遥感图像智能解译技术挑战赛说明材料

businiaoo
西安交通大学

目　录

1　队伍介绍

用户名：GoodGoodStudy
团队名：GoodGoodStudy
单位：西安交通大学
详情信息：
姓名：businiaoo 研究生二年级
研究方向：弱小目标检测、变化检测
联系方式：a1182693164@stu.xjtu.edu.cn

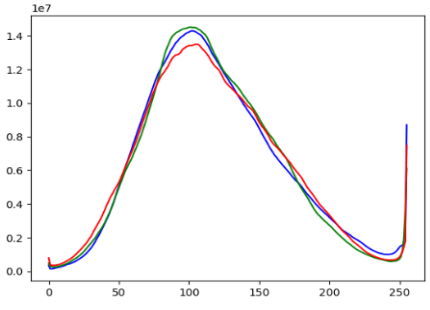
2　数据分析

2.1 图像数量与正负样本比例

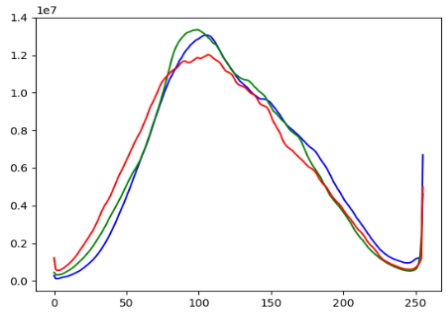
数据集	图像数量	尺寸	正样本比例	负样本比例
train	6000		22.48%	77.52%
val	2000	512*512*3	24.37%	75.63%
test	2000			

2.2 图像直方图统计

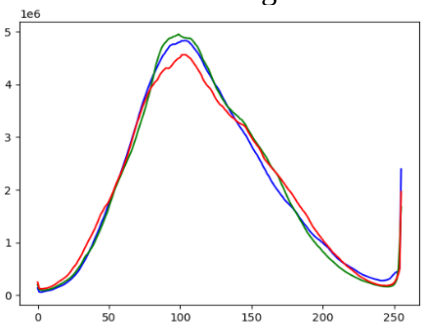
统计 train/image1 下的所有图像的直方图，如下图中的左上角的图；统计 train/image2 下的所有图像的直方图，如下图中的右上角的图。其他图像同理。



