文章编号：2095-2163(2022)11-0078-09 中图分类号：TP391 文献标志码：A

# 基于一阶段目标检测网络头部算法研究

肖贵明，丁德锐，梁 伟，魏国亮
（上海理工大学 光电信息与计算机工程学院，上海 200093）

**摘 要：**目标检测的网络框架对目标检测结果影响极大，其中网络头部的研究是网络框架改进的重点之一。本文针对一阶段目标检测的网络头部进行改进。通过对当前两阶段网络头部的研究与一阶段网络框架 RetinaNet 头部热力图的输出进行分析，在一阶段网络头部创新性地引入池化层模块、提出双分类头模块、使用 2 个网络头部权重自适应分配结合的方法。本文使用 RetinaNet 作为 baseline、VOC0712 和 MS COCO2017 数据集作为实验数据集，最终在 VOC0712 上 *mAP* 达到了 80.8%，相比于 baseline 提高了 3.5%，在 MS COCO2017 测试集上 *mAP* 达到了 40.2%，相比于 RetinaNet 提高了 1.1%，使用多尺度后 *mAP* 达到了 41.7%，提高了 2.4%。

**关键词：**目标检测；baseline；VOC0712；MS COCO2017；RetinaNet；双分类头；热力图；*mAP*

## Research on head algorithm of network based on one-stage object detection

XIAO Guiming, DING Derui, LIANG Wei, WEI Guoliang
(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

**【Abstract】** The network framework of object detection has a great influence on the object detection results, and the research of the network head is one of the focuses of the improvement of the network framework. This paper improves the network header of one-stage object detection. By analyzing the current two-stage network head research and the output of the one-stage network framework RetinaNet head heatmap, the paper innovatively introduces the pooling layer module in the first-stage network head, proposes a dual-classification head module, and uses two networks. After that, a combined method is used for adaptive allocation of head weights. This paper uses RetinaNet as the baseline, VOC0712 and MS COCO2017 datasets as experimental datasets, thereafter *mAP* achieves 80.8% on VOC0712, which is 3.5% higher than baseline. Meanwhile *mAP* reaches 40.2% on MS COCO2017 test set, and compared with RetinaNet, it is improved by 1.1%. Furtherly after using multi-scale, *mAP* reaches 41.7%, which is an increase of 2.4%.

**【Key words】** object detection; baseline; VOC0712 ; MS COCO2017; RetinaNet; double classification head; heat map; *mAP*

## 0 引 言

计算机视觉中，目标检测是最基础的任务之一。近年来，目标检测发展迅速。形成两阶段目标检测和一阶段目标检测两种形式。两阶段目标检测中，RCNN[1]通过选择搜索生成区域，并利用深度网络提取特征。SPPNet[2]提出空间池化层对 RCNN 进行加速；Fast-RCNN[3]提出 RoI pooling 提升性能和速度；Faster-RCNN[4]提出 RPN 生成 RoI；R-FCN[5]提出 position sensitive RoI pooling 进行处理位置变动问题；FPN[6]建立从下到上、从上到下、横向连接的模块形式；Mask-RCNN[7]提出 RoIAlign。一阶段目标检测中：SSD[8]和 YOLO[9-11]直接预测对象类别和位置；Focal loss[12]针对一阶段类别不平衡问题，并提出 RetinaNet；Fcos 等[13-16]使用 anchor-free 的方式进行目标检测。

现下，目标检测网络框架正陆续推出，学界对两阶段头部网络的研究是其重点之一。Cascade RCNN[17]根据不同的 *IoU*，提出级联的检测头；Mask RCNN 增加额外的语义分割网络头部，提升性能；IoU-Net[18]增加头部分支来预测 *IoU*；Double-Head[19]研究发现 fc 层更有利于分类，使用双头网络对物体进行检测。近年来，一些两阶段网络框架的网络头部变动如图 1 所示。图 1(a)使用全连接头进行分类和定位；图 1(b)使用全卷积头进行分类和定位；图 1(c)和图 1(d)是 Double-Head 提出的

**基金项目：**国家自然科学基金面上项目(61973219)；上海市“科技创新行动计划”国内科技合作项目(20015801100)。

**作者简介：**肖贵明(1995-)，男，硕士研究生，主要研究方向：视觉目标检测；丁德锐(1981-)，男，博士，教授，博士生导师，主要研究方向：随机非线性控制与滤波、智能优化算法、图像处理；梁 伟(1996-)，男，博士研究生，主要研究方向：生成对抗网络、视觉跟踪；魏国亮(1973-)，男，博士，教授，主要研究方向：随机控制及电机控制等方面的教学与科研工作。

