题目

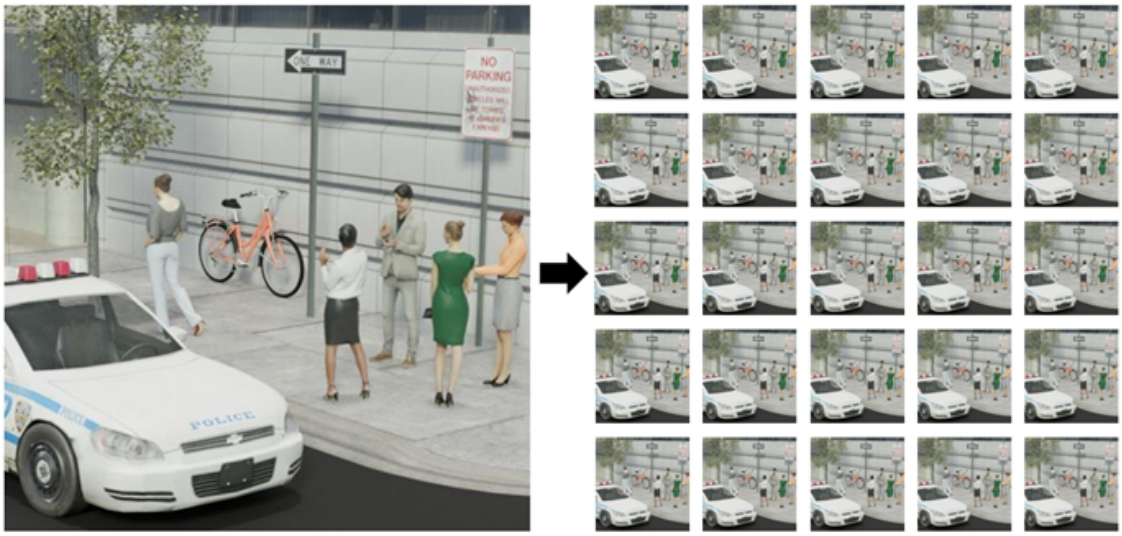
摘要

光场具有比传统光照模型更高维的特性，记录了场景中每一点光的强度和方向信息。在内存有限的情况下，光场语义分割问题十分棘手，主要表现在难以在保持单个视图上下文信息的同时充分利用不同视图之间的关系。本课题旨在复现一种新型的光场语义分割网络 LF-IENet 并对其进行改进。该网络使用两种不同的方式来挖掘从周围视图到分段中心视图的互补信息。一是隐式特征集成，利用注意力机制计算视图间和视图内的相似性，从而调制中心视图的特征。二是显式特征传播，利用视差估计器估计视差，在视差的引导下，将不同角度视图的特征直接扭曲到中心视图。这两种方法相互补充，共同实现光场跨视点的信息互补。本课题使用真实世界数据集 UrbanLF-Real 和合成光场数据集 UrbanLF-Syn。经过改进后，该方法合成的语义分割标签的平均交并比 mIoU 最高达到了 83.24%，比原文高出 0.53%。该方法在真实与虚拟数据集中都取得了优异的性能，证实了 LF-IENet 架构的有效性。

关键词：光场语义分割；注意力机制；视差估计；特征集成；特征传播

1 引言

光场语义分割作为一个新兴的研究领域，可以借鉴通用语义分割的基础，并且在光场数据的处理和应用上提出新的方法和框架。基于语义分割的基础，有三种有效利用光场信息的处理方式。一是将光场组织成二维宏像素图像是一个简单而有效的解决方案。这样的处理方式使得光场数据可以直接应用于现有的图像语义分割方法，从而获得良好的结果。二是将光场转换成与视频序列形式相似的 SAI 阵列，这为视频语义分割提供了一个可行的方案。通过将光场转换为 SAI 阵列，可以利用现有的视频语义分割方法来处理光场数据。三是由于深度和视差可以互换使用，所以光场中包含的视差信息可以用于 RGB-D 语义分割。这种方法可以充分利用光场数据中的深度信息，并将其与 RGB 图像结合起来进行语义分割任务。



(a) 中心视图　　(b) 角分辨率为5*5的参考视图

图 1. 角分辨率为 5 × 5 的局部参考视图

尽管以上方法可以在一定程度上处理光场数据并进行语义分割，但直接将光场视为二维图像或视频序列可能无法充分发挥其优势。因此，有必要设计专门针对光场数据的语义分割框架。这个框架应该考虑以下几个方面。首先是角度信息的利用。光场数据包含丰富的角度信息，这些信息可以提供更多的场景视角和深度信息。设计框架时应考虑如何有效地利用这些角度信息，以提高语义分割的准确性和鲁棒性。其次是如何利用光场特有的时间信息。光场数据中包含时间信息，这与视频序列不同。这种时间信息可能与光场中的运动或变化相关联，因此设计框架时应该充分利用这种时间信息来增强语义分割的能力。第三是需要对光场数据进行深入处理。基于 RGB-D 的方法通常只提取深度信息作为输入，而忽略了光场数据中更丰富的信息。设计框架时应该考虑对光场数据进行更深入的处理，包括深度、角度、时间等多方面信息的综合利用。

为了实现光场四维信息的整体提取，一种有效的方法是分别对每个 SAI 中的空间关系进行建模，然后沿角维进行所有视图之间的交互。这种建模机制通常用于光场的超分辨率研究 [13, 17] 和视差估计 [10, 12]，可以使用基于局部图像1的输入，因为超分辨率研究只需要考虑周围的像素，视差估计强调跨视图的像素匹配。然而，如图1所示，语义分割不能基于图中的局部信息分配正确的标签，它需要完整视图2中参考视图的上下文信息。因为独立的局部图像丢弃了有价值的上下文信息。另一种方法是沿着空间维度将所有 SAI 的大小调整到极低的尺度，以降低计算成本。然而，这样做会放弃对密集预测任务至关重要的分辨率和粒度，可能会影响最终的语义分割结果。

