



QQ67658008

博客等级 码龄1年

109 1950 1664 630

原创 点赞 收藏 粉丝

关注 私信

TAB的精选

- 基于YOLOv8进行训练桥梁底部缺陷识别检测数据集、评估和推理。桥梁裂缝缺... 830 阅读
- 基于YOLOv8进行路面积水识别数据集+系统代码。涵盖准备、模型训练、... 1135 阅读
- voc怎么转yolo，如何分割数据集为验证集，怎样检测CUDA可用性并使用yolov8... 2157 阅读
- 基于LRW-1000 (CAS-VSR-W1k) 数据集来进行中文唇语数据集识别任务中... 2053 阅读
- YOLOv8的改进的方法 1743 阅读

查看更多 >

大家都在看

- 零门槛入行、年薪 30 万？年轻人正扎堆涌入学网络安全
- stm32 单片机USART 串口配置
- MySQL数据库——1.背景知识《干货笔记》 157
- 最新8月, 6+文献思路他来了, 贵州中医药大学这篇文章网络药理学+多组学还不来看者?
- C语言程序编译和链接的深度解析: 从源代码到可执行文件

最新8月, 6+文献思路他来了, 贵州中医药大学这篇文章网络药理学+多组学还不来看者?

C语言程序编译和链接的深度解析: 从源代码到可执行文件

基于Ultralytics的RT-DETR改进思考

原创 QQ67658008 于 2025-06-15 09:04:46 发布 阅读量 1k 收藏 27 点赞数 17

CC 4.0 BY-SA 版权

文章标签: Pytorch 人工智能 YOLO
G 天启AI社区 文章已被社区收录 加入社区

基于Ultralytics的RT-DETR改进思考参考

以下文字及代码仅供参考。

文章目录

: 基于Ultralytics的RT-DETR二次创新系列

1. 使用自研可变形卷积DCNV2 Dynamic改进resnet18 backbone中的BasicBlock
2. 使用ficientVIT CVPR2023中的CascadedGroupAttention对IRMB进行二次创新来改进resnet18-backbone中的BasicBlock
3. 使用RepVGG CVPR2021中的RepConv对FasterNet CVPR2023中的PConv进行二次创新后改进resnet18-backbone中的BasicBlock
4. 使用RepVGG CVPR2021中的RepConv对FasterNet CVPR2023中的Faster-Block进行二次创新后改进resnet18-backbone中的BasicBlock
5. 使用EMA ICASSP2023对FasterNet CVPR2023中的Faster-Block进行二次创新后改进resnet18-backbone中的BasicBlock
6. 使用RepVGG CVPR2021中的RepConv和EMA ICASSP2023对FasterNet CVPR2023中的Faster-Block进行二次创新后改进resnet18-backbone中的BasicBlock
7. 使用UniRepLNet中的DilatedReparamBlock对DWRSeg中的Diatin-wise Residual(DWR)进行二次创新改进rtdet
8. 在ultralytics/cfg/model/t_det/tdetr ASF yaml的基础上进行次创新, 引入P2检测层并对网络结构进行优化
9. 使用SlimNeck中的VGSCSPVVGSCSP和GSCConv和ASF_YOLO中的Attentional Scale Sequence Fusion改进rtdet中的CCFM

: 基于Ultralytics的RT-DETR改进思考-自研系列

自研系列

1. Parallel Atrous Convolution Attention Pyramid Network (PAC-APN)
2. Focusing Diffusion Pyramid Network (FDPN)
3. FDPN with Dimension-Aware Selective Integration Module (DA-SIM)
4. RepGhostsPELAN
5. Context Guide Fusion Module (CGFM)

实现细节与代码示例

- PAC-APN 示例代码
FDPN 示例代码

: 基于Ultralytics的RT-DETR二次创新系列

基于Ultralytics的RT-DETR模型进行的一系列改进的简要介绍和[配置文件](#)示例。每个改进点都旨在提升模型性能或效率, 通过修改基本块 (BasicBlock) 来优化 ResNet18 骨干网络。

1. 使用自研可变形卷积DCNV2 Dynamic改进resnet18 backbone中的BasicBlock

配置文件: [utrlicscfg/models/t_detr/tdetr.DCNV2_Dynamic.yaml](#)

yaml

```
1 backbone:
2   name: 'resnet18'
3   type: 'customized'
4   blocks:
5     BasicBlock: 'DCNV2_Dynamic' # 自定义BasicBlock使用DCNV2 Dynamic
```

