在混合密集连接的Transformer中使用分块单元对比用于多模态肿瘤分割

摘要

大多数深度学习方法都受到表征能力不足、模态数量特定和计算复杂度高等限制。在原文中，作者提出了一种用于肿瘤分割的混合密集连接网络，名为H-DenseFormer，它结合了卷积神经网络（CNN）和Transformer结构的表示能力。具体来说，H-DenseFormer集成了基于Transformer的多路径并行嵌入（MPE）模块，该模块可以采用任意数量的模态作为输入，从不同模态中提取融合特征。此外，作者设计了一个轻量级的密集连接Transformer（DCT）块来代替标准的Transformer块，从而显着降低了计算复杂度。本报告在此基础上，引入了分块单元对比损失来解决Transformer输出特征值相似度高的问题。本报告对两个公共多模态数据集HECK TOR21和PI-CAI22进行了复现实验，并在PI-CAI22的分割任务中引入了分块单元对比损失。实验结果表明，改进的方法与复现的方法相当甚至更好，同时具有较快的收敛速度。

关键词：肿瘤分割；多模态医学图像；Transformer；深度学习；分块对比

1 引言

在计算机视觉领域不断有人尝试将transformer引入，近期也出现了一些效果不错的尝试，典型的如目标检测领域的detr和可变形detr，分类领域的vision transformer等等，在大规模数据集中取得了显著成功。与自然图像相比，多模态医学图像具有明确且重要的远程依赖性。肿瘤分割（tumor segmentation）是医学图像分析中一个具有挑战性的问题，医学图像分割的目的是使图像中解剖或病理结构的变化更加清晰;它在计算机辅助诊断和智能医疗中发挥着至关重要的作用，极大地提高了诊断的效率和准确性。目前流行的医学图像分割任务包括肝脏和肝脏肿瘤分割，脑和脑肿瘤分割，视盘分割，细胞分割，肺和肺结节分割等，其中肿瘤分割的目标是使用正确定位的masks生成肿瘤区域的准确轮廓。

MICCAI是医学图像分析领域的顶会，为了结合多模态数据分析的研究方向和面向国际前沿，本报告在此基础上检索MICCAI 2023接收论文中有关多模态的文章，通过比较github的star数和数据集链接公布情况选择复现对象。

为了帮助临床医生做出准确的诊断，有必要对医学图像中的一些关键目标进行分割，并从分割区域中提取特征。由于特征表示的困难，图像分割仍然是计算机视觉领域中最具挑战性的课题之一。特别是医学图像的特征提取比普通RGB图像更难，因为前者往往存在模糊、带噪声、对比度低等问题。由于深度学习技术的快速发展，医学图像分割将不再需要

手工制作的特征。本报告复现的工作旨在解决多模态医学图像的分割问题，其网络中堆叠了transformer模块，而这样的堆叠会使得输出值更加平均。为此，本报告的改进是引入patch token对比损失，驱使网络得到更具判别性的特征，改善网络表现。

2 相关工作

2.1 肿瘤分割

研究人员从不同角度进行肿瘤分割，如Wang等人 [1]提出一种新型肿瘤敏感的合成模块，并展示其与肿瘤分割集成后的用途； Zhang等人 [2]提出一种具有可变形特征融合和不确定区域细化的新型多模态肿瘤分割方法；为了解决不合理地融合多模态图像的问题，文献 [3]利用放射科医生如何通过多种MRI模态诊断脑肿瘤的临床知识，提出临床知识驱动的脑肿瘤分割模型。

2.2 多模态医学图像

在对医学图像的处理上也有不同的方法，如Zhang等人 [4]利用模态间关系，提出多模态对比域共享生成对抗网络，以实现有效的多模态对比自监督医学图像分割；文献 [5]使用了文本信息，提出一种新的文本增强医学图像分割模型LViT（Language meet Vision Transformer）；Zhang等人 [6]从结构角度出发，提出一个与模型无关的框架，以通过单一网络检测各种器官和模态的异常。