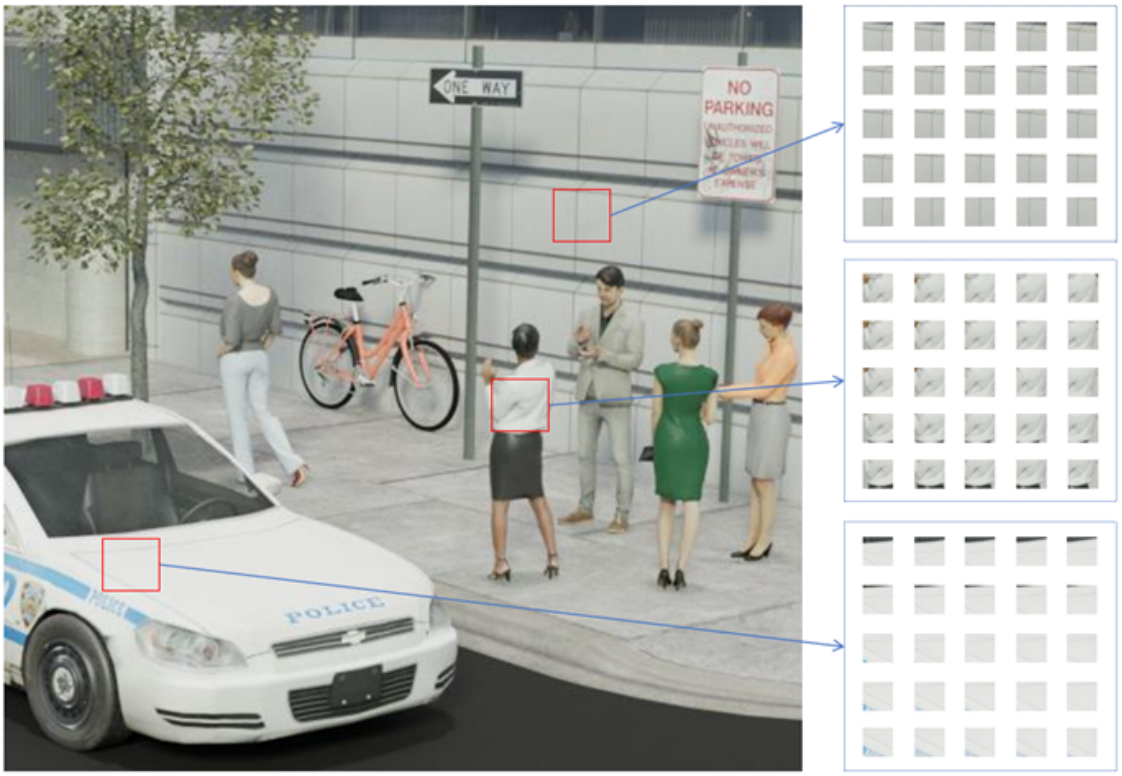


图 2. 角分辨率为 5 × 5 的光场输入示意图

针对以上问题，本文使用 LF-IENet 进行高效的光场语义分割。为了充分挖掘光场信息，LF-IENet 通过隐式分支和显式分支联合学习的方式对中心视图进行特征增强，显式分支显式传播参考视图的特征到中心视图，隐式分支使用注意力机制计算视图内和视图间的相似度进行特征集成。这两个分支相互补充，在保留视图上下文信息的同时，有效利用了光场所包含的不同视角信息。

2 相关工作

这里介绍通用语义分割方法、4D 光场语义分割方法以及在本文中使用到的光场视差估计方法和应用。

2.1 通用语义分割

本节介绍了图像语义分割、视频语义分割和 RGB-D 语义分割三种常见的语义分割方法。图像语义分割的目标是将图像中的不同元素（如道路、建筑、人、车辆等）进行像素级分类。视频语义分割则在此基础上，处理视频序列中的每一帧，并关注帧间的动态变化。RGB-D 语义分割利用了深度信息（除了 RGB 颜色信息，还包括每个像素的深度信息），能够更准确地处理遮挡问题和提升场景理解的精度。

在图像语义分割方面，传统方法和深度学习方法并存，目前深度学习方法是主流。 [6] 通过全卷积网络进行端到端的像素级预测，是深度学习语义分割的开创性工作。PSPNet [20] 通过并行自适应池化捕获多尺度上下文，DeepLab 利用空洞卷积和金字塔池化技术提升精度和效率。Mask R-CNN [4] 结合目标检测和语义分割，关注物体级别的分割。近期的 [20] 和

SegFormer [14] 等研究则通过变压器模型捕获全局上下文。视频语义分割的研究重点除了提高精度，还强调运行效率。提高分割效率的方法通过重用关键帧特征和光流传播，减少计算量并提高实时性；高精度分割方法则通过时间上下文信息的整合和新模块（如注意力机制、多尺度特征融合等）增强模型对复杂场景的理解，尽管可能增加计算开销。在 RGB-D 语义分割方面，深度信息的引入有助于区分颜色和纹理相似的物体。部分方法通过并行处理 RGB 图像和深度数据，再将输出特征进行融合，提升分割准确性。一些方法则设计特定的卷积层（如 S-Conv [2] 和 ShapeConv [1]），通过深度值来引导分割任务。此外，多任务学习策略 [11, 15] 也被应用于联合训练语义分割和深度估计，从而提升模型的鲁棒性和精度。

2.2 光场语义分割

光场视差估计是计算机视觉领域的一个重要研究方向，旨在利用光场信息来估计场景中物体的深度和形状。光场视差估计的目标是计算一个位移图，该位移图表示相邻视图之间的空间投影坐标变换。经过长期的研究和积累，已经提出了大量的方法 [3, 5, 7, 9, 10, 12, 16, 21] 来促进这一领域的发展。主要分为以下几类。常规方法和基于学习的方法：常规方法通常通过构建成本体积来衡量不同深度下像素的一致性，然后通过优化算法（如图割、Belief Propagation 等）来选择最佳的深度值。这种方法在处理光场视差估计问题时能够充分利用优化算法来提高准确性。而基于学习的方法则通过训练模型来预测和估计视差。通过训练深度神经网络等模型，可以在一定程度上提高视差估计的准确性和泛化能力。基于 epi 的方法和非基于 epi 的方法：基于 epi（极线）的方法利用极线几何来简化视差估计问题，减少计算复杂度；而非基于 epi 的方法则不依赖于极线几何，通常具有更灵活的应用场景和更广泛的适用性。全监督方法和无监督方法：全监督方法需要有标注的训练数据来进行模型训练，而无监督方法则尝试在没有标注数据的情况下进行视差估计，通常利用自监督学习或者无监督学习的方法。