AI运行代码 复制

2. 使用ficientVIT CVPR2023中的CascadedGroupAttention对IRMB进行二次创新来改进resnet18-backbone中的BasicBlock

配置文件: [utlytics/cfg/models/t_det/tdetr.IRMB-Cascaded.yaml](#)

yaml

```
1 backbone:
2   name: 'resnet18'
3   type: 'enhanced'
4   blocks:
5     BasicBlock: 'IRMB_Cascaded' # 使用IRMB CascadedGroupAttention改进
```

AI运行代码 复制

3. 使用RepVGG CVPR2021中的RepConv对FasterNet CVPR2023中的PConv进行二次创新后改进resnet18 backbone中的BasicBlock

配置文件: [ultalytics/cfg/models/t_det/rtdet-PConv_Rep.yaml](#)

yaml

```
1 backbone:
2   name: 'resnet18'
```

AI运行代码 复制

烈火超变

【圣神】落星·破荒者

武器】99价 攻击+30% 搞怪狂野功 双属性暴击+6%

开始游戏

目录

文章目录

: 基于Ultralytics的RT-DETR二次创新系列

展开全部 ▾

亚马逊云科技 性价比之选 - 基于 ARM 架构 Graviton4 处理器实例 性能提升 30%!

上一篇: 基于TransU-Net的遥感图像语义分割...
下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升 30%!
△ 广告

上一篇: 基于TransU-Net的遥感图像语义...
下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升 30%!
△ 广告

上一篇: 基于TransU-Net的遥感图像语义...
下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升 30%!
△ 广告

上一篇: 基于TransU-Net的遥感图像语义...
下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升 30%!

大学这篇文章网络药理学+多组学还不来看
看?
C语言程序编译和链接的深度解析:从源代
码到可执行文件

最新8月, 6+文献思路他来了, 贵州中医药
大学这篇文章网络药理学+多组学还不来看
看?
C语言程序编译和链接的深度解析:从源代
码到可执行文件

最新8月, 6+文献思路他来了, 贵州中医药
大学这篇文章网络药理学+多组学还不来看
看?

C语言程序编译和链接的深度解析:从源代
码到可执行文件
最新8月, 6+文献思路他来了, 贵州中医药
大学这篇文章网络药理学+多组学还不来看
看?
C语言程序编译和链接的深度解析:从源代
码到可执行文件

最新8月, 6+文献思路他来了, 贵州中医药
大学这篇文章网络药理学+多组学还不来看
看?
C语言程序编译和链接的深度解析:从源代
码到可执行文件

最新8月, 6+文献思路他来了, 贵州中医药
大学这篇文章网络药理学+多组学还不来看
看?
C语言程序编译和链接的深度解析:从源代
码到可执行文件

最新8月, 6+文献思路他来了, 贵州中医药
大学这篇文章网络药理学+多组学还不来看
看?
C语言程序编译和链接的深度解析:从源代
码到可执行文件

最新8月, 6+文献思路他来了, 贵州中医药
大学这篇文章网络药理学+多组学还不来看
看?
C语言程序编译和链接的深度解析:从源代
码到可执行文件

最新8月, 6+文献思路他来了, 贵州中医药
大学这篇文章网络药理学+多组学还不来看
看?
C语言程序编译和链接的深度解析:从源代
码到可执行文件

最新8月, 6+文献思路他来了, 贵州中医药
大学这篇文章网络药理学+多组学还不来看
看?
C语言程序编译和链接的深度解析:从源代
码到可执行文件

```
3 type: 'optimized'  
4 blocks:  
5   BasicBlock: 'PConv_Rep' # 使用RepConv优化后的PConv
```

4. 使用RepVGG CVPR2021中的RepConv对FasterNet CVPR2023中的Faster-Block进行二次创新后改进resnet18-backbone中的BasicBlock

配置文件: [ultralytics/cfg/models/t_detr/rtdetr-Faster_Rep.yaml](#)

```
yaml  
1 backbone:  
2   name: 'resnet18'  
3   type: 'fast_optimized'  
4   blocks:  
5     BasicBlock: 'Faster_Block_Rep' # 使用RepConv优化后的Faster-Block
```