3 本文方法

3.1 本文方法概述

大多数现有方法要么由于不对称连接的设计而仅限于特定的模态数，要么由于模型参数量庞大而面临巨大的计算复杂性。因此，如何在保证计算效率的同时提高模型能力是本文的主要关注点。

为此，本文提出了一种高效的多模态肿瘤分割解决方案，称为混合密集连接网络（H-DenseFormer） [7]。首先，方法利用Transformer来增强不同模式的全局上下文信息。其次，H-DenseFormer集成了基于Transformer的多路径并行嵌入（MPE）模块，该模块可以提取和融合多模态图像特征，作为朴素输入级融合结构的补充。具体来说， MPE为每个模态分配独立的编码路径，然后合并所有路径的语义特征并将其馈送到分割网络的编码器。这解耦了不同模态的特征表示，同时放宽了对特定模态数量的输入约束。最后，本文设计了一个轻量级的密集连接Transformer（DCT）模块来替代标准Transformer，以确保性能和计算效率。对两个公开数据集的广泛实验结果证明了提出方法的有效性。

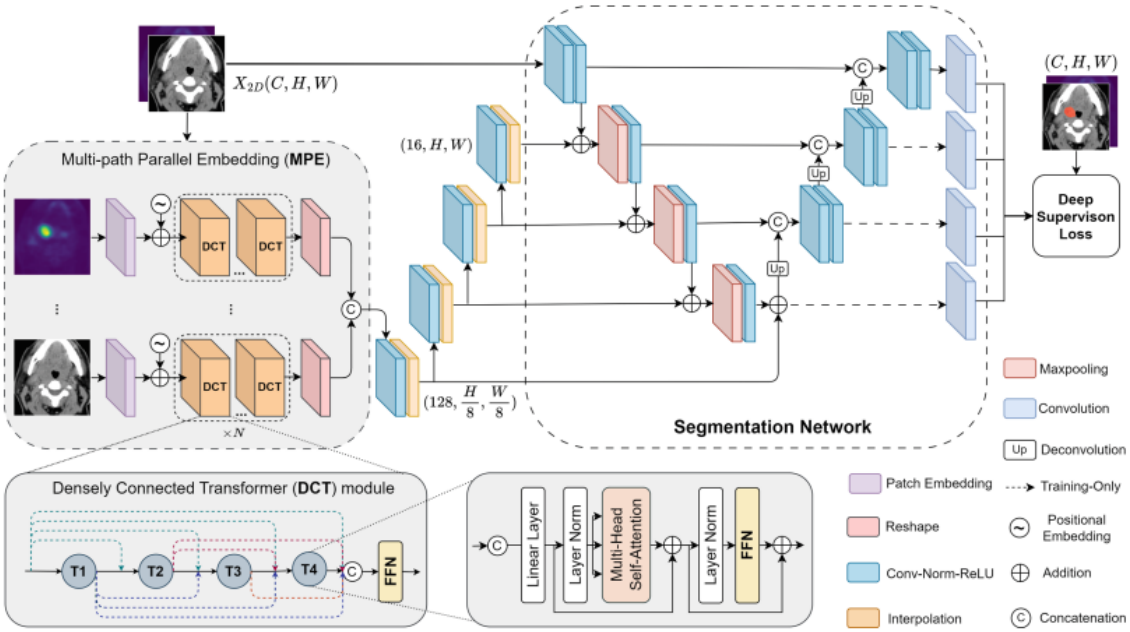

图 1. 提出的H-DenseFormer的整体架构

3.2 多路径并行嵌入

本文设计了多路径并行嵌入（MPE）模块来增强网络的表示能力，如图1所示，每个模态都有一个独立的编码路径，由patch嵌入模块、堆叠的密集连接Transformer（DCT）模块和reshape操作组成，不同路径的独立性允许MPE处理任意数量的输入模态。此外，Transformer的引入提供了对全局上下文信息进行建模的能力。