2.3 光场视差估计

光场视差估计的目标是计算表示相邻视图之间空间投影坐标变换的位移图。经过长期的研究和积累，大量的方法 [8, 13, 14, 18, 19, 22] 被提出，推动了该领域的发展。它们可以分为常规方法和基于学习的方法、基于 EPI 的方法和非 EPI 的方法、全监督方法和无监督方法。

3 本文方法

3.1 本文方法概述

本章描述了用于光场语义分割的 LF-IENet。给定一个由 $U \times V$ 个 SAI 组成的、空间分辨率为 $H \times W$ 的 4D 光场 $L \in \mathbb{R}^{U \times V \times H \times W \times 3}$。只有中心视图图像 $L_{a_c} \in \mathbb{R}^{H \times W \times 3}$ 具有带注释的语义标签 $Y_{a_c} \in \mathbb{R}^{H \times W \times C_{cls}}$，其中 C_{cls} 是语义类别的总数。其余的 SAI 除了 L_{a_c} 外都标记为参考视图图像 $L_{a_r^i} \in \mathbb{R}^{H \times W \times 3}$ $(i = 1, \cdots, UV - 1)$。LF-IENet 在参考视图的帮助下预测中心视图的语义标签 $Y_{a_c} \in \mathbb{R}^{H \times W \times C_{cls}}$。这里 a 表示角度坐标 (u, v)，我们使用分布在角维上的正方形数组的 SAI 来实现语义分割，即 $U = V = A$。

3.2 网络框架

整个 LF-IENet 网络的框架如图3所示。它以整个光场为输入，输出中心视图的结果。网络中有两个分支。在隐式分支中，首先利用特征提取主干对包括中心视图在内的多幅视图图像进行处理，从输入数据中提取特征的核心部分，将输入数据转换为高层次的抽象特征表示，以供后续计算使用。然后通过自注意力机制和交叉注意力机制来增强特征表征的能力。自注意力机制旨在收集中心视图内的上下文信息，有助于捕捉输入数据内部的长距离依赖关系，而交叉注意力机制更多地关注跨视图的不一致区域，可以利用不同视图之间的相似性来补偿特征表示。在显式分支中，首先通过成熟的光场视差估计技术计算初始视差图。根据照片一致性假设，通过建立参考视图和中心视图之间的对应关系，根据估计的视差和相对角度位置传播参考试图的特征。最后，将两个分支 F_{im} 和 F_{ex} 的输出特征进行聚合，生成中心视图 F_{pred} 的最终特征，然后将其馈送到分割头中以获得预测的分割映射。

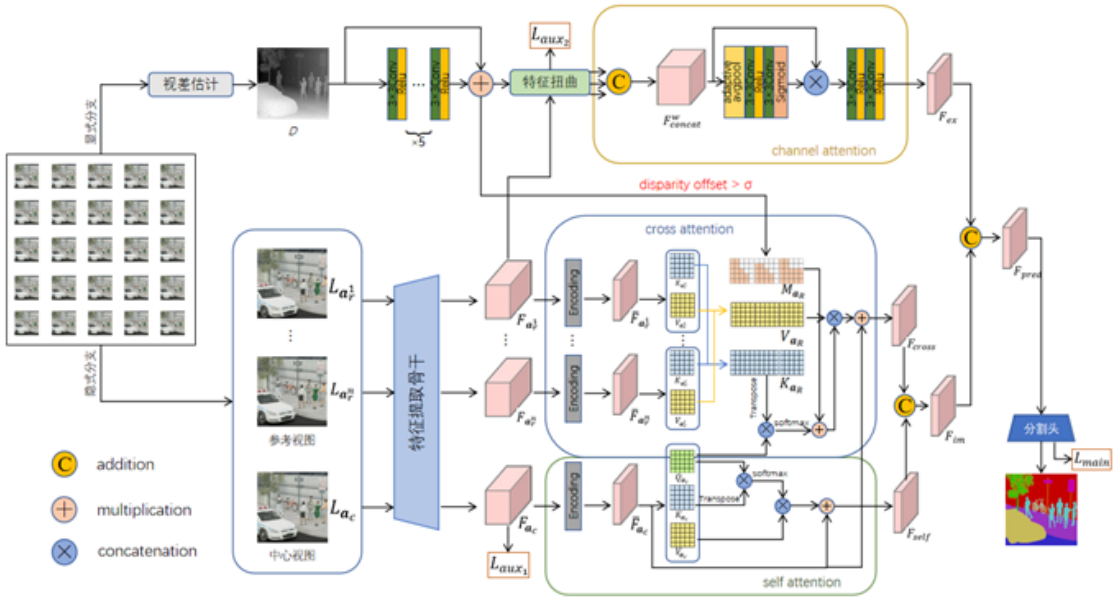


图 3. LF-IENet 的网络框架

3.3 隐式特征集成

根据光场图像超分辨的残差网络 [?]，我们从周围具有水平、垂直或对角线亚像素位移的参考视图图像中提取和融合光场图像的多尺度特征，并探索参考视图和中心视图的互补信息。将包括中心视图的 $\{L_{a_r^1}, \cdots, L_{a_r^n}, L_{a_c}\}$ 共 $n+1$ 幅图像馈送到权重相同的特征提取主干，提取特征 $\{F_{a_r^1}, \cdots, F_{a_r^n}, F_{a_c}\}$，每个特征的大小为 $\mathbb{R}^{c \times h \times w}$。经过不同的并行编码层，这些特征进一步压缩为 $\{\overline{F}_{a_r^1}, \cdots, \overline{F}_{a_r^n}, \overline{F}_{a_c}\}$，大小为 $\mathbb{R}^{c_v \times h \times w}$。然后我们生成中央视图图像的 $Q_{a_c} \in \mathbb{R}^{N \times c_q}$，$K_{a_c} \in \mathbb{R}^{N \times c_k}$，$V_{a_c} \in \mathbb{R}^{N \times c_v}$ 和每个参考图像的 $K_{a_r^i} \in \mathbb{R}^{N \times c_k}$，$V_{a_r^i} \in \mathbb{R}^{N \times c_v}$，其中 $c_q = c_k$，$N = hw$。最后利用自注意力机制融合压缩后的视图特征和经过激活函数处理后的强化特征，以增强中心视图特征，其表达式定义为公式（1）：