5. 使用EMA ICASSP2023对FasterNet CVPR2023中的Faster-Block进行二次创新后改进resnet18-backbone中的BasicBlock

配置文件: [ultralytics/cfg/models/t_det/detr.Faster-EMA.yaml](#)

```
yaml  
1 backbone:  
2   name: 'resnet18'  
3   type: 'ema_optimized'  
4   blocks:  
5     BasicBlock: 'Faster_Block_EMA' # 使用EMA技术优化后的Faster-Block
```

6. 使用RepVGG CVPR2021中的RepConv和EMA ICASSP2023对FasterNet CVPR2023中的Faster.Block进行二次创新后改进resnet18-backbone中的BasicBlock

配置文件: [ultralytics/cfg/models/t_det/tdetr.Faster_Rep_EMA.yaml](#)

```
yaml  
1 backbone:  
2   name: 'resnet18'  
3   type: 'rep_ema_optimized'  
4   blocks:  
5     BasicBlock: 'Faster_Block_Rep_EMA' # 使用RepConv和EMA技术双重优化的Faster-Block
```

7. 使用UniRepLNet中的DilatedReparamBlock对DWRSeg中的Dlatin-wise Residual(DWR)进行二次创新改进rtdetr

配置文件: [ultralytics/cfg/models/t_detr/rtdetr-DWRC3-DRB.yaml](#)

```
yaml  
1 backbone:  
2   name: 'rtdetr'  
3   type: 'dilated_reparam_optimized'  
4   blocks:  
5     BasicBlock: 'DWRC3_DRB' # 使用DilatedReparamBlock优化的DWR
```

8. 在ultralytics/cfg/model/t_det/tdetr ASF yaml的基础上进行次创新, 引入P2检测层并对网络结构进行优化

配置文件: [ultralytics/cfg/models/t_det/tdetr.ASF_P2.yaml](#)

```
yaml  
1 model:  
2   name: 'tdetr_ASF_P2'  
3   layers:  
4     - 'P2_Detection_Layer' # 引入P2检测层  
5   optimizations:  
6     network_structure: 'optimized' # 网络结构优化
```

9. 使用SlimNeck中的VoVGSCSPVoVGSCSP和GSConv和ASF_YOLO中的Attentional Scale Sequence Fusion改进rtdetr中的CCFM

配置文件: [ultralytics/cfg/models/t_det/tdetr-slimneck-ASF.yaml](#)

```
yaml  
1 model:  
2   name: 'tdetr_slimneck_asf'  
3   neck:  
4     type: 'slim_neck'  
5     components:  
6       - 'VoVGSCSP'  
7       - 'GSConv'  
8     attention_mechanisms:  
9       - 'Attentional_Scale_Sequence_Fusion' # 使用ASF机制
```

这些配置文件提供了不同方面的优化方案, 用户可以根据具体需求选择合适的配置进行实验。每个配置文件都针对特定的改进进行了详细的说明, 并且可以通过简单的修改实现不同的组合和创新。希望这些信息能帮助您更好地理解和应用这些改进到您的项目中。

RT-DETR基准模型

1. ultralytics/cfg/models/t-det/rtdetr-r18.yaml(有预训练权重COCO+Objects365,来自RTDETR-Pytorch版本的移植)

△ 广告

上一篇: 基于TransU-Net的遥感图像语义分割: ...

下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇: 基于TransU-Net的遥感图像语义分割: ...

下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇: 基于TransU-Net的遥感图像语义分割: ...

下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇: 基于TransU-Net的遥感图像语义分割: ...

下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇: 基于TransU-Net的遥感图像语义分割: ...

下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇: 基于TransU-Net的遥感图像语义分割: ...

下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇: 基于TransU-Net的遥感图像语义分割: ...

下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇: 基于TransU-Net的遥感图像语义分割: ...

下一篇: 基于U-Net的CT脊柱语义分割: ...

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇: 基于TransU-Net的遥感图像语义分割: ...

下一篇: 基于U-Net的CT脊柱语义分割: ...

C语言程序编译和链接的深度解析：从源代码到可执行文件

广告

最新8月, 6+文献思路他来了, 贵州中医药大学这篇文章网络药理学+多组学还不来看着?