**通讯作者：**丁德锐 Email：deruiding2010@usst.edu.cn

**收稿日期：**2022-03-11

使用卷积和全连接相互作用的双头网络。

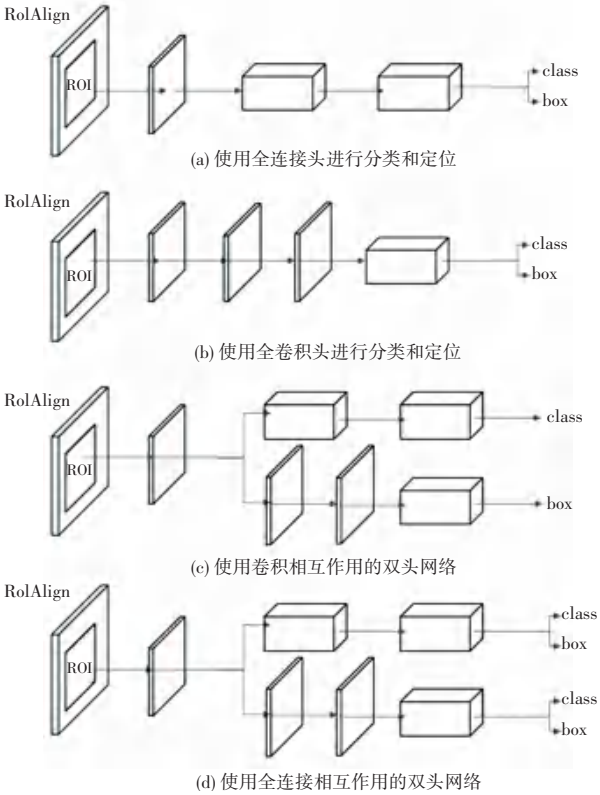


**图 1　两阶段目标检测头部变化图**

**Fig. 1　Two-stage object detection head change figure**

根据两阶段网络 Double-Head 的研究，*fc* 层更加适合分类，主要由于分类具有空间敏感性。一阶段网络没有两阶段网络的 *RoIAlign* 模块，前向网络传入的特征到网络头部时无法减少大量参数，故一阶段网络分类头部使用 *fc* 时会产生大量的参数，极大地减少一阶段网络头部改进的可能性。

再者，本文将一阶段网络 RetinaNet 的网络分类头的热力图进行输出，如图 2~图 4 所示。图 2~图 4 中，左图均为图像刚进入网络分类头部的热力图输出，右图均为经过网络头部后的热力图输出。可以看出，经过网络头部后，使得关注点能较好地集中在其中的一部分。但也出现以下 2 个问题。其一，经过网络头部后，由于图中识别的物体范围太大、含有较多的背景，关注点无法集中在识别的物体上；其二，头部网络提取的特征反映在原图中，有些物体识别不到或对物体识别的信息较少。

**图 2　网络头部热力图**

**Fig. 2　Network head heat map**

**图 3　网络头部热力图**

**Fig. 3　Network head heat map**

**图 4　网络头部热力图**

**Fig. 4　Network head heat map**

综上所述，本文针对一阶段网络 RetinaNet 的分类头部进行改进，提出双分类头，主要由 2 个分类头部组成：循环通道注意力模块头（Circulatory Channel Attention Module）；多头注意力机制模块头（Multi-head attention mechanism module head）。其中，循环通道注意力模块头用于着重增强当前提取特征，在进行分类时能够对物体进行准确识别，在关注物体本身的同时减少对背景的关注。多头注意力机制模块头具有 *fc* 的特性，同时又具有自注意力的性能，在获得更大空间性的同时，加强对物体本身的关注与识别物体本身。本文有以下 3 个创新点：

（1）一阶段网络头部引入池化层模块，以此减少参数量。

（2）使用双分类头，提出循环通道注意力模块，将 NLP 中的多头注意力机制模块有改进性地应用到本文中。

（3）2 个网络头部进行自适应权重分配，进行输出。

## 1　基于 RetinaNet 的双分类头

### 1.1　分析网络框架

本文采用的目标检测网络框架是在 RetinaNet 基础上加以改进的，如图 5 所示。整体来说，由主干网络、特征金字塔（FPN）和网络头中的分类、定位两个任务分支、三大模块所组成。本次选取 ResNet101 为主干网络。

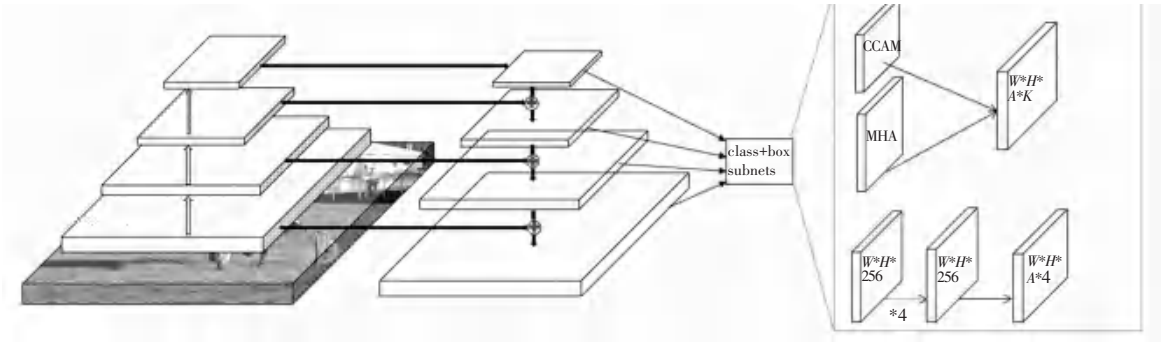


**图 5 网络结构图**

**Fig. 5 Network structure**

特征金字塔是从主干网络 $P_3$ 到 $P_6$ 构造得到的。研究中，$l$ 表示特征金字塔的层数，第 $P_l$ 层大小为该层输入图像分辨率 $\frac{1}{2^l}$ 倍，图 5 中只简单地显示了其中的 3 层。特征金字塔的每一层都是用来检测大小不同的物体，从 $P_3$ 到 $P_6$ 的金字塔中参考框的面积大小为 $32^2$ 到 $256^2$，其中第 $l$ 层参考框大小的面积为第 $l$ 层采样率 $2^l$ 乘 4 的平方大小。这里将长宽比为{2∶1，1∶1，1∶2}三种比率使用在每一层上，同时在同一层增加了 $\{2^0, 2^{\frac{1}{3}}, 2^{\frac{2}{3}}\}$ 三种参考框的面积大小，故每层一共 $A = 9$ 种参考框。