$$F_{\text{self}} = R\left(\text{softmax}(Q_{a_c} \cdot K_{a_c}^T) \cdot V_{a_c}\right) + \overline{F}_{a_c} \quad (1)$$

其中 $F_{\text{self}} \in \mathbb{R}^{c_v \times h \times w}$，$R$ 表示将大小为 $N \times c_v$ 的张量重塑为 $c_v \times h \times w$。

自注意力机制对提取视图内部的上下文信息具有重要作用，但其无法充分利用光场的角度信息。为了使参考视图图像更有效地补偿中心视图图像，交叉注意力机制应更多地关注视图之间差异明显的区域。如图（图 3）所示，图中黄色矩形和蓝色矩形分别包含同一车轮的两个部分，并且与绿色矩形中的车轮属于同种物体，但由于行人遮挡而造成的信息缺失使得该部分从不同视角观察所呈现出的差异较为明显。为了更有效的处理类似区域的特征，我们计算每个像素在参考视图中的视差值，这个视差偏移量反映了中心视图像素与参考视图对应像素之间的差异程度。并通过计算的视差为所有的参考视图生成一个注意力掩码 $M_{a_r^i} \in \mathbb{R}^{h\times w}$，其表达式为公式（2）：

$$M_{a_r^i} = \begin{cases} 1, & d \cdot \|a_r^i - a_c\|_2 > \frac{\sigma}{\text{stride}} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

其中，$d \in \mathbb{R}^{h\times w}$ 是中心视图的视差值，$\|\cdot\|_2$ 是欧式距离，σ 是阈值，$\text{stride} = \frac{H}{h}$ 表示的输出步幅。我们设定了一个阈值 $\frac{\sigma}{\text{stride}}$，如果像素的视差偏移量 $d \cdot \|a_r^i - a_c\|_2$ 超过阈值，则对应的注意力掩码值设为 1；否则设为 0。通过这样的方式我们标记出了与原视图视差较大的像素点，并使用交叉注意力机制进行后续处理。

然后，这里将所有参考视图的键映射、值映射和注意力掩码依次连接、排列、重塑，分别得到 $K_{a_R} \in \mathbb{R}^{N_R\times c_k}$，$V_{a_R} \in \mathbb{R}^{N_R\times c_v}$ 和 $M_{a_R} \in \mathbb{R}^{1\times N_R}$，其中 $N_R = nhw$。交叉注意力权值定义为公式（3）：

$$W_{\text{cross}} = \text{softmax}(Q_{a_c} \cdot K_{a_R}^T) + \frac{\omega}{N_R} \cdot M_{a_R} \quad (3)$$

其中 $W_{\text{cross}} \in \mathbb{R}^{N\times N_R}$，$\omega$ 为视差较大的像素的附加权值。我们已经事先计算好注意力掩码 $M_{a_r^i} \in \mathbb{R}^{h\times w}$，若某像素的注意力掩码值为 1，说明该像素在参考视图中与中心视图中的视差较大，该部分区域包含了丰富角度信息，因此我们为这些像素附加一个权重 $\frac{\omega}{N_R} \cdot M_{a_R}$，以便交叉注意力机制增强对这部分区域的关注，并弱化冗余信息的影响。

最后我们将 W_{cross} 和 V_{a_R} 相乘来整合角度关系，从而增强中心视图的特征表示。交叉注意力机制的计算公式为公式（4）：

$$F_{\text{cross}} = R(W_{\text{cross}} \cdot V_{a_R}) + \overline{F}_{a_c} \quad (4)$$

其中 $F_{\text{cross}} \in \mathbb{R}^{c_v\times h\times w}$。交叉注意力机制通过借助源自显式分支的估计视差为像素分配交叉注意力权值，有效地利用了不同视角之间的信息互补性，从参考视图的不一致区域获取更多中心视图的补偿信息。最后，将 F_{self} 和 F_{cross} 沿通道维度进行串联，就可以得到隐式分支 F_{im} 的输出特征。

3.4 显式特征传播

为了在特征空间中建立准确的参考视图和中心视图的几何对应关系，首要任务是准确地计算参考视图和中心视图的视差。由于缺乏准确的地面真实数据作为监督信号来训练模型，这里选择性能优越的传统方法 OAVC [?] 作为视差估计器来预测中心视图的视差值 D，并设定标签范围为 256。OAVC 在 CPU 上运行，有效地减轻了 GPU 的工作负载，并且显著缩短了计算时间。我们采用 5 个级联的 3×3 卷积层，通过局部跳过残差连接连续校准视差图 D，通过下采样操作，得到与 $F_{a_r^i}$ 相同分辨率的 $d \in \mathbb{R}^{h\times w}$，用于后续的特征传播。

