例如 100)个静态 Query,然后通过查询操作生成 N 个掩码和对应的分类信息,最后基于置信度选出大于阈值,且不存在交集的 $M(M\leq N)$ 个掩码作为最后输出. 该过程统一了语义分割和实例分割,同时也简化了后处理过程,相比传统图像分割方法无论是在速度上,还是分割精度上都有很大优势.

虽然最新提出的基于查询的图像分割方法在全景分割任务上相对传统方法能取得更优异的成绩,但是其基于静态查询的方式仍然存在不足,主要表现在两个方面:(1)静态查询的值和输入图片无关联,可能导致计算注意力时相似度不够,进而影响分割性能;(2)对于一个数据集,静态查询的个数不好确定,如果数量过少,小于图片中的分割掩码数则会导致实例或者语义掩码信息丢失,性能下降,如果数量过大又会导致模型消耗的内存、计算等资源开销过大.

3 K-Query 全景分割方法

为了解决上述"静态查询"全景分割方法面临

的问题，本文设计了一种“动态查询”全景分割方法，名为 K-Query，它的查询向量源于图片中需要解析的对象，尽力使得其数量和输入图片中待解析的“掩码”数一致，且查询向量之间的存在足够的差异.

3.1　设计概述

如图 4 所示，相比其他静态查询方法，本文提出的 K-Query 动态查询方法增加了两个部分：“实例嵌入”和“关键点提取”. 按箭头所示，K-Query 的全景分割流程为：(1) 首先利用传统基于 CNN 网络的“编码-解码”模型提取图片的高维特征；(2) 把高维特征并行输入到“实例嵌入”和 Transformer-Encoder[23] 得到实例嵌入向量和 Transformer 高维特征，高维特征对应于注意力机制中的 K 和 V；(3) 通过关键点提取算法从实例嵌入编码中选择查询嵌入和对应的位置编码构建实例查询向量 $\boldsymbol{Q}_{\text{ins}}$，结合背景静态查询向量 $\boldsymbol{Q}_{\text{stuff}}$ 构建查询 $\boldsymbol{Q}$；(4) 然后利用 $\boldsymbol{Q}$、K、V 作为输入通过 Transformer-Decoder 得到最终目标对象的语义分类和对应的掩码. 最后通过后处理获取最优全景分割结果.

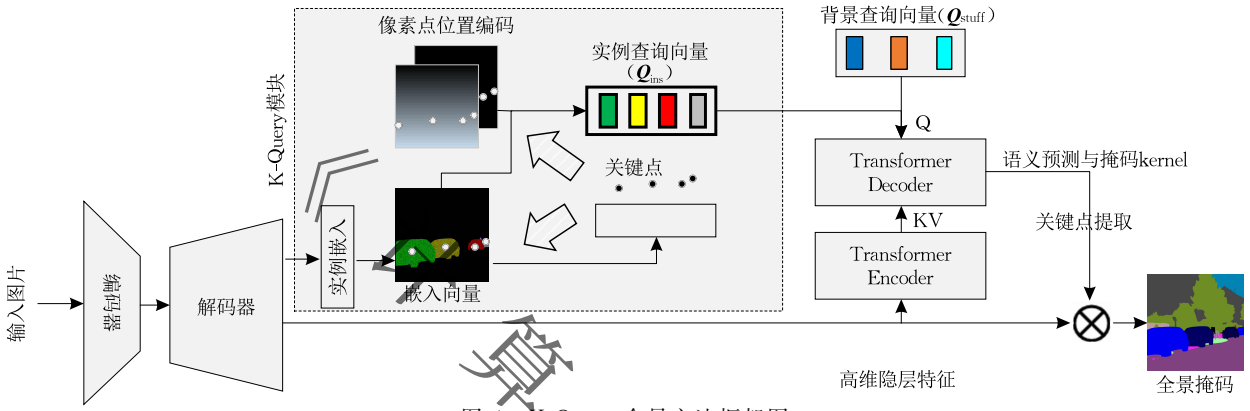


图 4　K-Query 全景方法框架图

为了让图片中实例表示具有可区分性，在“实例嵌入”部分中，本文对文献[42]中的实例嵌入(Instance Embedding)方法进行了改进. 它让来自同一个实例像素点对应的嵌入向量之间的距离足够近，来自不同对象的嵌入向量之间的距离足够远. 经过编码后的实例嵌入向量作为构建查询向量的“候选者”.

由于实例编码后的嵌入向量个数与其像素点个数相等，导致候选者数量巨大. 为了从众多“候选者”中快速为每个实例都选择一个“关键点”去构建查询向量，我们设计了一种“关键点提取”方法，该提取过程的核心在于其中的“行列式”聚类方法，它可以利用 GPU 设备的并行能力大幅提高提取速度.

3.2　实例嵌入

该模块由三层顺序二维卷积构成，输入来自于 Decoder 层的图像特征，输出为 $H_{1/4}\times W_{1/4}\times D$ 的向量，在本文中记作 E，其中 H，W 为输入图片的高度和宽度，D 为像素点嵌入维度. 该模块的目标是为每个图片中的所有对象都进行“可区分”的高维嵌入编码，保证不同实例像素点对应的嵌入向量之间的距离足够远，相同实例的像素点对应的嵌入向量之间的距离足够近. 为了达到该目的，参考文献[42]的方法，在模型训练时该部分采用如下所示的损失函数 L_{emb}，该损失函数由 L_v 和 L_d 两部分组成.

$$L_v=\frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}[\|\mu_c,x_i\|-\delta_v]_+^2 \tag{7}$$

$$L_d=\frac{1}{C(C-1)}\sum_{C_A=1}^{C}\sum_{C_B=1}^{C}[2\delta_d-\|\mu_{cA},\mu_{cB}\|]_+^2 \tag{8}$$

$$L_{emb}=\alpha L_v+\beta L_d \tag{9}$$

在公式中，$\|\boldsymbol{X},\boldsymbol{Y}\|$ 表示向量 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 之间的距离，在本文中该距离被定义为“余弦相似距离”. $[\boldsymbol{X}]_+$ 表示把 $\boldsymbol{X}$ 值进行最小值为 0 的裁剪，即 $\max(x,0)$. C 表示图片中的实例数，N_c 表示实例中像素点的个数；μ_c 表示实例所有像素点嵌入编码的平均值. 若要 L_v 趋于 0，则实例中所有像素点对应的嵌入向量到它们平均值之间的距离需要小于 δ_v. μ_{cA} 和 μ_{cB} 表示图片中实例 A 和 B 的嵌入向量平均值，若要 L_d 趋于 0，则需要图片中任意两个实例的嵌入向量均值之间的距离需要大于 2 倍 δ_d. 最终损失函数 L_{emb} 中的 α 和 β 表示权重，在本文中根据实验经验分别取值 0.9 和 0.1. 参数 δ_v 和 δ_d 分别取值 0 和 0.5