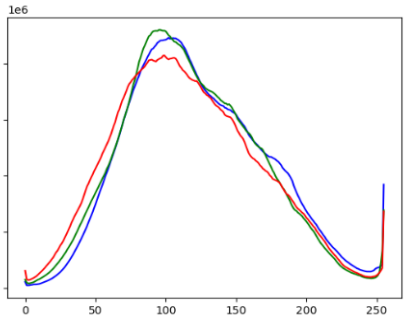
train/image1



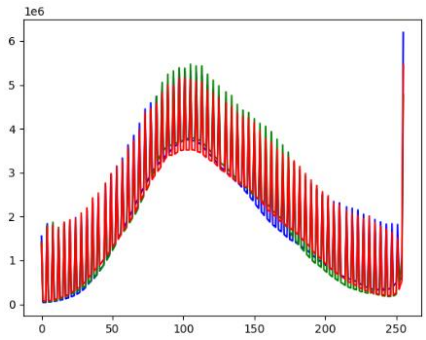
train/image2



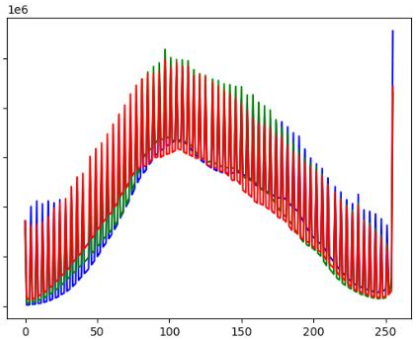
val/image1



val/image2

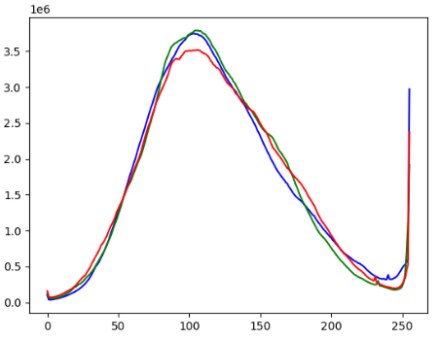


test/image1

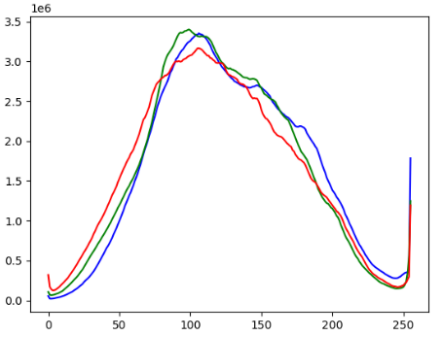


test/image2

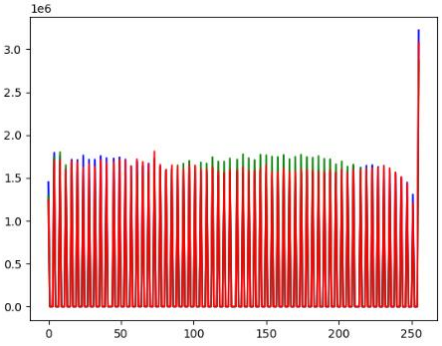
可以发现，val 与 train 的数据直方图基本一致，但是 test 的直方图非常奇怪，有大量尖峰。进一步，通过查看图像发现，编号为 9601-10000 的图像色彩不对劲。于是将 test 集的前 1600 张图像与后 400 张图像分别统计其直方图，如下。



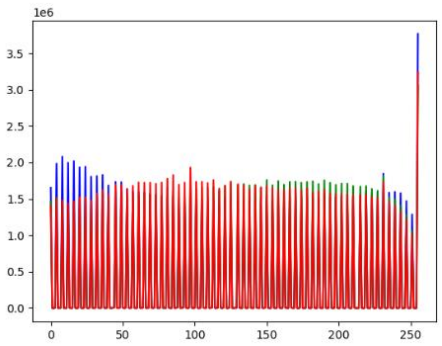
test/image1(前 1600)



test/image2(前 1600)



test/image1(后 400)

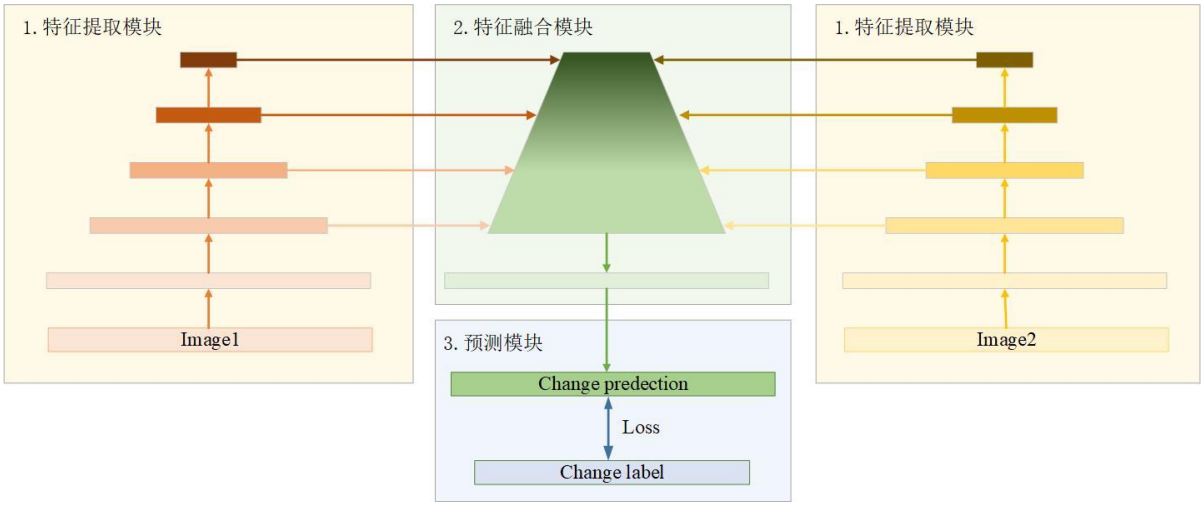


test/image1(后 400)