根据四维光场结构，给定中心视图 a_c 视差 d，参考视图 a_r^i 中像素 $L(u_c, v_c, h_c, w_c)$ 的空间坐标 (h_r^i, w_r^i) 可通过公式（5）中的变换计算：

$$(h_r^i, w_r^i) = (h_c, w_c) + d(h_c, w_c) \cdot (a_c - a_r^i) \quad (5)$$

然后，我们依次将每个参考特征扭曲到中心视图，得到一个对齐的特征组 $\{F^w_{a_r^1}, \cdots, F^w_{a_r^n}\}$，每个特征组的大小为 $\mathbb{R}^{c\times h\times w}$。考虑到不同角度的参考视图对最终特征表示的影响程度不同，我们采用通道注意力机制来生成聚合对齐特征的权重。首先将所有平行对齐的特征沿通道维度进行拼接，其中每个通道包含来自不同参考视图的特征信息。然后采用自适应池化方法对空间信息进行压缩，降低特征的空间维度，使其更易于处理。通过卷积和激活操作获得通道权重并应用到对齐的特征组上，将更多的注意力集中在对齐特征组中具有重要信息的部分，减少对齐特征组中冗余信息的影响，使得最终特征更加紧凑和有效，提高了特征的表征能力。最后，通过两层卷积对特征通道进行压缩。整个过程可以描述为公式（6）、公式（7）和公式（8）：

$$F^w_{\text{concat}} = [F^w_{a_r^1}, \cdots, F^w_{a_r^n}] \quad (6)$$

$$W_{\text{channel}} = \phi\left(H_{1\times1}\left(H_{1\times1}\left(\text{AdaptPool}(F^w_{\text{concat}})\right)\right)\right) \quad (7)$$

$$F_{\text{ex}} = H_{3\times3}\left(H_{1\times1}\left(W_{\text{channel}} \cdot F^w_{\text{concat}}\right)\right) \quad (8)$$

其中 $[\cdot]$ 表示级联操作，$H_{s\times s}$ 表示核大小为 $s \times s$ 的卷积层，ϕ 表示 sigmoid 函数。F_{ex} 表示显式分支的输出特征，利用来自所有参考视图的有效信息增强中心视图的特征表示。

4 复现细节

4.1 与已有开源代码对比

本文参考发布在 CVPR2023 上的开源项目https://github.com/Congrx/LF-IENet，并基于该项目进行了微小的修改。

4.2 实验环境搭建

本仓库基于开源的 `mmsegmentation` 工具箱和视差估计器 `OAVC` 实现。以下是环境搭建的详细步骤：

4.3 依赖安装

确保系统中已安装以下依赖项：

- **Python**: 3.8.10
- **CUDA**: 10.2
- **PyTorch**: 1.8.0
- **Torchvision**: 0.9.0
- **MMCV**: 1.6.2

可以通过以下命令安装 PyTorch 和 Torchvision：

```
pip install torch==1.8.0+cu102 torchvision==0.9.0+cu102 -f https://download.pytorch.org/
```

安装 MMCV：

```
pip install mmcv-full==1.6.2 -f https://download.openmmlab.com/mmcv/dist/index.html
```

4.4 数据集准备

请参考 UrbanLF 获取 UrbanLF-Real 和 UrbanLF-Syn 数据集。
数据集的文件结构如下：

```
data
    UrbanLF_Real
        train
            Imagexxx
                u_v.png (1_1.png ~ 9_9.png)
                label.npy
                disparity_OAVC.npy
        val
        test
    UrbanLF_Syn
        train
            Imagexxx
                u_v.png (1_1.png ~ 9_9.png)
                5_5_label.npy
                5_5_disparity_OAVC.npy
        val
        test
```

将数据集放置在 ./data/UrbanLF/UrbanLF_Real 和 ./data/UrbanLF/UrbanLF_Syn 路径下。

4.5 获取视差结果

使用 OAVC 生成预测的视差结果：

```
cd OAVC
python main.py
```

4.6 训练模型

根据需求选择不同的配置进行训练。

4.6.1 在 UrbanLF-Real 数据集上训练:

- **ResNet-50 模型**:

```
CUDA_VISIBLE_DEVICES='0,1' python -m torch.distributed.launch --nproc_per_node
```

- **HRNet-48 模型**:

```
CUDA_VISIBLE_DEVICES='0,1' python -m torch.distributed.launch --nproc_per_node
```

4.6.2 在 UrbanLF-Syn 数据集上训练:

- **ResNet-50 模型**:

```
CUDA_VISIBLE_DEVICES='0,1' python -m torch.distributed.launch --nproc_per_node
```

- **HRNet-48 模型**:

```
CUDA_VISIBLE_DEVICES='0,1' python -m torch.distributed.launch --nproc_per_node
```

4.7 测试模型

根据需求选择不同的配置进行测试。

4.7.1 在 UrbanLF-Real 数据集上测试:

- **ResNet-50 模型**:

```
CUDA_VISIBLE_DEVICES='0' python test.py configs/lf/UrbanLF_Real/LF_IENet_r50-d8
```

- **HRNet-48 模型**:

```
CUDA_VISIBLE_DEVICES='0' python test.py configs/lf/UrbanLF_Real/LF_IENet_hr48_4
```

4.7.2 在 UrbanLF-Syn 数据集上测试：

- **ResNet-50 模型**：

```
CUDA_VISIBLE_DEVICES='0' python test.py configs/lf/UrbanLF_Syn/LF_IENet_r50-d8_
```

- **HRNet-48 模型**：

```
CUDA_VISIBLE_DEVICES='0' python test.py configs/lf/UrbanLF_Syn/LF_IENet_hr48_48
```

测试结果将保存在以下路径：

- 分割结果：`numpy_res/real`（UrbanLF-Real）和 `numpy_res/syn`（UrbanLF-Syn）
- 定性结果：`img_res`

4.8 创新点

在本章中，我们主要介绍对 LF-IENet 网络的改进和优化。LF-IENet 网络利用注意力机制和视差估计来增强中心视图的特征表示。注意力机制在选择性关注重点区域和减弱冗余信息的影响方面具有显著作用。整个网络通过自注意力机制提取上下文信息，并使用交叉注意力机制挖掘光场的角度信息以补偿中心视图。视差估计在显式分支中显著影响特征传播的质量，并在隐式分支中用于指导生成交叉注意力掩码。