3.3 密集连接的Transformer

标准Transformer结构 [8]通常由密集的线性层组成，其计算复杂度与特征维度成正比。因此，集成Transformer可能会导致大量额外的计算和内存需求。缩短特征长度可以有效减少计算量，但同时也削弱了表示能力。为了解决这个问题，本文提出了密集连接Transformer（DCT）模块，以平衡计算成本和表示能力。图1详细介绍了DCT模块，该模块由四个Transformer层和一个前馈层组成。不同的Transformer层紧密连接，以保持较低特征维度的表征能力，最后的前馈层生成不同层的融合特征。

3.4 分割骨架网络

H-DenseFormer采用U形编码器-解码器结构作为其骨干，如图1所示，编码器提取特征并逐渐降低其分辨率。为了保留更多细节，本文将最大下采样因子设置为8。MPE的多级多模态特征以按位加法的方式融合以丰富语义信息，解码器用于恢复特征的分辨率，由跳跃连接到编码器的反卷积层和卷积层组成。特别是，本文采用深度监督（DS）损失来提高收敛性，这意味着解码器的多尺度输出参与最终的损失计算。

深度监督损失 在训练中解码器有4个输出，为了缓解像素不平衡问题，作者使用Focal loss [9]和Dice loss的组合损失作为优化目标。

4 复现细节

4.1 与已有开源代码对比

```
output,fmap,cam_aux = net(data)
loss = criterion(output, target)
pseudo_label_aux = cam_to_label(cam_aux.detach(), ignore_mid=True,
bkg_thre=0.5, high_thre=0.7,low_thre=0.25, ignore_index=255)
aff_mask = label_to_aff_mask(pseudo_label_aux)
ptc_loss = get_masked_ptc_loss(fmap, aff_mask)
loss=loss+ptc_loss
```

扩展了网络的输出，然后生成了aff_mask，计算ptc损失并与原loss相加。

```
i=0
for block, in self.blocks:
        x = block(x)
        if i==self.depth/2:aux=x
                i+=1
x = self.re_patch_embedding(x)
aux = self.re_patch_embedding(aux)
return F.interpolate(x,self.outsize),
F.interpolate(aux,self.outsize)
```

额外获取了网络的中间输出，经处理后返回。

```
inputs_cat = torch.cat([x, x.flip(-1)], dim=0)
att,aux=[],[]
for i in range(self.in_channels):
        att.append(self.attns[i](x[:,i:i+1,:,:])[0])
        aux.append(self.attns[i](inputs_cat[:,i:i+1,:,:])[1])
attnall = torch.cat(att,1)
aux = torch.cat(aux,1)
cam_aux = F.conv2d(aux, self.aux_classifier.weight).detach()

b, c, h, w = x.shape
_cam_aux = torch.max(cam_aux[:b,...], cam_aux[b:,...].flip(-1))

cam_aux_list = [F.relu(_cam_aux)]
cam_aux = torch.sum(torch.stack(cam_aux_list, dim=0), dim=0)
cam_aux = cam_aux + F.adaptive_max_pool2d(-cam_aux, (1, 1))
cam_aux /= F.adaptive_max_pool2d(cam_aux, (1, 1)) + 1e-5
```