3.3　关键点提取

根据全景分割的目标，以及注意力机制的定义，本文期望查询向量动态地来自图片中实例对应的高维嵌入编码，且每个查询向量都对应一个图片中的实例. 从上一节对图像实例嵌入的定义可知，在输入图片经过嵌入编码后，图片中各个实例对应的像素

点在高维嵌入空间上是可以区分的“簇”.根据一定的“距离”阈值,我们可以对各“簇”进行区分,然后为每个实例都从其“簇”中选择一个关键点作为其对应的查询向量.

基于距离的“关键点提取”流程如算法1所示,具体过程概述如下:

(1)变量初始化.用 False 初始化标记矩阵 $\boldsymbol{M}$,该矩阵的长宽与特征嵌入向量 $\boldsymbol{E}$ 相同;初始化查询向量集合 $\boldsymbol{Q}$ 和对应位置集合 P,初始值为空,设定它们的最大容量为 L.

(2)过滤实例边界.利用核卷积操作,计算 E 中 3×3 矩阵中心点与其余 8 个边界点之间的最大距离,若其最大距离大于阈值 T_DIS,则表示该中心点对应的位置为分割对象的边界,在对应矩阵 $\boldsymbol{M}$ 中标记为 True.

(3)通过聚类选择关键点.从 $\boldsymbol{M}$ 中为 False 的点中随机选择 K 个点作为候选点,并标记 $\boldsymbol{M}$ 中对应位置为 True,然后利用聚类算法把候选点按距离阈值 T_DIS(取值需要小于 δ_d)分成多个簇,从每个簇中随机选择一个点作为该“簇”对应实例的“关键点”.

(4)过滤重复实例的关键点.计算每个选择的关键点和 $\boldsymbol{Q}$ 中向量的最小距离,如果最小距离大于阈值 T_DIS 则将该“关键点”添加至 $\boldsymbol{Q}$ 中,其对应位置信息加入 P 中.

(5)重复执行步骤(3)和(4),直到标记矩阵 $\boldsymbol{M}$ 中的值全都为 True 或者 $\boldsymbol{Q}$ 的长度达到 L.算法执行完成,返回集合 $\boldsymbol{Q}$ 和 P.

算法 1. 关键点提取.

输入:$\boldsymbol{E}, L, K, T\_DIS$

输出:$\boldsymbol{Q}, P$

```
1. //初始化 M,Q 和 P
2. H,W=E.shape[:2]
3. M=zeros(H,W).bool()
4. Q=List(size=L), P=List(size=L)
5. //边缘过滤
6. M[edge(E,T_DIS)]=True
7. WHILE M.false_count()>0 OR Q.full(): //遍历
8.    //在候选点中随机选取 K 个点
9.    mask=random_mask(M==False,K)
10.   //标记以处理点
11.   M[mask]=True
12.   //从聚类后的各簇中随机选取一个点
13.   position=random_gc(rc_cluster(E[mask],T_DIS))
14.   //计算所选点与已选关键点集合 Q 之间的最小
        距离
15.   //大于 T_DIS 则为新加入的关键点
16.   position=paire_dist(E[position],Q).min()>
        T_DIS
17.   P.append(position)
18.   Q.append(E[position])
19. RETURN P,Q
```

由于实例嵌入部分由深度神经网络实现,其输出的实例嵌入编码与预期存在一定的误差,特别在实例边界部分,其误差较大.因此在关键点提取算法中,需要利用步骤(2)对边界上的点进行过滤.在利用聚类算法生成“簇”的过程中,理想情况下可以对过滤后的整个实例嵌入向量进行聚类操作,但是该过程需要消耗大量的内存,导致现有的 GPU 等计算设备无法满足需求,因此采用了循环步骤(3)和(4)的方式利用多次聚类的方法减少算法运行过程中的内存消耗.

通常情况下,聚类操作的目的是将数据集合按照某种“规范”划分成若干个不相交的子集,每个子集称为一个簇.因此对于同样的数据,采用不同的聚类方法会得到不同的结果.例如 K-Means[43] 算法需要人为指定聚类的目标“簇”个数,而 Mean Shift[44]、DBSCAN[45] 等算法生成的“簇”的个数由数据和算法参数决定.常见的聚类算法需要“循环”计算数据点之间的距离,然后进行分类、合并等操作.该过程与现有基于张量的深度学习计算框架不友好,也不支持批处理操作.然而在基于查询的全景分割模型中,“查询向量”需要参与训练,即本文提出的关键点提取过程也需要参与模型训练,但由于常见聚类方法运行速度太慢,不适合参与该训练.为此本文提出了一种适用本场景,针对小规模数据、张量友好、简单快速的行列式聚类算法(Row Column Cluster, RC Cluster).该聚类算法见算法 2,具体执行流程如下:

(1)计算输入长度为 L 的数据 D 中每个元素之间的距离,得到距离矩阵,并按阈值 T_DIS 构建运算矩阵 ***DMAP***.元素之间距离值小于阈值 T_DIS 的元素,在 ***DMAP*** 中对应的位置记为 1,其他位置标记为 0,然后对 ***DMAP*** 的对角线按取值 1 到 L 进行赋值;

(2)复制矩阵 ***DMAP*** 当前值为 TMP;

(3)在 ***DMAP*** 所有非零元素中,按“列”赋予该列最大值;然后按行赋予该“行”最大值;

(4)对比 ***DMAP*** 和 TMP,如果它们的值不相同,则记录 ***DMAP*** 当前值到 TMP 然后转至骤(3)重复执行;

(5)此时矩阵 ***DMAP*** 的对角线为聚类结果,返回该结果.