结合实验数据，我们对 LF-IENet 的网络结构进行了两点改进：一是优化交叉注意力权重的计算方法，使其充分利用视差估计提供的角度信息；二是对参考视图引入自注意力模块，先对参考视图的特征进行强化后再使用交叉注意力机制进行处理。

4.9 优化交叉注意力权重的计算方法

在 3.2 节中提到，为了更有效地处理类似区域的特征，需要计算每个像素在参考视图中的视差值，这个视差偏移量反映了中心视图像素与参考视图对应像素之间的差异程度。并通过计算的视差为所有的参考视图生成一个注意力掩码 $M_{a_r^i} \in \mathbb{R}^{h\times w}$，其表达式为公式（10）：

$$M_{a_r^i} = \begin{cases} 1, & d \cdot \|a_r^i - a_c\|_2 > \frac{\sigma}{\text{stride}} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

其中，$d \in \mathbb{R}^{h\times w}$ 是中心视图的视差值，$\|\cdot\|_2$ 是欧式距离，σ 是阈值，$\text{stride} = \frac{H}{h}$ 表示输出步幅。

然后，将所有参考视图的键映射、值映射和注意力掩码依次连接、排列、重塑，分别得到 $K_{a_R} \in \mathbb{R}^{N_R\times c_k}$，$V_{a_R} \in \mathbb{R}^{N_R\times c_v}$ 和 $M_{a_R} \in \mathbb{R}^{1\times N_R}$，其中 $N_R = nhw$。交叉注意力权值定义为公式（11）：

$$W_{\text{cross}} = \text{softmax}(Q_{a_c} \cdot K_{a_R}^T) + \frac{\omega}{N_R} \cdot M_{a_R} \tag{2}$$

其中 $W_{\text{cross}} \in \mathbb{R}^{N \times N_R}$，$\omega$ 为视差较大的像素的附加权值。

分析 W_{cross} 的计算过程，可以发现，原方法使用一个阈值 $\frac{\sigma}{\text{stride}}$ 将整个视图区域分为两部分，一是视差距离大于 $\frac{\sigma}{\text{stride}}$ 的不一致区域，二是视差距离小于 $\frac{\sigma}{\text{stride}}$ 的相似区域。这种划分方法只需设置一个阈值，就可以将超出阈值范围的位置的权重统一调整为同一个值，操作简单。但很显然，由于所有超出阈值范围的位置都被设置为同一个权重值，不能很好地区分不同位置的重要性，其精细度较低。

为了灵活处理视差距离对补偿中心视图产生的影响，需要考虑更为合理的计算方式。考虑到视差距离对补偿中心视图的影响程度是随着视差距离的增加而增加的，可以有两种方式描述这种变化特征：一是建立距离和权重的线性函数，将点 $(\frac{\sigma}{\text{stride}}, \frac{\omega}{N_R} \cdot M_{a_R})$ 作为函数的固定点，建立过该点的线性函数：权重 = 变化速率 * 视差距离。但该方法的实现需要进行大量尝试并选择合适的递增速率，实现起来难度较高，因此选择第二种方式表达这种变化特征，即把视差距离与权重之间的关系描述成梯度变化，在原表达式的基础上，设置阈值梯度，并赋予随视差距离递增的权重。其表达式为公式（12）：

$$M_{a_r^i} = \begin{cases} 0.25, & d \cdot \|a_r^i - a_c\|_2 > 0.25 \times \frac{\sigma}{\text{stride}} \\ 0.5, & d \cdot \|a_r^i - a_c\|_2 > 0.5 \times \frac{\sigma}{\text{stride}} \\ 1, & d \cdot \|a_r^i - a_c\|_2 > \frac{\sigma}{\text{stride}} \\ 2, & d \cdot \|a_r^i - a_c\|_2 > 2 \times \frac{\sigma}{\text{stride}} \\ 4, & d \cdot \|a_r^i - a_c\|_2 > 4 \times \frac{\sigma}{\text{stride}} \end{cases} \tag{3}$$

经过实验证明，根据不同的视差距离范围设定不同的权重值，可以更精细地对模型不同位置的关注程度进行调整，提高了处理特定距离范围特征的能力，能够更好地保留不同位置的关键信息。

4.10 引入自注意力模块强化参考视图特征

在隐式分支中，LF-IENet 首先利用特征提取主干对包括中心视图在内的多幅视图图像进行处理，从输入数据中提取特征的核心部分，将输入数据转换为高层次的抽象特征表示，然后输送到交叉注意力模块中挖掘光场信息。由于交叉注意力机制利用不同视图之间的相似性来补偿中心视图的特征表示，因此能否准确地捕捉到参考视图与中心视图中的不一致区域是交叉注意力模块要解决的重要问题。本文认为先前针对该部分的处理有两点不足：

其一，基于视差引导的交叉注意力权值计算十分依赖于视差估计的精度，视差估计的精度直接影响了不同视图之间的空间对齐程度。如果视差估计不准确，导致特征在不同视图之间没有良好的对应关系，那么交叉注意力机制计算的权值就会受到影响，可能无法准确地捕获视图间的语义关联。视差值通常用于控制权值的分配，例如在 4.1 中提出的根据视差大小调整权值的大小或范围。如果视差估计不准确，这种权值控制就会失效或产生错误的结果，影响模型的性能。因此在实际应用时，视差的作用是十分有限的。

其二，在处理视差距离和交叉注意力权值之间的联系时，原网络采用单一阈值计算交叉注意力掩码，经过 4.1 的优化后本文设置了阈值梯度来进行更精细的处理，但这两种方法都无法准确地表达视差距离与权值中的数学关系。若要在实际操作中用各种函数来拟合视差距离和交叉注意力权值之间的关系变化，这需要大量的实验和数据作为支撑，难以实现。因此准确表达这一关系在实际应用中是十分困难的。

在 4.1 节中提到，通过设置视差距离梯度来区分不同视差距离范围对补偿中心视图的重要程度，使交叉注意力机制能够更准确的处理不一致区域。本节在引入自注意力模块强化参考视图特征后，再通过交叉注意力模块进行处理。