对返回的中间输出进一步处理，经过卷积和CAM生成操作，得到cam_aux。

```python
def label_to_aff_mask(cam_label, ignore_index=255):
    b,h,w = cam_label.shape

    _cam_label = cam_label.reshape(b, 1, -1)
    _cam_label_rep=_cam_label.repeat([1,_cam_label.shape[-1],1])
    _cam_label_rep_t = _cam_label_rep.permute(0,2,1)
    aff_label=(_cam_label_rep==_cam_label_rep_t).type(torch.long)

    for i in range(b):
        aff_label[i,:,_cam_label_rep[i,0,:]==ignore_index]=
        ignore_index
        aff_label[i,_cam_label_rep[i,0,:]==ignore_index, :]=
        ignore_index
    aff_label[:, range(h*w), range(h*w)] = ignore_index
    return aff_label

def cam_to_label(valid_cam, img_box=None, bkg_thre=None,
high_thre=None,low_thre=None, ignore_mid=False, ignore_index=None):
    cam_value, _pseudo_label = valid_cam.max(dim=1, keepdim=False)
    _pseudo_label += 1
    _pseudo_label[cam_value<=bkg_thre] = 0

    if img_box is None:
        return _pseudo_label

    if ignore_mid:
        _pseudo_label[cam_value<=high_thre] = ignore_index
        _pseudo_label[cam_value<=low_thre] = 0
    pseudo_label = torch.ones_like(_pseudo_label) * ignore_index

    for idx, coord in enumerate(img_box):
        pseudo_label[idx, coord[0]:coord[1], coord[2]:coord[3]] =
        _pseudo_label[idx, coord[0]:coord[1], coord[2]:coord[3]]

    return valid_cam, pseudo_label

def get_masked_ptc_loss(inputs, mask):
    b, c, h, w = inputs.shape

    inputs = inputs.reshape(b, c, h*w)
```

```
    def cos_sim(x):
        x = F.normalize(x, p=2, dim=1, eps=1e-8)
        cos_sim = torch.matmul(x.transpose(1,2), x)
        return torch.abs(cos_sim)

    inputs_cos = cos_sim(inputs)

    pos_mask = mask == 1
    neg_mask = mask == 0
    loss=0.5*(1-torch.sum(pos_mask*inputs_cos)/(pos_mask.sum()+1))
    + 0.5 * torch.sum(neg_mask * inputs_cos) / (neg_mask.sum()+1)
    return loss
```