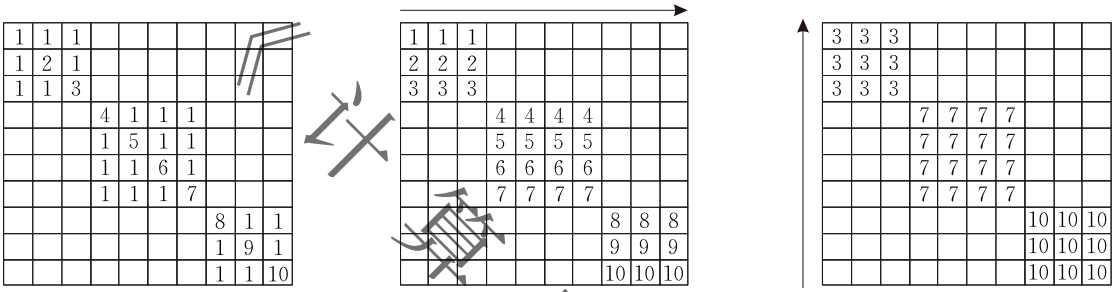
算法 2. “行列式”聚类算法.

输入：D、T_DIS

输出：I

```
//初始化
L=D.length()
//计算数据元素之间的距离矩阵得到邻接矩阵
DMAP=(paire_dist(D,D)<=T_DIS).int()
//对角线赋值
DMAP[diag]=range(1,L)
TMP=DMAP.copy()
//行列循环计算
WHILE TRUE:
  //每行非零元素赋该行最大值
  DMAP[:,…]=DMAP.row_max()
  //每列非零元素赋该列最大值
  DMAP[…,:]=DMAP.colu_max()
  //如果矩阵不在变化，完成聚类
  IF DMAP==TMP:
    BREAK
  TMP=DMAP.copy()
RETURN DMAP[diag] //返回对角线聚类结果 I
```

如图 5 所示，示例距离邻接矩阵在进行一次行列计算后完成收敛，最终得到三个“簇”，其编号分别为 3、7、10. 在该聚类算法中，其核心运算操作只有按行(列)计算最大值，然后进行赋值，该计算过程支持常见深度学习中的批量化操作(batch)，且对现有 TensorFlow[46]、PyTorch[47]、MxNet[48] 等张量计算框架友好，相比传统算法在张量框架中具有速度快、实现简单等优点.



(a) 构建邻接矩阵，初始化对角线　(b) 按“行”赋予最大值　(c) 按“列”赋予最大

图 5　“行列式”聚类方法处理流程示例

由于本文提出的行列式聚类方法，是针对实例嵌入中每个实例之间在嵌入编码上具有特定距离的条件来生成“簇”，因此该方法只有生成的实例嵌入编码符合预期定义的情况下才能达到区分实例的目的.

4　实验验证

为了验证本文提出的 K-Query 是否具有更好的全景分割性能，我们对该方法进行了编码实现，并在两个典型数据集上进行了全景分割效果测试.

4.1　具体实现

4.1.1　数据集

本文采用Cityscapes数据集和MS COCO panoptic 2017 数据集作为测试数据. 其中 Cityscapes 数据主要针对自动驾驶场景，拍摄于欧洲部分城市的街道，图片分辨率为固定 2k 分辨率(2048×1024)，数据集共有 19 个语义分类，其中 8 个类别具有实例. 其训练集、验证集、测试集分别由 2975、500、1525 张图片组成. MS COCO panoptic 2017 数据集中的图片来源于日常场景，如包括室内物品成列、室外运动等. 该数据集有语义分类 133 种，其中具有实例信息的分类有 80 种. 该数据集的图片大小不固定，但最大分辨率不超过 640×640. 其对应的训练集、验证集、测试集分别由 118k、5k、20k 张图片组成. 该两类全景分割测试数据集是领域类的代表性数据集.

4.1.2　实现细节

本文基于 detectron2[49] 框架对 K-Query 进行编码实现. 该编程框架对常见的骨干网络、损失函数、优化器等都有官方实现. 基于其进行模型开发能大幅地减少工作量，把开发重点放在核心逻辑上. 由于在结构上，K-Query 也是基于查询的全景分割方法，可以基于所有静态查询方法进行实现，在实验中我们选用最具代表性的、目前最先进的工作 Mask2Former[10] 作为基础，在其之上增加嵌入结构和关键点查询模块. 在模型配置上，采用 Res50[28] 作为骨干网络. 根据实验经验，嵌入模块的输出维度 D 设置为 256. 关键点算法参数 L 取值 512，K 取值 1000，T_DIS 取值 0.1. 其他参数，例如 batch_size、学习率(learning rate)、训练迭代次数(steps)、优化器(optimizer)等设置为与工作[10]相同. 最后采用式(10)作为最终的损失函数，其中 γ 按实验经验设置为 10.

$$Loss=L_{\text{Mask2Former}}+\gamma L_{emb} \tag{10}$$

4.1.3　硬件环境

本文在测试验证过程中使用的硬件环境为单台

GPU服务器,其CPU配置为2路AMD EPYC 7302处理器,内存为256 GB,GPU为8块NVIDIA GeForce RTX 3090.

4.2 分割结果

在数据集Cityscapes val数据集上的测试结果如表1所示,其中K-Query在相同骨干网络配置下超过其他所有方法取得了63.2% *PQ*的分割效果,相对当前最好的静态查询方法Mask2Former[10]提升1.1个点.

表1 数据集Cityscapes val上的全景分割结果对比

方法名称	骨干网络	*PQ*	PQ^{th}	PQ^{st}
自上而下的方法				
Panoptic FPN[12]	Res101-FPN	58.1	52.0	62.5
RT Panoptic[50]	Res50-FPN	58.8	52.1	63.7
AUNet[7]	Res101-FPN	59.0	54.8	62.1
UPSNet[6]	Res50-FPN	59.3	54.6	62.7
Seamless[51]	Res50-FPN	60.2	55.6	63.6
EfficientPS[15]	Res50	60.3	55.3	63.9
TASCNet[17]	Res50-FPN	60.4	56.1	63.3
Unifying[14]	Res50-FPN	61.4	54.7	66.3
自下而上的方法				
DeeperLab[19]	Xception-71	56.5	—	—
SSAP[18]	Res50-FPN	58.4	50.6	—
AdaptIS[52]	Res50	59.0	55.8	61.3
Panoptic-DeepLab[11]	Res50	60.3	51.1	67.0
静态查询的方法				
PanopticFCN[53]	Res50-FPN	61.4	54.8	66.6
Mask2Former[10]	Res50	62.1	54.8	67.3
动态查询的方法				
K-Query	Res50	**63.2**	**56.2**	**68.3**

K-Query在MS COCO panoptic 2017val数据集上的全景分割结果如表2所示,它的*PQ*结果为52.9%,在相同骨干网络配置下,同样超过了所有其他的全景分割方法.相比当前最优的Masks2Former方法提升1.0个点.