可以看出，test 中前 1600 张图像与 train&val 的直方图基本一致，但是后 400 张图像的直方图是不连续的均匀分布，与 train 的直方图分布相差甚远，进一步查看图像，其分辨率也较训练集的图像低；这成为决赛阶段的研究重点，如何使得算法能自动适应这种不同于训练集分布及分辨率的数据，详细讨论将在第 3 节算法介绍中展开。

3　算法介绍

3.1 整体框架



如上图，整个网络由三部分构成：

1. 两个共享参数的特征提取模块（backbone），用来提取双时像影像的多尺度特征；
2. 特征融合模块，融合提取到的多尺度特征；
3. 预测模块，根据融合之后的特征进行变化预测，并与变化标签计算损失，进行反向传播。

3.2 数据增强

为了提高泛化性能，在数据输入端采用了数据增强，主要包括以下几种增强手段：

1. 随机直方图均衡化
2. 随机颜色抖动（HSV 空间）
3. 随机上下左右翻转
4. 随机旋转
5. 随机多尺度([0.5-1.5])
6. 随机交换双时像图像顺序
7. 归一化

3.3 特征提取网络

特征提取网络用来提取双时像影像的多尺度特征，输入为两个图像，对每个图像输出 4 个尺度的特征。

这里主要采用了两种特征提取网络：Efficientnetv2_s 和 CSwin Transformer tiny。

1. Efficientnetv2[1]

Efficientnetv2 是在 Fused-MBConv 与 MBConv 两种基本模块的基础上，使用 NAS 搜索，得到的网络架构，相比于 Efficientnetv1 在训练速度有 6 倍以上提升，参数量上

进一步下降，具体的参数量与计算量分析将在第 4 节实验细节展开。

Table 4. EfficientNetV2-S architecture – MBConv and Fused-MBConv blocks are described in Figure 2.

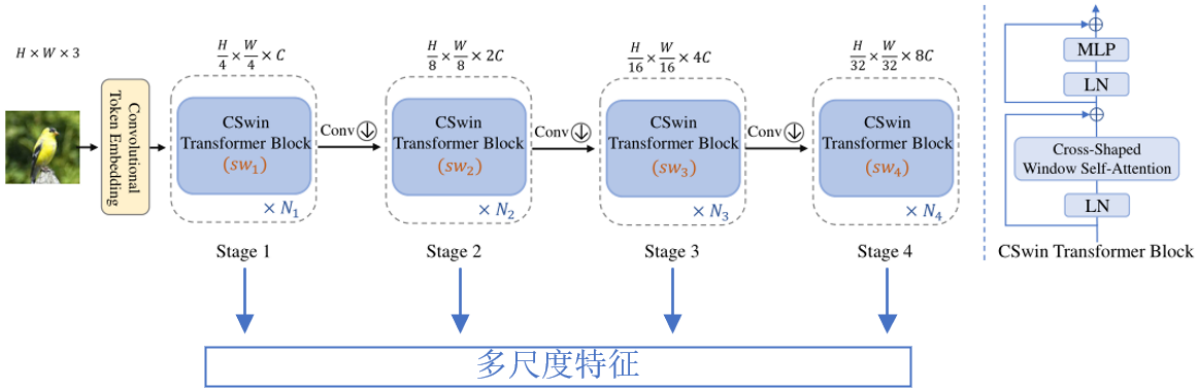
Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

多尺度特征

上表[1]为 EfficientNetv2_s 的结构，本次比赛采用前 6 个 stage，未使用第 7 个 stage(黄色区域)，分别输出 stage3、stage4、stage5、stage6 的特征，构成四个尺度的多尺度特征。