添加了将标签转为亲和mask的函数label_to_aff_mask，CAM转换为标签的函数cam_to_label，实现了ptc损失的具体步骤并返回损失值。

4.2 实验环境搭建

Linux系统和单张Nvidia 4090显卡，使用Pytorch框架，需要的python包见requirements.txt。

4.3 界面分析与使用说明

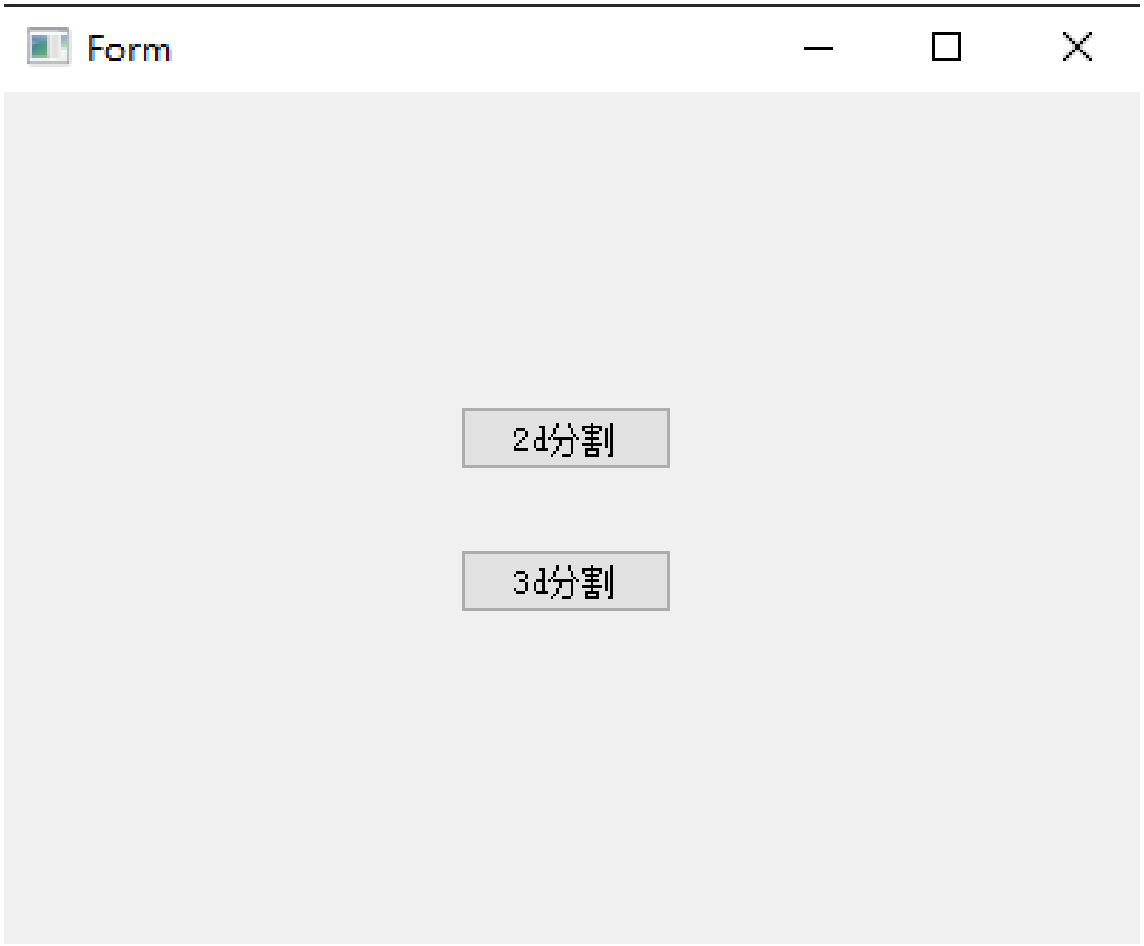

图 2. 操作界面示意

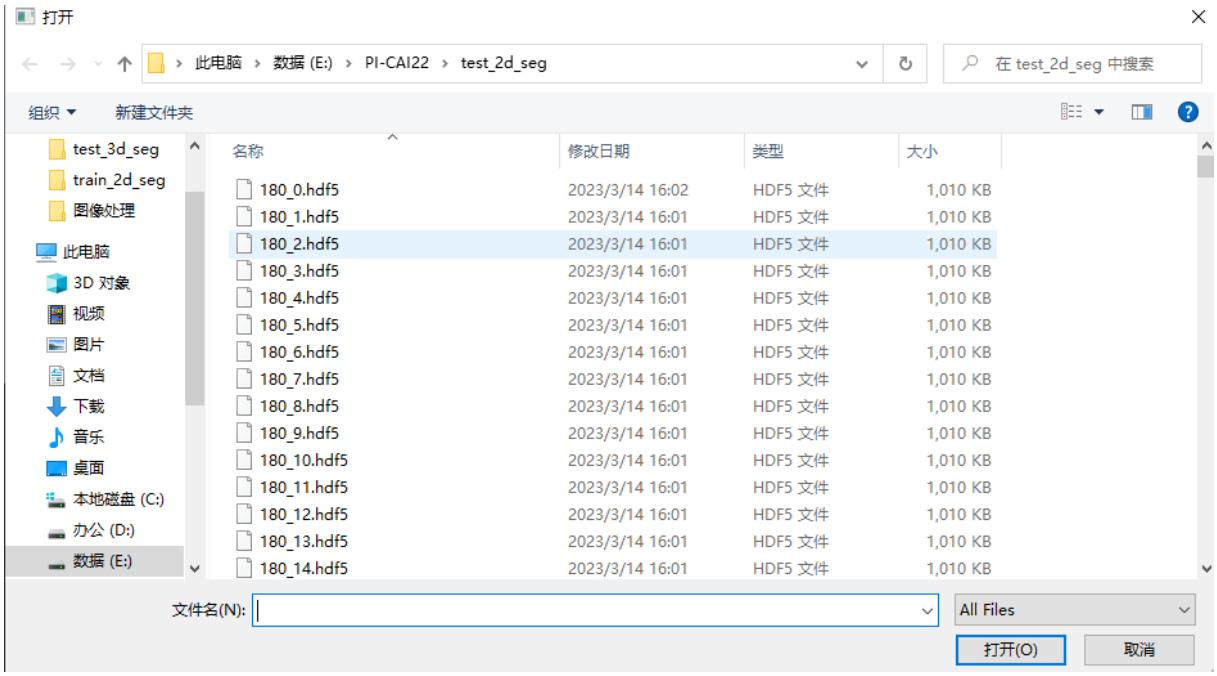

图 3. 选择文件界面示意

4.4 改进点

一、针对transformer网络中出现的过平滑问题，引入PTC(Patch Token Contrast)模块 [10]，缓解transformer输出值相似，趋于均匀分布的问题，提升模型性能和收敛速度。由于transfomer中间层仍然可以保留patch tokens的语义多样性，因此改进利用了中间层的知识，即可靠的成对token关系，来监督最终的patch tokens。报告添加了PTC损失来监督训练，公式化为