表2 MS COCO panoptic2017 val上的全景分割结果对比

方法名称	骨干网络	*PQ*	PQ^{th}	PQ^{st}
自上而下的方法				
Panoptic FPN[12]	Res101-FPN	39.0	45.9	28.7
AUNet[7]	Res101-FPN	39.6	49.1	25.2
UPSNet[6]	Res50-FPN	42.5	48.6	33.4
Unifying[14]	Res50-FPN	43.4	48.6	35.5
自下而上的方法				
DeeperLab[19]	Xception-71	33.8	—	—
SSAP[18]	Res50-FPN	36.5	—	—
Panoptic-DeepLab[11]	Res50	35.5	37.8	32.0
静态查询的方法				
PanopticFCN[53]	Res50-FPN	44.3	50.0	35.6
K-Net[9]	Res50-FPN	47.1	51.7	40.3
Max-DeepLab[54]	MaX-S	48.4	53.0	41.5
Mask2Former[10]	Res50	51.9	57.7	43.0
动态查询的方法				
K-Query	Res50	**52.9**	**58.9**	**43.8**

由于K-Query在实现上,是基于Mask2former增加动态查询而来,在上述训练测试过程中,它们在骨干网络、transformer配置、训练参数等完全一样,因此我们可以认为K-Query在Cityscapes数据集和MS COCO panoptic 2017数据集上,相对Mask2Former平均1个点*PQ*的提升完全来源于其动态查询特征.

图6展示了K-Query与Mask2Former的部分全景分割图片效果的对比.在实例结果中,对于部分图片Mask2Former存在实例丢失,且对于左下角的

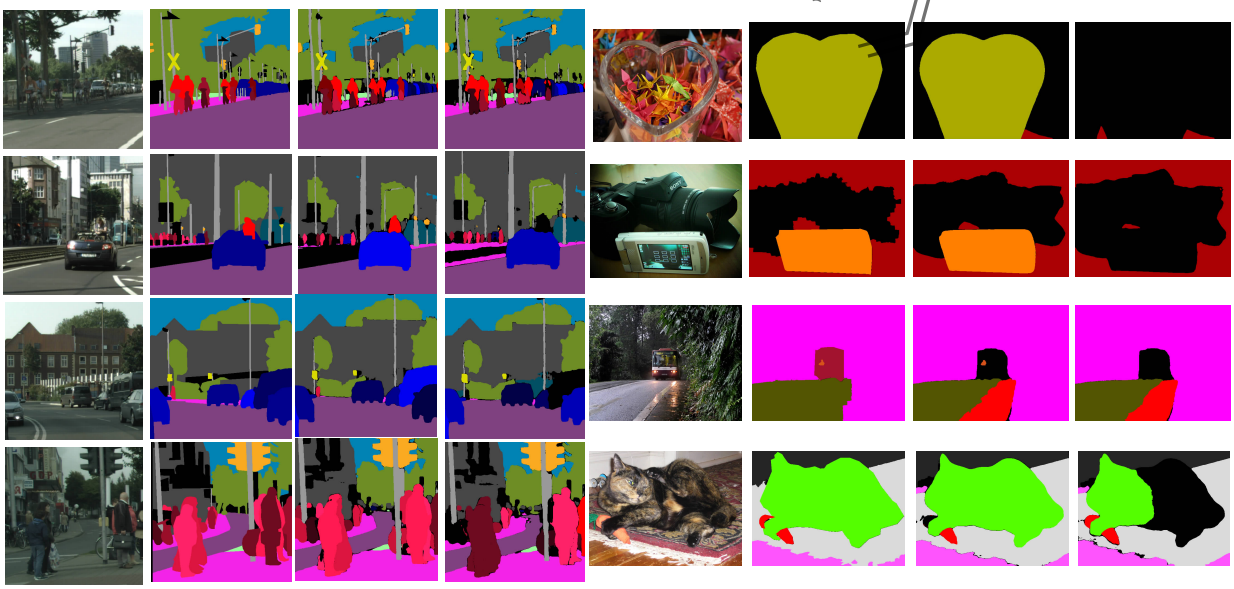


图6 K-Query与Mask2Former在部分图片下的全景分割效果对比

图片，错误地把两个行人实例识别成了同一个导致实例混淆，而 K-Query 则没有类似问题. 本文认为，该提升的根本原因在于 K-Query 的查询向量是和输入图片相关，且动态地来自每个需要分割的目标物体，相对静态查询有更好的分割精度.

4.3　扩展实验

在扩展实验中使用的数据集为 Cityscapes val，实验中未明确说明的参数和硬件环境的配置与上一章节相同.

4.3.1　骨干网络

为了测试骨干网络对 K-Query 全景分割性能得影响，本文选用不同大小的 Swin[30] 网络作为骨干网络进行测试，通常情况下模型越大，其泛化能力越强，如果性能瓶颈不在 K-Query 的查询向量，则使用更强的骨干网络将会得到更优的分割效果. 测试结果如表 3 中的第 3、4、5、6 列所示，实验表明相比 Res50 网络，Swin 网络具有更强表征能力，且随着模型表征能力的增强，其 PQ 分割结果也在继续增加，分别为 64.8%、66.7%、66.7%、67.2%，趋势对比如图 7 所示. 该结果表明，K-Query 在全景分割上，其动态查询过程不存在明显的性能瓶颈.