网络头的分类和定位两个分支是全卷积分支。定位分支是无关类别的，通过 4 个 $W * H * 256$ 卷积层，具有 $W * H$ 的空间位置数，用来回归 $A$ 个边界框的偏移量，其中每个框有 4 个偏移量，分别表示左下、右上两个点相对偏移量、或者中心点和长宽的相对偏移量，定位分支最后输出大小为 $W * H * A * 4$。分类分支用来预测每张图片的每个特定空间位置的类别，由图 5 可知，本文在分类分支上使用 2 个分类头并将 2 部分进行自适应权重加权，得到的空间位置数为 $W * H$，每个位置有 $A$ 个边界框，$K$ 个种类，预测该位置属于某种类别的可能性，故分类分支大小为 $W * H * A * K$。

### 1.2 说明循环通道注意力模块头

图 6 为通道注意力模块。通过将输入的特征图进行平均池化和最大池化，即使得输入特征变成 $B * 1 * 1 * C$，并通过 *Conv* 进行通道数的降维，减小参数量，在此基础上将平均池化和最大池化得到的特征进行相加，再归一化得到最终的结果。通过通道注意力模块可以对得到的特征进行区分，使其重点关注某些特征，减小关注的特征范围。

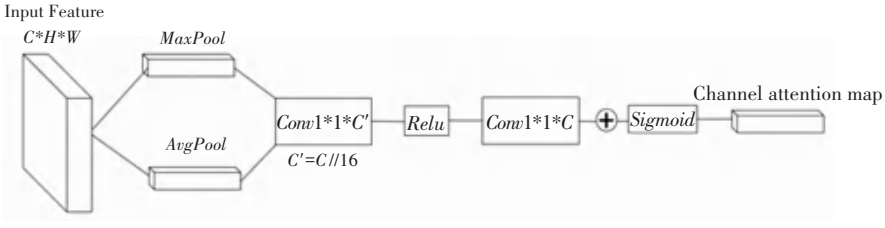


**图 6 通道注意力模块图**

**Fig. 6 Channel attention module figure**

图 7 为循环通道注意力模块，选择通道注意力模块是由于该部分在网络头部，语义特征足够强，只需要对提取得到特征的物体部分进行重点关注。循环通道注意力模块中进行了以下 3 点操作：

(1)引入恒等变换。图 7 中为输出特征和通道注意力所得到的结果相乘后和输出特征进行相加，恒等变换的使用是为了保持当前性能的同时，尽可能地增强性能，同时保持网络的稳定性。

(2)通道注意力模块使用的是未经过 *Conv* 变换的输入特征，由于输入特征相比于输出特征保留有更多的细节，所以将通道注意力模块使用在输入特征上，可以对每个通道做更细致的分析，得到的结果更加地准确。

(3)使用结构循环，为了通过多次操作对重点关注到的特征进一步加强，对不应关注的特征进一步弱化，使得提取得到的特征更加地准确。

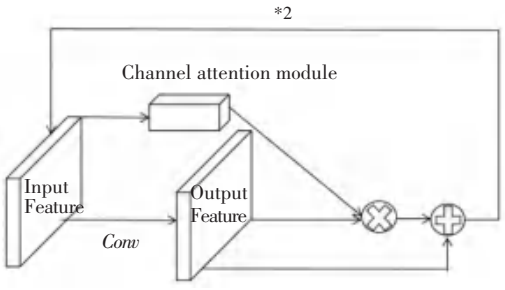


**图 7　循环通道注意力模块图**

**Fig. 7　Circulation channel attention module figure**

### 1.3　阐述多头注意力机制模块头

多头注意力机制如图 8 所示。图 8 中为多头注意力模块头部为 1 时的情况(self-attention)。多头注意力是在 NLP 的 Transformer 中提出的,旨在解决 RNN 中数据不能并行所带来的运行速度较慢的问题。这里，$x_i$ 表示输入数据，通过嵌入层进行 embedding得到 $a_i$。由此推得的数学公式可写为：