2. CSwin Transformer[2]

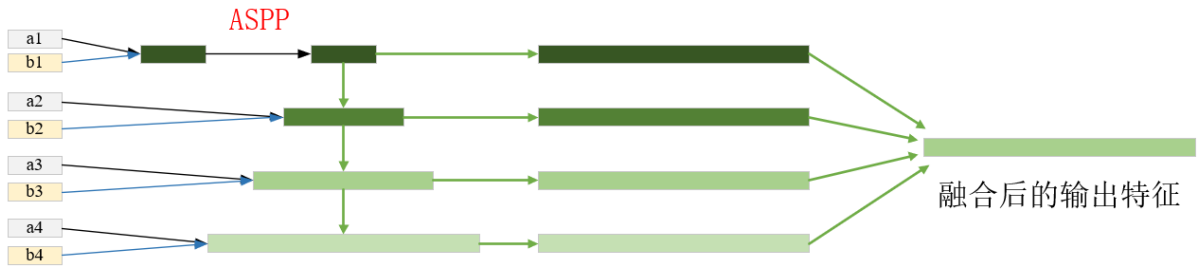
Transformer 的结构近来在很多视觉任务上已经超越了 CNN，本次比赛也使用了 CSwin Transformer tiny 模型，它是 Swin Transformer[3]的进阶版，通过引入十字交叉状的自注意力机制和 LePE，大大减少了计算量，同时在语义分割任务上取得了 SOTA。



上图[2]为 CSwin 的整体结构图，输出 stage1、stage2、stage3、stage4 的特征，作为后续多尺度特征融合的输入。

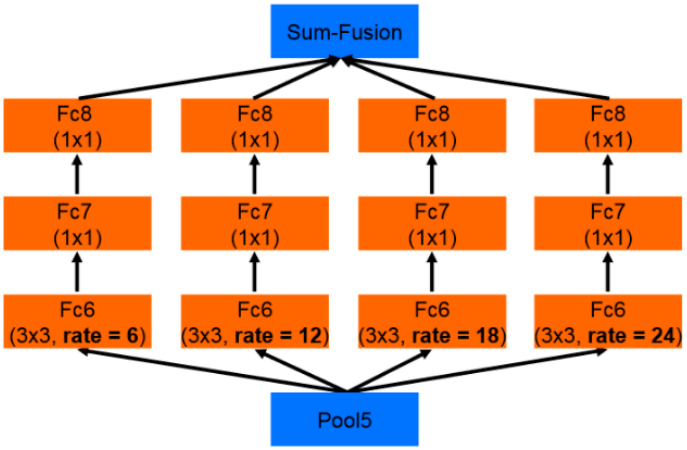
3.4 特征融合网络

特征融合网络是对多尺度特征进行融合，输入为 8 个(每张图 4 个)多尺度特征，输出为 1 个融合之后的变化检测特征。



本次比赛借鉴 FPN[4]与 Deeplabv2[5]，提出如上图所示的特征融合网络。首先，

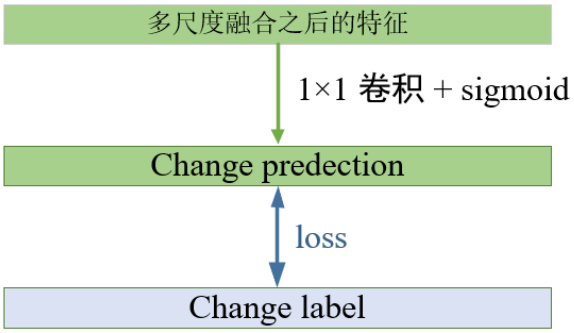
最左侧为 8 个输入特征，a 和 b 表示一张图像，a1 表示尺度最小的特征图，a4 表示尺度最大的特征图。两个同一尺度的特征进行级联，最顶层的特征通过 ASPP[5]模块，增大感受野；然后与 FPN[4]一样自上而下进行特征融合；最后将初步融合的 4 个尺度的特征都 resize 到同一大小，然后级联形成最终的输出特征图。通过这种多尺度融合的方式，增强了模型对不同尺度建筑物的感知能力。



ASPP[5]模块的结构如上图，利用不同膨胀因子(6, 12, 18, 24)的空洞卷积融合多尺度信息。不做池化损失信息的情况下，保证了低计算量，加大了感受野，让每个卷积输出都包含较大范围的信息。

3.5 预测模块

通过预测模块得到最终的变化区域预测，输入为经过多尺度融合之后的特征，输出为变化预测。得到变化预测之后，进一步与真实标签计算 loss。



如上图，这一部分的实现非常简单，得到融合之后的特征之后，通过一个 1*1 的卷积层和 sigmoid 激活层便可得到变化预测。

$$Loss = loss\_dice + loss\_bce$$

在与真值标签计算 loss 的时候采用 dice 与 bce 的混合 loss。dice 是针对交并比优化的损失函数，本次比赛采用 miou 作为指标，这个 loss 很合适。