表 3　各测试条件下的全景分割结果对比

行骨干网络	PQ	PQ^{th}	PQ^{st}	备注
Res50	63.2	56.2	68.3	原始 K-Query
Res50	63.0	55.8	68.3	无边界过滤
Swin-tiny	64.8	58.2	69.6	—
Swin-small	66.7	60.6	71.1	—
Swin-base	66.7	61.1	70.8	—
Swin-large	67.2	61.2	71.2	—

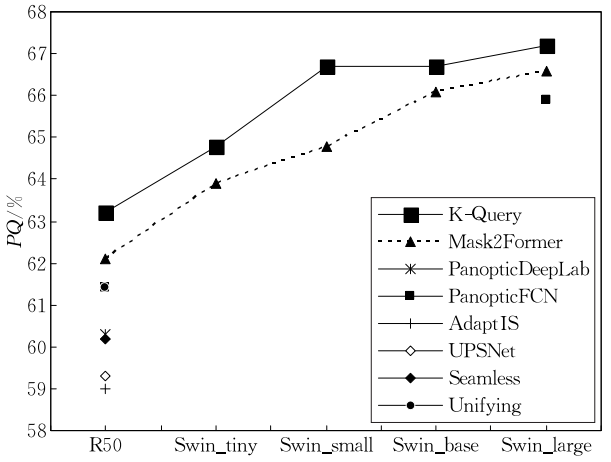


图 7　不同工作在 cityscape val 上的 PQ 值

4.3.2　关键点数量

在默认的 K-Query 配置中，关键点默认最大提取个数 $L=512$，为了测试该值对全景分割的影响，本实验对不同 L 值下的 K-Query 进行了测试. 测试结果如图 8 所示，随着最大查询数取值上限的增加，其分割性能也在增加，但大于一定阈值（$L=128$）后，PQ 值不在变化.

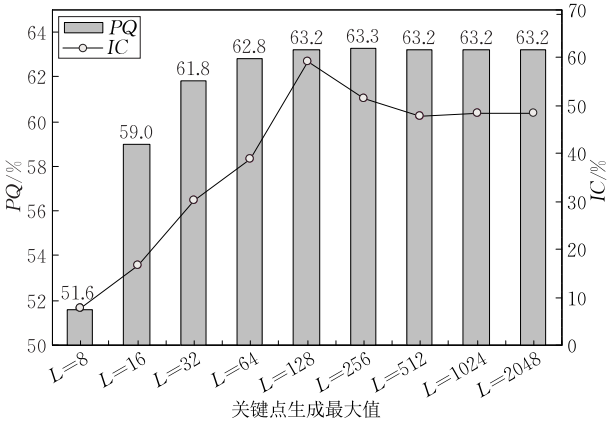


图 8　PQ 和 IC 随 L 的变化趋势

为了进一步验证实例的关键点提取效果，本文按如下公式定义“实例覆盖率”(Instance Coverage，IC).

$$V(I)=\begin{cases}0, & I\cap K<1\\ 1, & I\cap K\geq 1\end{cases} \tag{11}$$

$$IC=\frac{\sum_{i=0}^{N}V(GT\_INS[i])}{N} \tag{12}$$

式(11)表示实例是否被“关键点”匹配上，其中 I 表示目标实例掩码，K 为关键点集合，它们交集不为空表示实例与关键点匹配上，标记为有效实例. 式(12)中 GT_INS 表示目标实例集合，“实例覆盖率”IC 即有效实例在所有实例中的占比，占比越高表示关键点提取越好.

如图 8 中副轴所示，指标 IC 也随着 L 地增加在不断增加，当大于一定阈值（$L=128$）后，也不在增加，覆盖率保持在 50%左右. 覆盖率不高可能由两方面的原因导致：(1) 模型生成的实例嵌入与目标预期相差较大；(2) 行列式聚类算法在实例嵌入编码上表现不够好. 该实验表明，K-Query 中的实例嵌入向量编码方法和聚类算法还存在较大的优化空间.

4.3.3　对象边界点

在 K-Query 的关键点提取过程中，首先利用卷积操作，根据像素点嵌入之间的距离对目标图片中待分割对象的边界进行了过滤. 为了检测边界点对关键点提取的影响，本文在不进行边界过滤的情况下对其全景分割性能进行了测试，其结果如表 3 中的第 2 行所示，相对原始性能，PQ 值下降 0.2 个点，PQ^{th} 下降 0.4 个点. 该测试结果表明在目标对象

的边界部分,其像素点的嵌入向量存在比较大的波动,导致关键点提取效果变差,进而导致全景分割结果变差.

4.3.4 聚类算法

本文提出的"行列式"快速聚类算法是 K-Query 中的关键,为了验证该聚类算法的速度和内存开销,本文对其进行了对比测试,测试结果如图 9 和图 10 所示. 图中 Mean Shift 方法是常见的、最具代表性的聚类方法,本文选用 Pytorch-GPU 版本的实现作为对比目标. 其他常见的无需指定"簇"个数的聚类算法通常用于传统数据处理,在深度学习中不常用,目前没有适用于 Pytorch 等深度学习框架的实现. 本实验利用两种方法分别处理长度从 1 k 到 22 k 的二维数据,然后对比它们的时间和内存消耗.