$$a_i = Wx_i$$

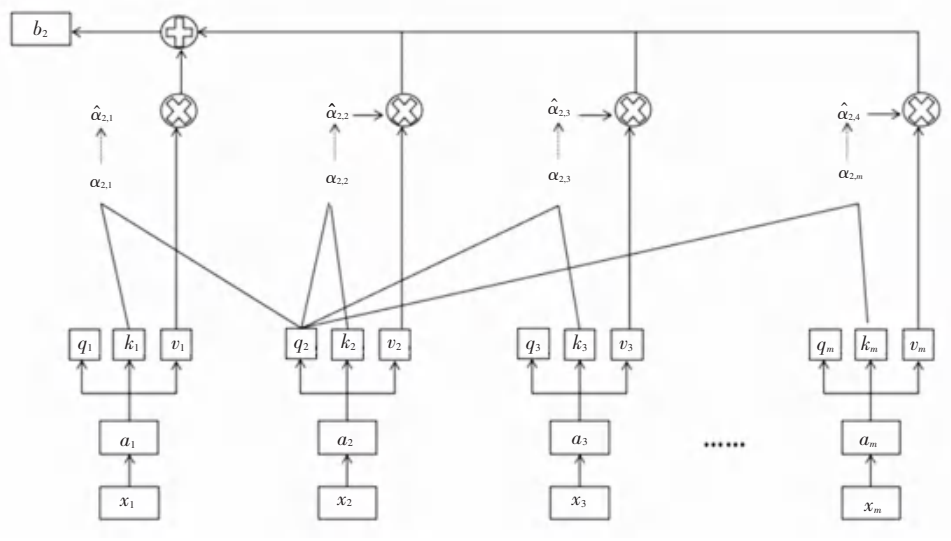


**图 8　多头注意力机制**

**Fig. 8　Multi-head attention mechanism**

对输入 $a_i$ 进行变换。相应数学公式分别如下：

$$q_i = W^Q a_i \tag{1}$$

$$k_i = W^k a_i \tag{2}$$

$$v_i = W^V a_i \tag{3}$$

得到 $q_i$、$k_i$、$v_i$。将得到的 3 个输入进行向量点积,即：

$$a_{j.i} = \frac{q_j^{Transpose} \cdot k_i}{\sqrt{d_{q,k}}} \tag{4}$$

计算得到 $a_{j,i}$ 后,对其进行归一化,表示为：

$$\hat{a}_{j,i} = Softmax(a_{j,i}) \tag{5}$$

进一步得到 $\hat{a}_{j,i}$，最后经过求和,具体公式为：

$$b_j = \sum_{i=1}^{T} \hat{a}_{j,i} \cdot \nu_i \tag{6}$$

如上计算后得到输出 $b_j$。这种自注意力机制具有很好的空间性,同时对自身的重要部分、即该识别的物体进行了重点关注,能够很好地满足局部特征和全局特征的提取和使用。

图 9 为多头注意力机制模块头。多头注意力机制模块头总共由 3 部分组成,分述如下：

(1)平均池化。由于一阶段目标检测网络不具有两阶段目标检测网络的 *RoIAlign* 模块，将传入到网络头部的特征进行再提取,减少大量传入头部的特征,故无法直接使用 *fc* 层。研究中可将传进网络头部的特征进行平均池化，从原先的 $W*H$ 缩小到 $N*N$,减小参数量。同时采用平均池化是由于到网络头部时,语义信息强,这时候使用平均池化能带来更好的性能。

(2)使用多头注意力机制,利用其本身具有良好的空间性和自注意力机制,能够更好地识别到物体。其中,多头注意力机制中的输入 *query*、*key*、*value* 分别是输入特征,大小为 $B*W*H*C$,进行池化后的特征大小为 $B*N*N*C$ 和 $D*N*N*C$。输入特征使用了恒等变换,保证性能的同时,加快收敛速度和网络稳定性。由于分类头部对位置信息不敏感,所以在使用多头注意力时并未使用位置信息。

(3)Layer Norm，通过多头注意力输出的特征为 $B*M*C$,$M$ 为 $H*W$,将 $C$ 个通道作为一个样本,对 $M$ 个位置的 $C$ 个通道进行归一化,使得每一个位置不同的通道具有和为 1 的权重,进一步加快训练速度。

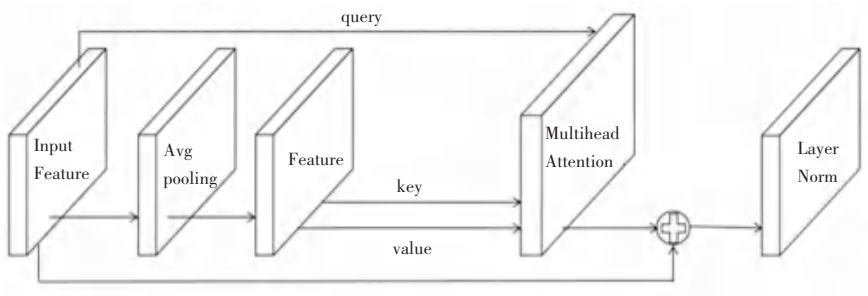


**图 9 多头注意力机制模块**

**Fig. 9 Multi-head attention mechanism module**

### 1.4 实现头部结合

本文中 2 个网络头部的偏重点有所不同,循环通道注意力模块头更加偏向能准确识别物体,减少背景混入;多头注意力机制模块头偏向于通过更大的空间性和自注意力将物体识别出来。所以将 2 个网络分类头部进行自适应加权。通过对 2 个头部进行平均池化生成 $B*1*1*C$ 的特征,进行合并,通过 *softmax* 进行归一化,将得到的权重和各自的网络头部特征进行相乘,相加得到最终的分类输出,让网络自身来决定该偏向于哪一部分,从而使性能更好。

## 2 实验结果与分析

### 2.1 实验数据集和评价指标

PASCAL VOC 数据集,包含 20 个类别,加上背景、共有 21 个类别。其中,VOC2007 共包含了 9 963 张图片:5 011 张为训练图片,4 952 张为测试图片。VOC2012 共包含了 11 540 张训练图片。本次实验联合 VOC2007 和 VOC2012 训练集进行训练,在 VOC2007 测试集上进行测试。

MSCOCO 数据集总共有 80 类,本次实验使用 MSCOCO2017。其中,训练集有 118 287 张图片,验证集有 5 000 张图片,测试集一共有 40 670 张图片。本次实验通过将检测结果上传到评估服务器来报告测试结果。

本次实验指标使用的是 *mAP*。通过计算召回率和精确率,并绘制成 *PR* 曲线进行统计。指标的数学定义式可写为:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

其中,真阳性(Ture Positive, *TP*) 表示正确识别为物体样本;真阴性(True Negative, *TN*) 表示正确识别为背景样本;假阳性(False Positive, *FP*) 表示背景样本被识别为物体样本;假阴性(False Negative, *FN*) 表示物体样本被识别为背景样本。

### 2.2 模型与参数设置

本文实验基于 ubuntu18.04 系统,采用 Pytorch 深度学习框架和 Python 编程语言,硬件使用英伟达 GeForce GTX 1080 显卡。以 ResNet 为主干网络,每个 *batch* 大小为 2 张图片,使用 SGD 进行优化,动量为 0.9,初始学习率为 0.002 5。VOC0712 运行 12 个 *epoch*, MS COCO2017 运行 24 个 *epoch*, 第 16 个 *epoch* 和第 22 个 *epoch* 分别出现了学习率的下降,下降后的学习率为0.000 25和0.000 025。