将包括中心视图的 $\{L_{a_r^1}, \cdots, L_{a_r^n}, L_{a_c}\}$ 共 $n+1$ 幅图像馈送特征提取主干，提取特征 $\{F_{a_r^1}, \cdots, F_{a_r^n}, F_{a_c}\}$，每个特征的大小为 $\mathbb{R}^{c \times h \times u}$。经过并行编码层，这些特征进一步压缩为 $\{\bar{F}_{a_r^1}, \cdots, \bar{F}_{a_r^n}, \bar{F}_{a_c}\}$，大小为 $\mathbb{R}^{c_v \times h \times w}$。然后生成中央视图图像的 $Q_{a_c} \in \mathbb{R}^{N \times c_q}$, $K_{a_c} \in \mathbb{R}^{N \times c_k}$, $V_{a_c} \in \mathbb{R}^{N \times c_v}$ 和每个参考图像的 $Q_{a_r^i} \in \mathbb{R}^{N \times c_q}$, $K_{a_r^i} \in \mathbb{R}^{N \times c_k}$, $V_{a_r^i} \in \mathbb{R}^{N \times c_v}$，其中 $c_q = c_k$, $N = hw$。然后利用自注意力机制强化压缩后的中心视图特征和参考视图特征，其表达式为公式（13）和公式（14）：

$$\bar{F}_{\text{self}} = R(\text{softmax}(Q_{a_c} \cdot K_{a_c}^T) \cdot V_{a_c}) + \bar{F}_{a_c} \tag{4}$$

$$\bar{F}_{\text{self}_{a_r^i}} = R(\text{softmax}(Q_{a_r^i} \cdot K_{a_r^i}^T) \cdot V_{a_r^i}) + \bar{F}_{a_r^i} \tag{5}$$

其中 $\bar{F}_{\text{self}} \in \mathbb{R}^{c_v \times h \times w}$，$\bar{F}_{\text{self}_{a_r^i}} \in \mathbb{R}^{c_v \times h \times w}$, R 表示将大小为 $N \times c_v$ 的张量重塑为 $c_v \times h \times w$。经过该步处理后得到更为准确和丰富的参考视图特征 $\{\bar{F}_{\text{self}_{a_r^1}}, \cdots, \bar{F}_{\text{self}_{a_r^n}}, \bar{F}_{a_c}\}$。为了将强化后的特征输送到交叉注意力模块中进行处理，这里需要重新生成参考视图的键映射 $K_{a_r^i} \in \mathbb{R}^{N \times c_k}$ 和值映射 $V_{a_r^i} \in \mathbb{R}^{N \times c_v}$，其中 $c_q = c_k$, $N = hw$。

然后，将所有参考视图的键映射、值映射和注意力掩码依次连接、排列、重塑，分别得到 $K_{a_R} \in \mathbb{R}^{N_R \times c_k}$，$V_{a_R} \in \mathbb{R}^{N_R \times c_v}$ 和 $M_{a_R} \in \mathbb{R}^{1 \times N_R}$，其中 $N_R = nhw$。随后我们按照原网络的方法或优化后的方法计算交叉注意力权值 W_{cross} 和交叉注意力 F_{cross}。并将 F_{self} 和 F_{cross} 沿通道维度进行串联，就可以得到隐式分支 F_{im} 的输出特征。

实验证明，通过先后使用自注意力机制和交叉注意力机制对参考视图进行处理，可以逐步提升数据表示的丰富程度和模型对数据关系的理解能力。自注意力机制主要关注参考视图内部的关系，通过视图内部的相似性强化特征；而交叉注意力机制则关注不同视图之间的关联，二者结合可以使模型更全面地理解和处理参考视图所包含的光场信息。这种处理方式有效提高了隐式分支的学习能力，经过隐式分支进行特征集成后的输出特征 F_{im} 具有更加丰富且准确的特征表达，有效提升了整个 LF-IENet 的细节处理能力和分割性能。

5 实验结果分析

本课题使用的 UrbanLF 数据集包含两个子集：如图4，UrbanLF-Real 由 824 个类似的真实世界样本组成，这些样本是从现实世界的场景中收集的。这些样本提供了丰富的信息，可用于训练和评估算法对真实世界环境的理解能力。如图5，UrbanLF-Syn 包括 250 个类似的合成样本，它们是通过合成方法生成的，通常用于补充真实数据的不足，或者用于测试算法在非真实数据上的泛化能力。每个样本由 81 个 SAI 组成，能够提供关于场景的多个视角的信息。这些信息对于理解场景的深度和立体结构至关重要。

图 4. UrbanLF-Real

图 5. UrbanLF-Syn

以下将在 UrbanLF-Syn 上把原文与复现结果进行对比，如图6所示，复现结果与原文结果接近甚至更优。

方法	骨干网络	数据类型	参数大小	Acc	mAcc	mIoU
LF-IENet-Res50 (原文)	ResNet-50	LF	94.6M	90.42	86.17	78.27
LF-IENet-Res50 (复现)	ResNet-50	LF	94.6M	91.27	85.51	77.85
LF-IENet-HR48 (原文)	HRNetV2-W48	LF	117.4M	92.41	88.31	81.78
LF-IENet-HR48 (复现)	HRNetV2-W48	LF	117.4M	93.14	89.09	82.71

图 6. UrbanLF-Syn

经过改进后，如7所示，在 UrbanLF-Syn 上得到的分割结果相比复现结果有微弱的提升，使用 LF-IENet-HR48 的分割 mIoU 提升了 0.53%，其他指标也有不同程度的提升。

方法	骨干网络	数据类型	参数大小	Acc	mAcc	mIoU
LF-IENet-Res50 (复现)	ResNet-50	LF	94.6M	91.27	85.51	77.85
LF-IENet-Res50 (改进)	ResNet-50	LF	94.6M	**91.41**	**86.04**	**78.18**
LF-IENet-HR48 (复现)	HRNetV2-W48	LF	117.4M	93.14	89.09	82.71
LF-IENet-HR48 (改进)	HRNetV2-W48	LF	117.4M	**93.47**	**89.14**	**83.24**