$$L_{ptc} = \frac{1}{N^+} \sum_{Y_i=Y_j} (1 - \text{sim}(F_i, F_j)) + \frac{1}{N^-} \sum_{Y_i \neq Y_j} \text{sim}(F_i, F_j) \tag{1}$$

其中，N^+/N^-分别是正负对的数量，$\text{sim}(.,.)$用来计算相似性，F是最终层的patch tokens。

表 1. 网络模块参数量

模块名	参数量
deep_conv.double_conv.0	884736
block_4_2_left.conv	589824
up1.double_conv.0	294912
attns.0.patch_embeddings	262144

二、此外，为了进一步降低计算复杂性，报告还分析并列出了框架中参数量前4大的模块，如表1所示。令输入特征图的尺寸为h×w，卷积层的输入和输出通道数分别为c_{in}和c_{out}，方形卷积核的大小为k，则卷积的计算量可表示为$\text{hwk}^2 c_{in} c_{out}$。这四个模块的核心操作都为卷积，这里可将其卷积核大小从3降为1。

5 实验结果分析

定量结果 在表2中可以看到，在HECKTOR21数据集上，复现结果与原文的差距在可接受的范围内；在PI-CAI22数据集上，复现结果与原文的差距较大，可能是因为评估方法不同，事实上训练过程中的表现是与原文相似的。由于运行时间开销，改进的方法暂时没进行交叉

表 2. 在测试集上与原文结果的比较

方法	DSC(%) ↑	HD95(mm) ↓	JI(%) ↑	epoch
HECKTOR21, 两种模态 (CT and PET)				
原文(3D)	73.9 ± 0.5	8.1 ± 0.6	62.5 ± 0.5	—
复现	69.3 ± 0.02	**3.4 ± 3.1**	58.6 ± 0.02	80
PI-CAI22, 三种模态 (T2W, DWI and ADC)				
原文(2D)	49.9 ± 1.2	35.9 ± 8.2	37.1 ± 1.2	—
复现	40.3 ± 0.03	42.1 ± 7.8	32.4 ± 0.03	61
改进	**41.4**	**40.4**	**33.5**	**31**

验证。以Dice和JI为例，改进的方法都有1.1个点的提升。同时收敛速度也大大提升了，使用的epoch数大致为复现方法的一半，这是因为引入了浅层特征的信息，在之前的论述中也提到了多尺度输出有助于提高收敛性。