### 2.3 实验结果与分析

本文对提出的双分类头进行了实验验证,表 1 为近年来一些网络框架在 VOC0712 中的实验结果。由表 1 可以看出,相比于作为 baseline 的 RetinaNet,本次研究提出的双分类头网络构架 RetinaNet-DCH 具有很高的网络性能, *mAP* 上达到了 80.8,提高了 3.5%。

表 2~表 6 为本次双分类头的消融实验。由表 2 可知,双分类头权重自适应相比于手动设置权重,能达到更好的效果。由表 3 可知,循环通道注意力模块头循环次数为 2 时效果最好。表 4 的实验结果表明多头注意力机制模块头中使用单头就能达到理想的效果。表 5 中使用了不同大小的平均池化,当使用 16×16 时,效果好、参数少。表 6 分别对 2 个头部分开实验,相比于循环通道注意力模块头而言,实验表明多头注意力机制模块头能带来更好的实验效果,2 个网络头部一起使用时性能最好。

表 7 和表 8 是运用 MSCOCO2017 数据集后的实验结果。表 7 在 ResNet50 上进行实验,表 8 在 ResNet101 上进行实验,表 8 中 RetinaNet-DCH-MS 为增加了多尺度训练的结果。实验表明本文提出的 RetinaNet-DCH 网络框架具有很高的性能。相比于两阶段网络 Double-Head-Ext, 一阶段网络 RetinaNet-DCH-MS 只相差了 0.2%的 *mAP*。

**表 1　VOC0712 数据集实验结果**

**Tab. 1　VOC0712 dataset experimental results**

| Method | Backbone | *mAP* | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSD300 | VGG16 | 0.743 | 0.755 | 0.802 | 0.723 | 0.663 | 0.476 | 0.830 | 0.842 | 0.861 | 0.547 | 0.783 | 0.739 | 0.845 | 0.853 | 0.826 | 0.762 | 0.486 | 0.739 | 0.760 | 0.834 | 0.740 |
| ION300 | VGG16 | 0.756 | 0.792 | 0.831 | 0.776 | 0.656 | 0.549 | 0.854 | 0.851 | 0.870 | 0.544 | 0.806 | 0.738 | 0.853 | 0.822 | 0.822 | 0.744 | 0.471 | 0.758 | 0.727 | 0.842 | 0.804 |
| Faster | VGG16 | 0.732 | 0.765 | 0.790 | 0.709 | 0.655 | 0.521 | 0.831 | 0.847 | 0.864 | 0.520 | 0.819 | 0.657 | 0.848 | 0.846 | 0.775 | 0.767 | 0.388 | 0.736 | 0.739 | 0.830 | 0.726 |
| Faster | Residual-101 | 0.764 | 0.798 | 0.807 | 0.762 | 0.683 | 0.559 | 0.851 | 0.853 | 0.898 | 0.567 | 0.878 | 0.694 | 0.883 | 0.889 | 0.809 | 0.784 | 0.417 | 0.786 | 0.798 | 0.853 | 0.720 |
| MR-CNN | VGG16 | 0.782 | 0.803 | 0.841 | 0.785 | 0.708 | 0.685 | 0.880 | 0.859 | 0.878 | 0.603 | 0.852 | 0.737 | 0.872 | 0.865 | 0.850 | 0.764 | 0.485 | 0.763 | 0.755 | 0.850 | 0.810 |
| R-FCN | Residual-101 | 0.805 | 0.799 | 0.872 | 0.815 | 0.720 | 0.698 | 0.868 | 0.885 | 0.898 | 0.670 | 0.881 | 0.745 | 0.898 | 0.906 | 0.799 | 0.812 | 0.537 | 0.818 | 0.815 | 0.859 | 0.799 |
| DSSD321 | Residual-101 | 0.786 | 0.819 | 0.849 | 0.805 | 0.684 | 0.539 | 0.856 | 0.862 | 0.889 | 0.611 | 0.835 | 0.787 | 0.867 | 0.887 | 0.867 | 0.797 | 0.517 | 0.780 | 0.809 | 0.872 | 0.794 |
| ASSD | VGG16 | 0.800 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| R-SSD | VGG16 | 0.785 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| R-SSD(4) | VGG16 | 0.762 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| R-SSD(6) | VGG16 | 0.770 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| YOLO V2 | DarkNet-19 | 0.737 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| DSOD | DS/64-192-48-1 | 0.777 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| FSSD | VGG16 | 0.788 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| RefineDet320 | VGG16 | 0.800 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| PELEE | VGG16 | 0.709 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| RetinaNet | Residual-50 | 0.773 | 0.828 | 0.843 | 0.819 | 0.692 | 0.676 | 0.846 | 0.877 | 0.882 | 0.642 | 0.780 | 0.642 | 0.860 | 0.834 | 0.812 | 0.835 | 0.526 | 0.758 | 0.712 | 0.833 | 0.779 |
| RetinaNet-DCH | Residual-50 | 0.808 | 0.881 | 0.863 | 0.839 | 0.715 | 0.706 | 0.853 | 0.882 | 0.888 | 0.670 | 0.874 | 0.737 | 0.863 | 0.863 | 0.841 | 0.849 | 0.568 | 0.831 | 0.791 | 0.845 | 0.808 |