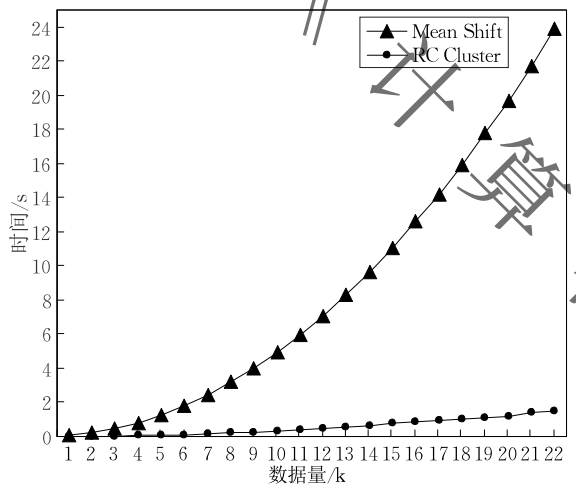


图 9 聚类方法速度对比

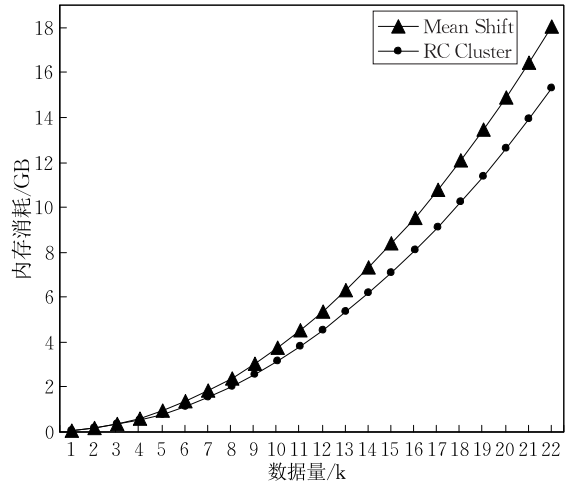


图 10 聚类方法 GPU 内存消耗对比

从图 9 中可知,随着数据规模不断扩大,Mean Shift 方法处理数据的时间成指数形式变长,而本文提出的"行列式"快速聚类方法的数据处理时间线型增加. 通过对比可以发现,本文提出的 RC Cluster 聚类方法相对 Mean Shift 的处理速度,在 1 k 到 22 k 长度的数据上,平均提升了 14.7 倍. 本文认为导致该结果的直接原因是本文提出的基于行列式的聚类方法操作简单,适合 Pytorch 等张量计算框架,可以更好地发挥 GPU 等设备的并行能力.

如图 10 所示,对于内存消耗,随着数据规模的增大,Mean Shift 和 RC Cluster 方法消耗的内存都随之显著增大,且内存增大的程度大于数据增大的程度,当数据规模大于 10 k 后,它们内存消耗都大于 3 GB,RC Cluster 内存平均消耗为 Mean Shift 的 88%. 该实验表明,聚类方法在大规模数据下消耗的内存过大,应用到深度学习还需要进一步优化.

K-Query 中的聚类算法只是用来生成实例嵌入中的"簇",不需要基于其进行预测,且不同聚类方法适用于不同的数据和场景. 因此在一般意义上的聚类预测精度上,本文提出的算法相比其他算法没有可比性. 此外,针对 K-Query 若采用 Mean Shift 算法提取"关键点"会由于训练时间过长、GPU 内存消耗过大而无法有效完成模型训练,因此其在最终全景分割精度上的表现需要在将来条件容许的情况下进行进一步验证.

4.3.5 性能开销

在深度学习领域,性能开销主要表现在内存开销和时间开销两个方面. 本文在 K-Query 的实现上,是基于目前开源且具有代表性的静态查询方法 Mask2Former,在其之上增加了动态查询部分. 因此对比 K-Query 和 Mask2Former 的性能开销,就能得出所增加部分导致的性能差异.

在模型参数设置方面,两种方法都采用 Mask2-Former 的原始配置. 训练和推理时的 batch_size 设置为 16,训练时的图片预处理 crop_size 设为 1024×512. 由于推理时不对图片进行裁剪,推理过程中输入图片的分辨率为 2048×1024,大于训练时的图片输入.

如图 11 所示,在推理过程中,两种方法在 8 块 GPU 上的内存消耗都不均衡,差异较大. 它们的平均内存消耗(AVG)为 20.2 GB 和 15.4 GB,除第一块 GPU 外,K-Query 在其他 GPU 上的内存消耗都大于 Mask2Former. 该实验表明,在推理过程中,K-Query 中的"基于聚类的动态查询过程"平均增加了 31%的 GPU 内存开销.

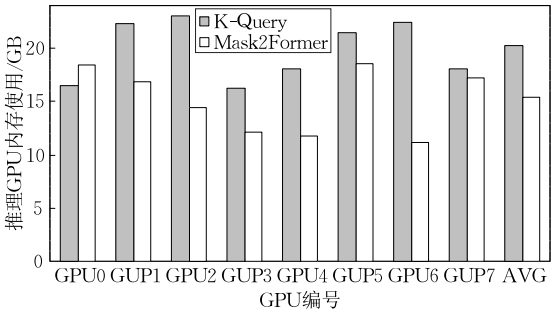


图 11　推理时 GPU 内存消耗对比

两种方法在训练阶段的内存消耗如图 12 所示，K-Query 和 Mask2Former 的平均(AVG)内存消耗为 9.7GB 和 9.1GB，该数据表明“动态查询”过程在训练阶段的 GPU 内存开销增加了 6.6%. 从图中可以看出，相对推理过程，模型在内存消耗上相对小，GPU 之间内存消耗的波动也相对较小. 本文认为，该现象主要是训练过程中的输入数据小于推理时的数据所导致(推理数据输入量是训练时的 4 倍).

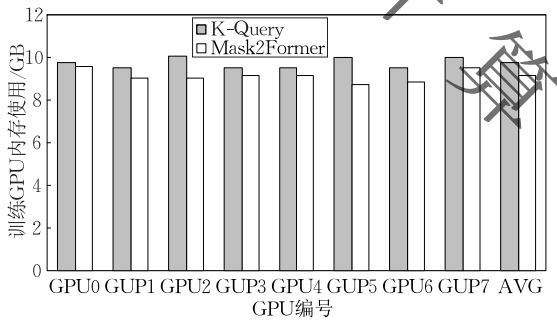


图 12　训练时 GPU 内存消耗对比

K-Query 和 Mask2Former 方法在相同参数配置下平均每次训练和推理迭代的执行时间如图 13 所示. 由于训练过程中需要进行反向梯度传播，其时间消耗远大于推理阶段. K-Query 相比 Mask2Former 在训练和推理阶段，时间消耗分别增加了 378.4 ms 和 388.7 ms. K-Query 相对 Mask2Former 在结构上只增加了三层卷积嵌入和聚类算法，根据经验[28]，该卷积部分开销小，结合 4.3.4 节可确定 K-Query 的主要时间开销为其中的聚类过程.

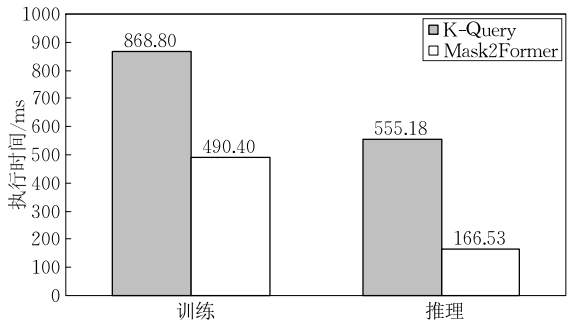


图 13　模型执行时间对比

综上所述，K-Query 的“动态查询特征”能带来全景分割质量的显著提升，并存在进一步的提升空间，但在目前基于聚类的实现上存在一定的开销，需要进行进一步地优化.