C语言程序编译和链接的深度解析：从源代码到可执行文件

广告

yaml

```
1 model:  
2   name: 'tdetr_FDPN_DA_SIM'  
3   feature_integration:  
4     module: 'DimensionAwareSelectiveIntegration' # DA-SIM模块  
5   scales: 3 # 多尺度处理  
6   context_guidance: True # 上下文引导
```

AI运行代码 复制

上一篇：基于TransU-Net的遥感图像语义分割：...

下一篇：基于U-Net的CT脊柱语义分割：...

4. RepGhostCSPELAN

配置文件：[ultralytics/clg/models/t_detr/tdetr-RGCSPELAN.yaml](#)

yaml

```
1 model:  
2   name: 'tdetr_RGCSPELAN'  
3   backbone_design:  
4     use_ghost_module: True # GhostNet思想  
5     rep_conv: True # 使用RepConv增强特征提取  
6     gradient_flow: True # 增强梯度流畅能力  
7     scale_factor: 1.0 # 可调整模型大小的缩放因子
```

AI运行代码 复制

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇：基于TransU-Net的遥感图像语义分割：...

下一篇：基于U-Net的CT脊柱语义分割：...

5. Context Guide Fusion Module (CGFM)

配置文件：[ultralytics/cfg/models/t_detr/tdetr-ContextGuideFPN.yaml](#)

yaml

```
1 model:  
2   name: 'tdetr_ContextGuideFPN'  
3   feature_pyramid_network:  
4     fusion_module: 'ContextGuideFusion' # CGFM模块  
5     se_attention: True # SE注意力机制  
6     weighted_reorganization: True # 权重重组操作
```

AI运行代码 复制

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇：基于TransU-Net的遥感图像语义分割：...

下一篇：基于U-Net的CT脊柱语义分割：...

实现细节与代码示例

对于上述配置文件中的各个模块，这里提供一些简化的代码示例来帮助理解其实现方式。请注意，以下代码仅为示例，需根据实际情况进一步完善和优化。

PAC-APN 示例代码

python

```
1 class ParallelAtrousConvAttention(nn.Module):  
2     def __init__(self, in_channels, out_channels, dilation_rates):  
3         super(ParallelAtrousConvAttention, self).__init__()  
4         self.branches = nn.ModuleList([  
5             nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=d, dilation=d) for d in dilation_  
6         ])  
7         self.gate = nn.Sequential(  
8             nn.Conv2d(out_channels * len(dilation_rates), out_channels, kernel_size=1),  
9             nn.Sigmoid()  
10        )
```

AI运行代码 复制

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇：基于TransU-Net的遥感图像语义分割：...

下一篇：基于U-Net的CT脊柱语义分割：...

FDPN 示例代码

python

```
1 class FocusingDiffusion(nn.Module):  
2     def __init__(self, scales):  
3         super(FocusingDiffusion, self).__init__()  
4         self.inception_style = nn.ModuleList([  
5             nn.Conv2d(in_channels, out_channels, kernel_size=k) for k in [1, 3, 5]  
6         ])  
7         self.scales = scales  
8  
9     def forward(self, inputs):  
10        ...
```

AI运行代码 复制

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇：基于TransU-Net的遥感图像语义分割：...

下一篇：基于U-Net的CT脊柱语义分割：...

以上代码提供了基础框架和思路，您可以根据具体需求进一步扩展和完善。希望这些信息能帮助您理解和应用这些自研模块到您的项目中。

只作为参考

QQ67658008 关注

17 27 0 分享 ...

基于ultralytics库的RTDETR独家大全（原创）

luoshiyong123的博客 ◎ 1353

RTDETR作为目标检测和分割领域的新兴技术，具有巨大的发展潜力。通过本专栏的基础入门和高阶创新改进部分的学习，读者将能够全面了解RTDETR...

RT-DETR改进策略【卷积层】| ECCV 2024 小波卷积WTConv 增大感受野，降低参数量计算量，独家创新助力涨点 Limiling的博客 ◎ 920

本文记录的是利用块优化的目标检测网络模型。目的是在不出现过参数化的情况下有效地增加卷积的感受野，从而解决了在感受野扩增中的参数膨胀...

基于Ultralytics的RT-DETR一些改进思考 (PAC-APN)、Focusing Diffusion...

9-28

Graviton4 处理器实例 性能提升30%!

△ 广告

上一篇：基于TransU-Net的遥感图像语义分割：...

下一篇：基于U-Net的CT脊柱语义分割：...