**表 2　VOC0712 数据集权重分配实验结果**

**Tab. 2　VOC0712 dataset weight distribution experimental results**

| 权重分配 | *mAP* | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5∶0.5 | 0.793 | 0.857 | 0.851 | 0.815 | 0.693 | 0.686 | 0.845 | 0.878 | 0.878 | 0.648 | 0.849 | 0.725 | 0.842 | 0.849 | 0.819 | 0.843 | 0.570 | 0.805 | 0.774 | 0.839 | 0.802 |
| 0.4∶0.6 | 0.764 | 0.817 | 0.839 | 0.778 | 0.649 | 0.674 | 0.811 | 0.875 | 0.862 | 0.618 | 0.817 | 0.673 | 0.821 | 0.832 | 0.800 | 0.832 | 0.487 | 0.778 | 0.756 | 0.795 | 0.762 |
| 0.6∶0.4 | 0.795 | 0.867 | 0.851 | 0.799 | 0.696 | 0.701 | 0.846 | 0.882 | 0.878 | 0.638 | 0.852 | 0.732 | 0.843 | 0.858 | 0.826 | 0.840 | 0.557 | 0.806 | 0.768 | 0.857 | 0.799 |
| 0.7∶0.3 | 0.792 | 0.861 | 0.851 | 0.807 | 0.690 | 0.673 | 0.832 | 0.882 | 0.879 | 0.667 | 0.854 | 0.700 | 0.847 | 0.868 | 0.811 | 0.844 | 0.552 | 0.825 | 0.768 | 0.833 | 0.797 |
| 0.3∶0.7 | 0.793 | 0.857 | 0.851 | 0.806 | 0.704 | 0.703 | 0.836 | 0.877 | 0.873 | 0.858 | 0.849 | 0.728 | 0.823 | 0.860 | 0.819 | 0.842 | 0.538 | 0.814 | 0.769 | 0.847 | 0.799 |
| 自适应 | 0.808 | 0.881 | 0.863 | 0.839 | 0.715 | 0.706 | 0.853 | 0.882 | 0.888 | 0.670 | 0.874 | 0.737 | 0.863 | 0.863 | 0.841 | 0.849 | 0.568 | 0.831 | 0.791 | 0.845 | 0.808 |

**表 3　VOC0712 数据集循环次数实验结果**

**Tab. 3　The experimental results of the number of cycles of the VOC0712 dataset**

| 循环次数 | *mAP* | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.794 | 0.867 | 0.856 | 0.807 | 0.705 | 0.698 | 0.845 | 0.879 | 0.869 | 0.651 | 0.850 | 0.722 | 0.839 | 0.859 | 0.826 | 0.845 | 0.557 | 0.817 | 0.787 | 0.817 | 0.793 |
| 2 | 0.808 | 0.881 | 0.863 | 0.839 | 0.715 | 0.706 | 0.853 | 0.882 | 0.888 | 0.670 | 0.874 | 0.737 | 0.863 | 0.863 | 0.841 | 0.849 | 0.568 | 0.831 | 0.791 | 0.845 | 0.808 |
| 3 | 0.793 | 0.842 | 0.837 | 0.799 | 0.712 | 0.706 | 0.857 | 0.874 | 0.879 | 0.654 | 0.849 | 0.738 | 0.847 | 0.865 | 0.825 | 0.841 | 0.535 | 0.822 | 0.776 | 0.832 | 0.778 |

**表 4　VOC0712 数据集多头实验结果**

**Tab. 4　VOC0712 dataset multi-head experimental results**

| 多头个数 | *mAP* | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.808 | 0.881 | 0.863 | 0.839 | 0.715 | 0.706 | 0.853 | 0.882 | 0.888 | 0.670 | 0.874 | 0.737 | 0.863 | 0.863 | 0.841 | 0.849 | 0.568 | 0.831 | 0.791 | 0.845 | 0.808 |
| 2 | 0.796 | 0.844 | 0.844 | 0.816 | 0.698 | 0.684 | 0.841 | 0.878 | 0.875 | 0.660 | 0.864 | 0.728 | 0.823 | 0.852 | 0.848 | 0.849 | 0.557 | 0.837 | 0.792 | 0.835 | 0.804 |
| 4 | 0.808 | 0.855 | 0.865 | 0.840 | 0.740 | 0.707 | 0.848 | 0.882 | 0.885 | 0.669 | 0.877 | 0.740 | 0.866 | 0.867 | 0.839 | 0.849 | 0.566 | 0.833 | 0.785 | 0.843 | 0.811 |

**表 5　VOC0712 数据集平均池化实验结果**

**Tab. 5　Average pooling experimental results of VOC0712 dataset**

| 池化大小 | *mAP* | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 * 8 | 0.807 | 0.870 | 0.861 | 0.831 | 0.723 | 0.708 | 0.857 | 0.883 | 0.892 | 0.665 | 0.876 | 0.741 | 0.860 | 0.861 | 0.840 | 0.846 | 0.566 | 0.824 | 0.776 | 0.846 | 0.811 |
| 16 * 16 | 0.808 | 0.860 | 0.866 | 0.839 | 0.724 | 0.708 | 0.859 | 0.883 | 0.890 | 0.668 | 0.861 | 0.751 | 0.862 | 0.871 | 0.829 | 0.847 | 0.565 | 0.825 | 0.790 | 0.840 | 0.817 |
| 32 * 32 | 0.808 | 0.881 | 0.863 | 0.839 | 0.715 | 0.706 | 0.853 | 0.882 | 0.888 | 0.670 | 0.874 | 0.737 | 0.863 | 0.863 | 0.841 | 0.849 | 0.568 | 0.831 | 0.791 | 0.845 | 0.808 |