图 7. UrbanLF-Syn

以下将在 UrbanLF-Real 上把原文与复现结果进行对比，如图8所示，复现结果与原文结果接近。

方法	骨干网络	数据类型	参数大小	Acc	mAcc	mIoU
LF-IENet-Res50 (原文)	ResNet-50	LF	94.6M	92.01	85.10	78.09
LF-IENet-Res50 (复现)	ResNet-50	LF	94.6M	92.04	83.55	76.58
LF-IENet-HR48 (原文)	HRNetV2-W48	LF	117.4M	92.09	86.03	79.19
LF-IENet-HR48 (复现)	HRNetV2-W48	LF	117.4M	92.31	85.00	78.06

图 8. UrbanLF-Syn

经过改进后，如7所示，在 UrbanLF-Real 上得到的分割结果相比复现结果三个指标出现了不同程度的下降，这可能是因为加入的注意力模块破坏了隐式分支和显示分支对分割结果的平衡作用，也有可能是因为在真实数据集上有更多的干扰因素。

方法	骨干网络	数据类型	参数大小	Acc	mAcc	mIoU
LF-IENet-Res50 (复现)	ResNet-50	LF	94.6M	**92.04**	**83.55**	**76.58**
LF-IENet-Res50 (改进)	ResNet-50	LF	94.6M	91.49	82.05	75.31
LF-IENet-HR48 (复现)	HRNetV2-W48	LF	117.4M	**92.31**	**85.00**	**78.06**
LF-IENet-HR48 (改进)	HRNetV2-W48	LF	117.4M	91.54	83.92	76.52

图 9. UrbanLF-Syn

6 总结

本文针对单一视图图像语义分割的局限性，引入了光场模型，从多视角挖掘四维空间中的语义信息，提出了光场语义分割网络 **LF-IENet**。该网络综合利用深度、角度和时间等多方面信息，通过隐式和显式两个分支处理光场数据。隐式分支通过集成多个参考视图的特征强化中心视图，显式分支通过视差估计将参考视图特征传播到中心视图，最终融合两个分支的输出特征。这种显隐互补的方法高效利用了光场信息，显著提升了语义分割性能。网络的总损失函数由主要交叉熵损失 L_{main} 和两个辅助损失 L_{aux_1}、L_{aux_2} 组成。

本文进一步改进了 **LF-IENet**，通过设置阈值梯度精细化调整交叉注意力权重，根据视差距离赋予不同权重，增强了模型对关键信息的保留能力和特定距离范围特征的处理能力。同时，引入自注意力模块强化参考视图特征，逐步提升数据表示的丰富性和模型对数据关系的理解能力。实验表明，优化后的 **LF-IENet** 在分割合成数据集时平均交并比（mIoU）上稳定提升，尤其在 **LF-IENet-HR48** 中表现最为显著。然而，改进方法仍存在局限性：阈值设置可能未完全拟合视差距离与交叉注意力权重的关系，且在真实数据集 UrbanLF-Real 上性能有所下降。

参考文献

[1] J Cao, H Leng, D Lischinski, et al. Shapeconv: Shape-aware convolutional layer for indoor rgb-d semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7088–7097, 2021.

[2] LZ Chen, Z Lin, Z Wang, et al. Spatial information guided convolution for real-time rgbd semantic segmentation. *IEEE Transactions on Image Processing*, 30:2313–2324, 2021.

[3] K Han, W Xiang, E Wang, et al. A novel occlusion-aware vote cost for light field depth estimation. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):8022–8035, 2021.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[5] T Leistner, H Schilling, R Mackowiak, et al. Learning to think outside the box: Wide-baseline light field depth estimation with epi-shift. In *2019 International Conference on 3D Vision (3DV)*, pages 249–257. IEEE, 2019.

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[7] J Peng, Z Xiong, Y Wang, et al. Zero-shot depth estimation from light field using a convolutional neural network. *IEEE Transactions on Computational Imaging*, 6:682–696, 2020.

[8] D Seichter, M Köhler, B Lewandowski, et al. Efficient rgb-d semantic segmentation for indoor scene analysis. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 13525–13531. IEEE, 2021.

[9] C Shin, HG Jeon, Y Yoon, et al. Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4748–4757, 2018.

[10] Yu-Ju Tsai, Yu-Lun Liu, Ming Ouhyoung, and Yung-Yu Chuang. Attention-based view selection networks for light-field disparity estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12095–12103, 2020.

[11] S Vandenhende, S Georgoulis, and L Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 527–543. Springer International Publishing, 2020.

[12] Yingqian Wang, Longguang Wang, Zhengyu Liang, Jungang Yang, Wei An, and Yulan Guo. Occlusion-aware cost constructor for light field depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19809–19818, 2022.

[13] Yingqian Wang, Longguang Wang, Gaochang Wu, Jungang Yang, Wei An, Jingyi Yu, and Yulan Guo. Disentangling light fields for super-resolution and disparity estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):425–443, 2022.

[14] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[15] J Zhang, Q Su, B Tang, et al. Dpsnet: Multitask learning using geometry reasoning for scene depth and semantics. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):2710–2721, 2021.

[16] S Zhang, H Sheng, C Li, et al. Robust depth estimation for light field via spinning parallelogram operator. *Computer Vision and Image Understanding*, 145:148–159, 2016.

[17] Shuo Zhang, Song Chang, and Youfang Lin. End-to-end light field spatial super-resolution network using multiple epipolar geometry. *IEEE Transactions on Image Processing*, 30:5956–5968, 2021.

[18] H Zhao, Y Zhang, S Liu, et al. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European conference on computer vision (ECCV)*, pages 267–283, 2018.

[19] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[20] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[21] W Zhou, E Zhou, G Liu, et al. Unsupervised monocular depth estimation from light field image. *IEEE Transactions on Image Processing*, 29:1606–1617, 2019.

[22] J Zhuang, Z Wang, and B Wang. Video semantic segmentation with distortion-aware feature correction. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(8):3128–3139, 2020.