5　问题讨论

本节的问题讨论期望能够对相关研究工作带来一定的启发和参考，具体开放性问题如下：

统一表示方法. 在图片全景分割任务中，不同类型的方法有不同的实例表示，例如在自顶而下的方法中，通常使用“边界框”进行实例表示，在自下而上的方法中，例如工作 Panoptic-DeepLab，采用质心一类的关键点进行实例表示，而在基于查询的方法中，采用查询向量进行表示，它们之间并不兼容. 是否存在一种适用于这三大类的通用实例表示方法，可以提升所有全景分割方法的性能？

基于学习的关键点提取. 在 K-Query 方法中，利用传统聚类算法进行关键点的提取，该过程是否可以通过深度学习的方式进行替代，进而免去复杂的提取流程. 例如通过 FCN 直接预测关键点位置，或者预测待分割对象的某些物理特征，然后基于特征进行关键点提取.

融合传统图像算法. 基于深度学习的方法是目前计算机图像领域的代表性方法，但该方法严重依赖训练数据，另一方面在传统图形处理领域有大量的图像处理算法，例如分水岭二值图像分割方法、距离变换、边缘检测等. 是否有可能把传统图像处理方法和基于机器学习的方法进行有机结合，发挥出它们共同的优点，进一步提升图像分割效果和处理速度.

其他领域应用. 本文提出的 K-Query 全景分割方法以及其中包含的快速聚类方法，除了应用在图像分割领域，是否也可以应用在其他领域，例如目标检测、物体追踪等.

6　结束语

当前最新的基于查询的图像全景分割方法取得了很好的效果，但他们采用静态的查询方法进行目标物体掩码和类别信息的提取，该过程导致模型训练时查询个数不好确定，静态查询表示能力不足引发全景分割效果不够好等问题. 为此本文提出了一种基于关键点的动态查询全景分割方法 K-Query，该方法利用实例嵌入和基于“行列”的快速聚类方法

为待查询目标选取关键点,然后利用关键点的“嵌入”向量和位置作为 Transformer-Decoder 部分的查询输入.该方法避免了上述静态查询方法带来的问题,经过测试,K-Query 在典型数据集 Cityscapes 和MS COCO panoptic 2017的验证集上取得了 63.2% 和 52.9%的 *PQ* 结果,相比当前最好的方法,K-Query 分别提升 1.1 和 1.0 个点.该方法在处理速度和精度上还存在优化空间,在未来工作中我们将进行更深入地探索.

致 谢 感谢所有审稿人员在百忙之中对本文提出的建议和帮助!

参 考 文 献

[1] Kirillov A, He K, Girshick R, et al. Panoptic segmentation//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. California, USA, 2019: 9404-9413

[2] Thoma M. A survey of semantic segmentation. arXiv preprint arXiv:1602.06541, 2016

[3] Bai M, Urtasun R. Deep watershed transform for instance segmentation//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Hawaii, USA, 2017: 5221-5229

[4] Chen X, Wang J, Hebert M. PanoNet: Real-time panoptic segmentation through position-sensitive feature embedding. arXiv preprint arXiv:2008.00192, 2020

[5] Bonde U, Alcantarilla P F, Leutenegger S. Towards bounding-box free panoptic segmentation//Proceedings of the DAGM German Conference on Pattern Recognition. Tübingen, Germany, 2020: 316-330

[6] Xiong Y, Liao R, Zhao H, et al. UPSNet: A unified panoptic segmentation network//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. California, USA, 2019: 8818-8826

[7] Li Y, Chen X, Zhu Z, et al. Attention-guided unified network for panoptic segmentation//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. California, USA, 2019: 7026-7035

[8] Liu H, Peng C, Yu C, et al. An end-to-end network for panoptic segmentation//Proceedings of the Computer Vision and Pattern Recognition. California, USA, 2019: 6172-6181

[9] Zhang W, Pang J, Chen K, et al. K-Net: Towards unified image segmentation//Proceedings of the Neural Information Processing Systems. Quebec, Canada, 2021: 10326-10338

[10] Cheng B, Misra I, Schwing A G, et al. Masked-attention mask transformer for universal image segmentation//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. New Orleans, USA, 2022: 1290-1299

[11] Cheng B, Collins M D, Zhu Y, et al. Panoptic-DeepLab: A simple strong and fast baseline for bottom-up panoptic segmentation//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Washington, USA, 2020: 12475-12485

[12] Kirillov A, Girshick R, He K, et al. Panoptic feature pyramid networks//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. California, USA, 2019: 6399-6408

[13] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Ohio, USA, 2014: 580-587

[14] Li Q, Qi X, Torr P H S, et al. Unifying training and inference for panoptic segmentation. arXiv preprint arXiv:2001.04982, 2020

[15] Mohan R, Valada A. EfficientPS: Efficient panoptic segmentation. International Journal of Computer Vision, 2021, 129(5): 1551-1579

[16] He K, Gkioxari G, Dollár P, et al. Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386-397

[17] Li J, Raventos A, Bhargava A, et al. Learning to fuse things and stuff. arXiv preprint arXiv:1812.01192, 2018

[18] Gao N, Shan Y, Wang Y, et al. SSAP: Single-shot instance segmentation with affinity pyramid//Proceedings of the International Conference on Computer Vision. Seoul, Korea, 2019: 642-651

[19] Yang T-J, Collins M D, Zhu Y, et al. DeeperLab: Single-shot image parser. arXiv preprint arXiv:1902.05093, 2019

[20] Zhang R, Tian Z, Shen C, et al. Mask encoding for single shot instance segmentation//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Washington, USA, 2020: 10226-10235

[21] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need//Proceedings of the Neural Information Processing Systems. California, USA, 2017: 1-11

[22] Cordts M, Omran M, Ramos S, et al. The Cityscapes dataset for semantic urban scene understanding//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Nevada, USA, 2016: 3213-3223

[23] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers//Proceedings of the European Conference on Computer Vision. Glasgow, UK, 2020: 213-229