表 6 VOC0712 数据集头部实验结果

Tab. 6 Experimental results of the head of the VOC0712 dataset

| 网络头部 | *mAP* | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DCH | 0.808 | 0.881 | 0.863 | 0.839 | 0.715 | 0.706 | 0.853 | 0.882 | 0.888 | 0.670 | 0.874 | 0.737 | 0.863 | 0.863 | 0.841 | 0.849 | 0.568 | 0.831 | 0.791 | 0.845 | 0.808 |
| CCAM | 0.793 | 0.846 | 0.838 | 0.820 | 0.693 | 0.695 | 0.856 | 0.878 | 0.889 | 0.649 | 0.849 | 0.686 | 0.863 | 0.859 | 0.831 | 0.848 | 0.541 | 0.809 | 0.759 | 0.848 | 0.804 |
| MHA | 0.798 | 0.843 | 0.860 | 0.819 | 0.722 | 0.678 | 0.855 | 0.882 | 0.878 | 0.654 | 0.855 | 0.741 | 0.863 | 0.857 | 0.832 | 0.844 | 0.561 | 0.797 | 0.783 | 0.826 | 0.808 |

表 7 MSCOCO2017 数据集实验结果

Tab. 7 Experimental results on MSCOCO2017 dataset

| Method | Backbone | *AP* | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| Faster R-CNN | VGG16 | 21.9 | 42.7 | – | – | – | – |
| SSD | VGG16 | 28.8 | 48.5 | 30.3 | 10.9 | 31.8 | 43.5 |
| RefineNet | VGG16 | 33.0 | 54.5 | 35.5 | 16.3 | 36.3 | 44.3 |
| Faster R-CNN | ResNet-50-C4 | 34.8 | 55.8 | 37.0 | 19.1 | 38.8 | 48.2 |
| FPN | ResNet-50 | 36.8 | 58.7 | 40.4 | 21.2 | 40.1 | 48.8 |
| RetinaNet | ResNet-50 | 37.4 | 56.7 | 39.6 | 20.0 | 40.7 | 49.7 |
| RetinaNet-DCH(本文) | ResNet-50 | 38.7 | 58.7 | 41.0 | 22.2 | 41.4 | 49.0 |
| Double-Head | ResNet-50 | 39.8 | 59.6 | 43.6 | 22.7 | 42.9 | 53.1 |
| Double-Head-Ext | ResNet-50 | 40.3 | 60.3 | 44.2 | 22.4 | 43.3 | 54.3 |

表 8 MSCOCO2017 数据集实验结果

Tab. 8 Experimental results on MSCOCO2017 dataset

| Method | Backbone | *AP* | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| YOLOv2 | DarkNet-19 | 21.6 | 44.0 | 19.2 | 5.0 | 22.4 | 35.5 |
| SSD | ResNet-101 | 31.2 | 50.4 | 33.3 | 10.2 | 34.5 | 49.8 |
| YOLOv3 | DarkNet-53 | 33.0 | 57.9 | 34.4 | 18.3 | 35.4 | 41.9 |
| RefineNet[21] | ResNet-101 | 36.4 | 57.5 | 39.5 | 16.6 | 39.9 | 51.4 |
| Mask RCNN | ResNet-101 | 38.2 | 60.3 | 41.7 | 20.1 | 41.1 | 50.2 |
| RetinaNet | ResNet-101 | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| Deep Regionlets[20] | ResNet-101 | 39.3 | 59.8 | – | 21.7 | 43.7 | 50.9 |
| ExtremeNet | Hourglass-104 | 40.1 | 55.3 | 43.2 | 20.3 | 43.2 | 53.1 |
| RetinaNet-DCH(本文) | ResNet-101 | 40.2 | 60.3 | 43.0 | 23.1 | 43.1 | 51.3 |
| CornerNet | Hourglass-104 | 40.5 | 56.5 | 43.1 | 19.4 | 42.7 | 53.9 |
| FCOS | ResNet-101 | 41.5 | 60.7 | 45.0 | 24.4 | 44.8 | 51.6 |
| Double-Head | ResNet-101 | 41.5 | 61.7 | 45.6 | 23.8 | 45.2 | 54.9 |
| RetinaNet-DCH-MS(本文) | ResNet-101 | 41.7 | 61.6 | 45.4 | 25.2 | 44.7 | 52.0 |
| Double-Head-Ext | ResNet-101 | 41.9 | 62.4 | 45.9 | 23.9 | 45.2 | 55.8 |

图 10~图 12 为在 COCO 数据集上、RetinaNet 网络使用了双分类头后网络头部的热力图输出,左图为最初进入网络头部的热力图输出,右图为经过网络头部后的热力图输出。可以看出,在经过网络头部前的热力图相差不大,但在经过网络头部后的热力图输出相差很大。其中,2 个比较明显的提升在于:相比于之前的网络框架能更准确地识别到物体本身;对于识别到的物体能够更加地精准,不会有大量背景混入其中。

图 13 为在 COCO 数据集上物体原图的检测输

出，图 13（a）为 RetinaNet 网络输出，图 13（b）为 RetinaNet-DCH 网络输出。从图 13 中可以明显看出，改进后的网络框架目标定位和识别更加精准，识别到更多的物体的同时，分类的置信度也更高，特别对于小物体的识别准确率提升明显。

**图 10　网络头部热力图**

**Fig. 10　Network head heat map**

**图 11　网络头部热力图**

**Fig. 11　Network head heat map**

**图 12　网络头部热力图**

**Fig. 12　Network head heat map**

(a) RetinaNet网络输出

(b) RitinaNet-DCH 网络输出

**图 13　COCO2017 数据集上的结果图**

**Fig. 13　The result graph on the COCO2017 dataset**

## 3　结束语

本文对一阶段目标检测网络头部输出热力图，发现经过网络头部后提取的特征存在着识别不到物体和识别物体时范围太大两个问题，同时结合现如今两阶段目标检测头部研究成果，对一阶段目标检测网络头部进行改进提出了双分类头：循环通道注意力模块头、多头注意力机制模块头。其中，循环通道注意力模块头在识别物体时能够减小识别范围，准确识别物体。多头注意力机制模块头则通过利用多头注意力机制能够获得更好的空间性和自注意力，更好地识别到物体。利用池化可减少参数量，满足了可操作性。通过自适应权重操作，网络对 2 个头部获得最佳的权重分配、结合。本文基于 RetinaNet 在 VOC0712 和 COCO2017 数据集上进行实验，在 *mAP* 上达到了 80.8% 和 41.7%，分别提升了 3.5% 和 2.4%。

## 参考文献


[1] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] //2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus: IEEE, 2014: 580-587.