定性结果

图 4. 3D图像分割结果示意

图4是对3D图像的某一通道进行可视化。

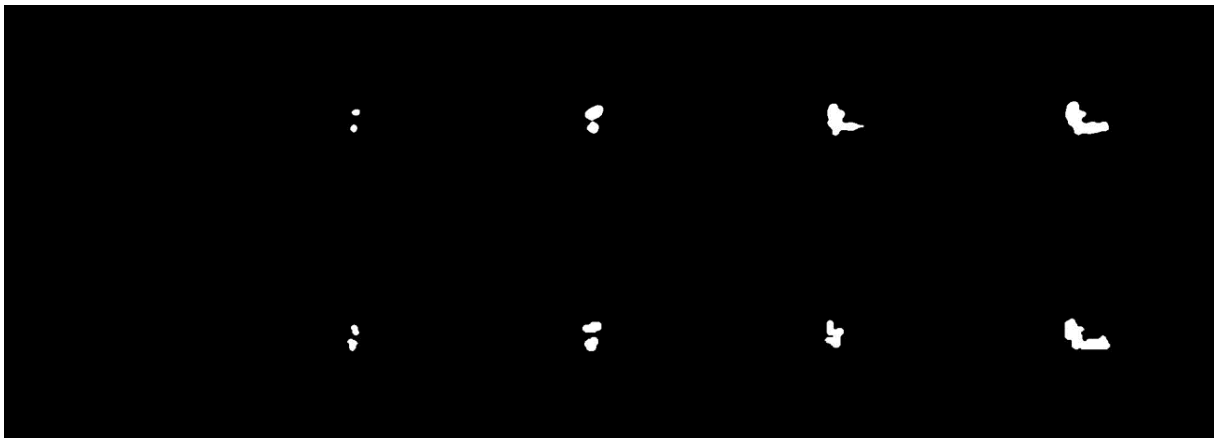

图 5. 2D图像分割结果示意

图5最左边一列表示当输入图的值是全0时，输出也为全0，表示模型能很好地处理这种情况，抑制了噪声的引入。

6 总结与展望

本报告复现了H-DenseFormer，针对transformer模块不断堆叠出现的过平滑问题，用中间层的输出来监督最终层的输出，提升了效果并有更快的收敛速度。不足之处是没有进一步分析transformer的层数对patch token对比的影响，改进的方向可以是完善界面，将图片直接在界面上显示；还可以探索3D transformer的类激活图生成。

参考文献

[1] Shuai Wang, Kun Sun, Li Wang, Liangqiong Qu, Fuhua Yan, Qian Wang, and Dinggang Shen. Breast tumor segmentation in dce-mri with tumor sensitive synthesis. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):4990–5001, 2023.

[2] Yue Zhang, Chengtao Peng, Ruofeng Tong, Lanfen Lin, Yen-Wei Chen, Qingqing Chen, Hongjie Hu, and S. Kevin Zhou. Multi-modal tumor segmentation with deformable aggregation and uncertain region inpainting. *IEEE Transactions on Medical Imaging*, 42(10):3091–3103, 2023.

[3] Jianwei Lin, Jiatai Lin, Cheng Lu, Hao Chen, Huan Lin, Bingchao Zhao, Zhenwei Shi, Bingjiang Qiu, Xipeng Pan, Zeyan Xu, Biao Huang, Changhong Liang, Guoqiang Han, Zaiyi Liu, and Chu Han. Ckd-transbts: Clinical knowledge-driven hybrid transformer with modality-correlated cross-attention for brain tumor segmentation. *IEEE Transactions on Medical Imaging*, 42(8):2451–2461, 2023.

[4] Jiaojiao Zhang, Shuo Zhang, Xiaoqian Shen, Thomas Lukasiewicz, and Zhenghua Xu. Multi-condos: Multimodal contrastive domain sharing generative adversarial networks for self-supervised medical image segmentation. *IEEE Transactions on Medical Imaging*, pages 1–1, 2023.

[5] Zihan Li, Yunxiang Li, Qingde Li, Puyang Wang, Dazhou Guo, Le Lu, Dakai Jin, You Zhang, and Qingqi Hong. Lvit: Language meets vision transformer in medical image segmentation. *IEEE Transactions on Medical Imaging*, pages 1–1, 2023.

[6] Yinghao Zhang, Donghuan Lu, Munan Ning, Liansheng Wang, Dong Wei, and Yefeng Zheng. A model-agnostic framework for universal anomaly detection of multi-organ and multi-modal images. In *Medical Image Computing and Computer Assisted Intervention*, pages 232–241, 2023.

[7] Jun Shi, Hongyu Kan, Shulan Ruan, Ziqi Zhu, Minfan Zhao, Liang Qiao, Zhaohui Wang, Hong An, and Xudong Xue. H-denseformer: An efficient hybrid densely connected transformer for multimodal tumor segmentation. In *Medical Image Computing and Computer Assisted Intervention*, pages 692–702, 2023.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16×16 words: transformers for image recognition at scale. In *Proc. Int. Conf. on Learning Representations*, 2021.

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. Int. Conf. on Computer Vision*, pages 2999–3007, 2017.

[10] Lixiang Ru, Heliang Zheng, Yibing Zhan, and Bo Du. Token contrast for weakly-supervised semantic segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3093–3102, 2023.