[24] Neven D, De Brabandere B, Proesmans M, et al. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. California, USA, 2019: 8837-8845

[25] Lin T-Y, Maire M, Belongie S, et al. Microsoft COCO: Common objects in context//Proceedings of the European Conference on Computer Vision. Zurich, Switzerland, 2014: 740-755

[26] Chen L-C, Papandreou G, Kokkinos I, et al. Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062, 2014

[27] Softmax. https://deepai.org/machine-learning-glossary-and-terms/softmax-layer, 2019

[28] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Nevada, USA, 2016: 770-778

[29] Sun K, Xiao B, Liu D, et al. Deep high-resolution representation learning for human pose estimation//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. California, USA, 2019: 5693-5703

[30] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows//Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, Canada, 2021: 10012-10022

[31] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020

[32] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Hawaii, USA, 2017: 2117-2125

[33] Chen L-C, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587, 2017

[34] Shrivastava A, Gupta A, Girshick R. Training region-based object detectors with online hard example mining//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Nevada, USA, 2016: 761-769

[35] Zhao S, Wang Y, Yang Z, et al. Region mutual information loss for semantic segmentation//Proceedings of the Neural Information Processing Systems. British Columbia, Canada, 2019: 1-11

[36] Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017

[37] Hong Y, Pan H, Sun W, et al. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. arXiv preprint arXiv:2101.06085, 2021

[38] Zhang X, Zhou X, Lin M, et al. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. arXiv preprint arXiv:1707.01083, 2017

[39] Bolya D, Zhou C, Xiao F, et al. YOLACT: Real-time instance segmentation//Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, Korea, 2019: 9157-9166

[40] Lee Y, Park J. CenterMask: Real-time anchor-free instance//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Washington, USA, 2020: 13906-13915

[41] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015

[42] De Brabandere B, Neven D, Van Gool L. Semantic instance segmentation with a discriminative loss function. arXiv preprint arXiv:1708.02551, 2017

[43] Hartigan J A, Wong M A. A *k*-means clustering algorithm. Journal of the Royal Statistical Society, 1979, 28(1): 100-108

[44] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(5): 603-619

[45] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. Knowledge Discovery and Data Mining, 1996, 96(34): 226-231

[46] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016

[47] Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style high-performance deep learning library//Proceedings of the Neural Information Processing Systems. British Columbia, Canada, 2019: 1-12

[48] Chen T, Li M, Li Y, et al. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274, 2015

[49] detectron2. https://github.com/facebookresearch/detectron2, 2019

[50] Hou R, Li J, Bhargava A, et al. Real-time panoptic segmentation from dense detections//Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. Washington, USA, 2020: 8523-8532

[51] Porzi L, Bulò S R, Colovic A, et al. Seamless scene segmentation//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, USA, 2019: 8277-8286

[52] Sofiiuk K, Barinova O, Konushin A. AdaptIS: Adaptive instance selection network//Proceedings of the EEE/CVF International Conference on Computer Vision. Seoul, Korea, 2019: 7354-7362

[53] Li Y W, Zhao H S, Qi X J, et al. Fully convolutional networks for panoptic segmentation. arXiv preprint arXiv:2012.00720, 2020

[54] Wang H Y, Zhu Y K, Adam H, et al. MaX-DeepLab: End-to-end panoptic segmentation with mask transformers//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA, 2021: 5463-5474

YAO Zhi-Cheng, Ph. D. candidate, engineer. His main research interests include computer architecture, computer vision, robot system, etc.

WANG Sa, Ph. D., associate professor, M. S. supervisor. His main research interests include computer architecture, cloud computing, operating system, etc.

BAO Yun-Gang, Ph. D., professor, Ph. D. supervisor. His main research interests include computer architecture, cloud computing, operating system, open-source chip, etc.

Background

Panoptic segmentation is a combination of semantic and instance segmentation, which gives semantic information and instance information to each pixel in the input picture. It is a key and hot research problem in computer vision and has important application value in automatic driving, robotics, etc. Common instance representation methods of panoptic are box-based, keypoint-based, and query-based. Those methods have good performance, but they suffer from insufficient expression ability in some specific scenarios.

The Box-based method, also known as the top-down method, is currently the most common method for instance representation, which uses the bounding box of the instance for instance representation. Specifically, it first generates a lot of proposals for each instance in the picture. Then predicts the bounding box (the offset to the boundary from the proposal) and instance category of the corresponding instance based on the proposal. Finally, it filters out the additional instances based on category confidence and the parie-wise box-IoU. This method is intuitive and effective but fails in certain scenarios. For example, when the bounding boxes of multiple instances of the same class overlap significantly, the box-based method will cause low-confidence instances to be lost.

The keypoint-based approach, often called the bottom-up method, utilizes the instance's inherent "point" property as the instance representation. For example, Panoptic-DeepLab uses the centroid as the instance representation. It predicts the centroid of each instance and the offset of the instance pixel to its centroid. And then, it calculates the distance from the predicted position of each pixel to all predicted centroids and uses the id of the nearest centroid from the "pixel prediction position" as the mask number of the pixel. It finally combines the semantic segmentation information to obtain the panoptic segmentation results. The segmentation method based on key points usually adopts a one-stage model, which is simple and fast. However, those key point-based methods also have insufficient expression ability in specific scenarios. For example, when the key points of two or more instances are similar or the same, the method will mark multiple instances as one.

The query-based method uses a query as the instance representation. This method is implicit and does not choose specific instance visible properties as the representation so that it can label the overlapped instances. This method has a good segmentation effect and can uniform the segmentation process of semantic and instance segmentation. However, the existing query-based methods all use static queries, in which the query is not associated with the instance in the input image. So, there is a problem in that the number of queries is difficult to determine. If the query number is too large, the memory and computing consumption is too high, while the expression ability is limited if it is too small.

We propose a key point-based query method named K-Query for the above static-query issue. The queries of K-Query are dynamically derived from the input picture's corresponding instances and have a better segmentation performance. The test results show that it improves the *PQ* by 1 point compared to the state-of-the-art query-based method.

This work was partially supported by the National Natural Science Foundation of China (Nos. 62090020, 61672499), the Youth Innovation Promotion Association of Chinese Academy of Sciences (No. 2013073), and the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDC05030200).