[2] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition [M]//FLEET D, PAJDLA T, SCHIELE B, TUYTELAARS T. Computer Vision-ECCV 2014. ECCV 2014. Lecture Notes in Computer Science. Cham: Springer, 2014, 8691: 346-361.

[3] GIRSHICK R. Fast R-CNN [C]// 2015 IEEE International Conference on Computer Vision. Santiago: IEEE, 2015: 1440-1448.

[4] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39 (6): 1137-1149.

[5] DAI J, LI Y, HE K, et al. R-FCN: Object detection via region-based fully convolutional networks [EB/OL]. [2016]. http://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf.
[6] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection [C]// The IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017:936-944.
[7] HE K, GKIOXARI G, DOLLÁR P, ET AL. MASK R-CNN[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2):386-397.
[8] LIU Wei, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector [M]// European Conference on Computer Vision. Cham:Springer, 2016,9905:21-37.
[9] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection [C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA:IEEE, 2016:779-788.
[10] REDMON J, FARHADI A. YOLO9000: better, faster, stronger [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA:IEEE, 2017:6517-6525.
[11] FARHADI A, REDMON J. Yolov3: An incremental improvement [C]//Computer Vision and Pattern Recognition. Salt Lake City, USA:IEEE, 2018:1-6.
[12] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence , 2020,42(2):318-327.
[13] TIAN Zhi, SHEN Chunhua, CHEN Hao, et al. Fcos: Fully convolutional one-stage object detection [C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South):IEEE, 2019:9626-9635.
[14] ZHOU Xingyi, ZHUO Jiacheng, KRAHENBUHL P. Bottom-up object detection by grouping extreme and center points [C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA:IEEE,2019:850-859.
[15] LAW H, DENG Jia. Cornernet: Detecting objects as paired keypoints [C]// The European Conference on Computer Vision. Munich:IEEE, 2018:1-17.
[16] ZHU Chenchen, HE Yihui, SAVVIDES M. Feature selective anchor-free module for single-shot object detection [C]//IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019. Long Beach, CA, USA:IEEE, 2019:840-849.
[17] CAI Zhaowei, VASCONCELOS N. Cascade R-CNN: Delving into high quality object detection [C]//The IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018:6154-6162.
[18] JIANG Borui, LUO Ruixuan, MAO Jiayuan, et al. Acquisition of localization confidence for accurate object detection [M]// FERRARI V, HEBERT M, SMINCHISESCU C, et al. Computer Vision-ECCV 2018. ECCV 2018. Lecture Notes in Computer Science() . Cham:Springer, 2018,11218:816-832.
[19] WU Yue, CHEN Yinpeng, YUAN Lu, et al. Rethinking classification and localization for object detection [C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, 2020: 10183-10192.
[20] XU Hongyu, LV Xutao, WANG Xiaoyu, et al. Deep regionlets for object detection [C]//the European Conference on Computer Vision. Munich, Germany:dblp, 2018:827-844.
[21] LIN Guosheng , MILAN A, SHEN Chunhua, et al. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA: IEEE, 2017:5168-5177.

(上接第 77 页)

[21] BASAK H, KUNDU R, SARKAR R. MFSNet: A multi focus segmentation network for skin lesion segmentation [J]. Pattern Recognition,2022, 128: 108673.
[22] ROTHER C, KOLMOGOROV V, BLAKE A. " GrabCut" interactive foreground extraction using iterated graph cuts [J]. ACM Transactions on Graphics (TOG), 2004,23(3): 309-314.
[23] CHEN L C, PAPANDREOU G, KOKKINOS I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017,40(4): 834-848.
[24] KAMNITSAS K, LEDIG C, NEWCOMBE V F, et al. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation [J]. Medical Image Analysis,2017 36: 61-78.
[25] KRÄHENBÜHL P, KOLTUM V. Efficient inference in fully connected crfs with gaussian edge potentials [C]// Advances in Neural Information Processing Systems. Spain:NIPS Foundation, 2011,24:109-117.
[26] SZUMMER M, KOHLI P, HOIEM D. Learning CRFs using graph cuts [C]// European Conference on Computer Vision. Berlin/Heidelberg:Springer ,2008:582-595.
[27] CODELLAN C F, GUTMAN D, CELEBI M E, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomed-ical imaging (isbi), hosted by the international skin imaging collaboration (isic) [C]//2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). Maryland, USA:IEEE, 2018:168-172.
[28] OKTAY O, SCHLEMPER J, FOLGOC L L, et al. Attention u-net: Learning where to look for the pancreas [J]. arXiv preprint arXiv:1804.03999, 2018.
[29] LEI Baiying, XIA Zaimin, JIANG Feng, et al. Skin lesion segmentation via generative adversarial networks with dual discriminators [J]. Medical Image Analysis, 2020,64:101716.
[30] CHEN Jieneng, LU Yongyi, YU Qihang, et al. Transunet: Transformers make strong encoders for medical image segmentation [J]. arXiv preprint arXiv:2102.04306,2021.
[31] VALANARASU J M J, OZA P, HACIHALILOGLU I, et al. Medical transformer: Gated axial-attention for medical image segmentation [C]//International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2021: 36-46.