3.6 防止过拟合

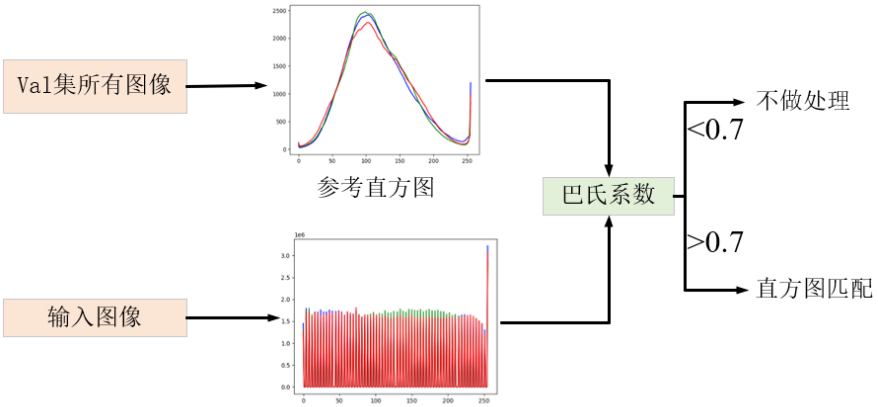
本次比赛采用了 dropout 与 dropblock[6]防止过拟合。具体来说，在预测模块的最

后一层之前使用 dropout，drop 率设置为 0.2；对特征融合网络的输入与输出部分使用 dropblock，block size 设置为 7*7，drop 率设置为 0.15，step 设置为 30。比赛前期使用 serenert50 对 dropblock 做消融实验如下表。

模型	是否使用 dropblock	miou	OA
Seresnet50	否	0.8137	0.9230
Seresnet50	是	0.8215	0.9274

3.7 同分布化处理

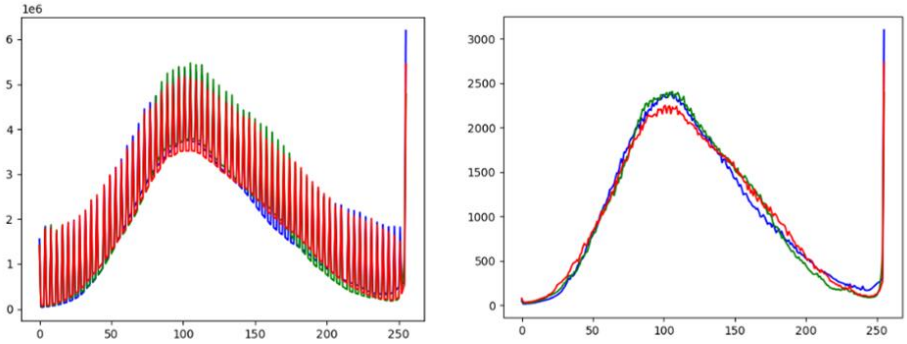
这一部分是 test 集发布之后，才意识到的问题。在实际中确实存在，输入图像与训练时的图像的分布不同。在本次比赛中采用直方图巴氏系数[0,1]作为衡量图像分布相关性的指标。



输入图像直方图

当输入图像直方图与参考直方图的巴氏系数小于 0.7 的时候，认为输入图像与训练图像同分布，此时不需要处理；反之，将输入图像的直方图通过直方图匹配的方式，与参考直方图进行匹配。

具体而言，参考直方图采用 val 集的平均直方图；巴氏系数的阈值选定过程如下：1. 假设 val 集有 n 张图像，统计 val 集的平均直方图，得到参考直方图；2. 遍历每一张 val 集的图像，分别与参考直方图计算巴氏系数，得到 n 个巴氏系数；3. 对得到的 n 个巴氏系数排序，取 1%处的最大值，作为巴氏系数阈值。



同分布化之前的直方图　　同分布化之后的直方图

测试集 image1 同分布化之前与之后的平均直方图如上图，可以看出，有明显改进。

4 实验细节

4.1 详细配置

本次比赛采用 adamw 作为优化器，权重衰减设置为 0.01；采用 Onecycle 学习率衰减策略，最大学习率为 0.0035，最小为 0.0035/500；epoch 设置为 150；batch size 为 16；EfficientNetv2_s 训练尺寸为 512；CSwin Transformer tiny 训练尺寸为 448；采用 SWA[7] 模型融合策略，额外训练 50 个 epoch，平均权重。

4.2 模型复杂度

本次比赛采用了两种模型，EfficientNetv2 与 CSwin。

下表为它们的训练复杂度。FLOPs 是输入 1 对图像计算的。参数量和 FLOPs 都是使用 thop（一个 python 包）统计出来的。显卡为单块 RTX3090。

模型	训练尺寸	参数量/M	FLOPs/G	训练速度/epoch
EfficientNetv2_s	512	27.536	38.537	7 分 20 秒
CSwin_t	448	41.523	52.183	9 分 33 秒

下表为它们的预测复杂度。经过测试，efficientNetv2_s 在 352 的输入尺度下，有更好的表现，因此 efficientnetv2_s 测试时采用 352 输入。速度测试是在 8bs 下测试的，此时 Efficientnet 占用显存 1654M，CSwin 占用显存 2240M。

模型	测试尺寸	参数量/M	FLOPs/G	测试速度/秒
EfficientNetv2_s	352	27.536	18.219	51 对图像
CSwin_t	448	41.523	52.183	29 对图像

下表为模型集成的测试效率，4×EfficientNetv2_s 表示 4 个 EfficientNetv2_s 模型集成。

模型	测试尺寸	测试速度/秒
EfficientNetv2_s+CSwin	448	21 对图像
4×EfficientNetv2_s	352	25 对图像

4.3 实验结果

a. 使用 TTA 之后，指标变差，所以以下测试结果均没有进行测试时增强(TTA)
b. Self-training：使用集成模型对 val 或 test 进行预测，保存预测结果；将 val 或 test 的图像以及对它们的预测结果加入训练集；对单模型进行微调。

4.3.1 val 集结果

模型	miou	OA
EfficientNetv2_s	0.8976	0.9599
CSwin_t	0.8962	0.9596
CSwin_t+ EfficientNetv2_s	**0.9065**	**0.9638**

Val 集使用 Self-training 结果如下表

模型	miou	OA
EfficientNetv2_s	0.9009	0.9613
CSwin_t	0.9056	0.9634
CSwin_t+ EfficientNetv2_s	0.9074	0.9641

可以看出，使用 self-training 之后，单模型的指标变得很高，但是集成之后的 miou 只有 0.09%升高。这种方式 self-training，类似于模型蒸馏的意思，使用集成模型的预测结果加到训练集中，让单个小模型去学习集成模型的结果，从而在单模型上有明显的指标提升，但对集成模型的指标提升作用不大。

4.3.2 Test 集结果

模型	miou	OA
EfficientNetv2_s+CSwin_t	0.8523	0.9423
EfficientNetv2_s +CSwin_t +同分布化处理	0.8655	0.9480
4×EfficientNetv2_s +同分布化处理+self-training	**0.8663**	**0.9483**

5 其他尝试

除了上文中提到的方法，在比赛的过程中还尝试了其他方法，对指标没有提升或者是提升效果没有上文中的方法大。

1. PPM[8]/SPP[9]模块（有提升，但不如 ASPP）
2. Cutout、随机平移像素数据增强（没有提升）
3. FaPN[10]，特征对齐金字塔（指标降低）
4. Focal loss[11] / topk loss（均不如 bce+dice）
5. CSwin_b / EfficientNetv2_m（没有提升，且模型太大）

6 参考文献

[1] EfficientNetV2: Smaller Models and Faster Training, 2021
[2] CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows, 2021
[3] Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, 2021
[4] Feature Pyramid Networks for Object Detection, 2017
[5] DeepLab v2: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, 2017
[6] DropBlock: A regularization method for convolutional networks, 2018
[7] Averaging Weights Leads to Wider Optima and Better Generalization, 2018
[8] Pyramid Scene Parsing Network, 2018
[9] Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, 2017
[10] FaPN: Feature-aligned Pyramid Network for Dense Image Prediction, 2021
[11] Focal Loss for Dense Object Detection, 2017