TransGOP-R: Transformer-based Real-World Gaze复现

Data Preparation

The GOO dataset contains two subsets: GOO-Sync and GOO-Real.

You can download GOO-Synth dataset and annotations from Baidu Netdisk:

GOOsynth-data and annotations(code:166j)

You can download GOO-Real dataset and annotations from Baidu Netdisk:

GOOreal-data and annotations(code:pfni)

目录框架：

代码块

```
Please ensure the data structure is as below

├── Datasets
    └── goosynth
        └── annotations
            ├── train.json
            ├── val.json
        └── val
            ├── 0.png
            ├── 1.png
            ├── ...
        └── train
            ├── 0.png
            ├── 1.png
            ├── ...
    └── gooreal
         └── annotations
            ├── train.json
            ├── val.json
         └── val
            ├── 0.png
            ├── 1.png
            ├── ...
         └── train
            ├── 0.png
```

```
        ├── 1.png
        ├── ...
```

Environment Preparation

1.环境配置

代码块

```
conda env create -n TransGOP-R -f environment.yaml
conda init bash
source ~/.bashrc    # 刷新终端
conda activate TransGOP-R
```

2.手动下载MultiScaleDeformableAttention包：将文件install_MultiScaleDeformableAttention放入TransGOP-R文件的根目录中，执行setup.py文件

(1)链接：

通过网盘分享的文件：install_MultiScaleDeformableAttention

链接: https://pan.baidu.com/s/12kkkB-BptDhonniDMXvqDQ?pwd=n8p5 提取码: n8p5

(2)进入install_MultiScaleDeformableAttention路径中并执行setup.py文件

代码块

```
cd install_MultiScaleDeformableAttention
python setup.py build install
```

3.TransGOP-R_train.sh与TransGOP-R_val.sh脚本文件修改：自行修改对应的数据集位置，配置文件位置，输出文件位置

参考：

TransGOP-R_train.sh

```
#!/bin/bash

python main.py \
    -c config/TransGOP-R/TransGOP-R_4scale.py \
    --coco_path /root/autodl-tmp/TransGOP-R/datasets/gooreal \
    --output_dir logs/TransGOP-R/R50-MS4 \
    --options dn_scalar=100 embed_init_tgt=TRUE dn_label_coef=1.0 dn_bbox_coef=1.0 use_ema=False dn_box_noise_scale=1.0

```

TransGOP-R_val.sh

```
#!/bin/bash
python main.py \
    --config_file config/TransGOP-R/TransGOP-R_4scale.py \
    --output_dir logs/TransGOP-R/R50-MS4-eval \
    --coco_path /root/autodl-tmp/TransGOP-R/datasets/gooreal \
    --eval \
    --resume /root/autodl-tmp/TransGOP-R/logs/TransGOP-R/R50-MS4/checkpoint_best_regular.pth \
    --options dn_scalar=100 embed_init_tgt=TRUE \
    dn_label_coef=1.0 dn_bbox_coef=1.0 use_ema=False \
    dn_box_noise_scale=1.0
```

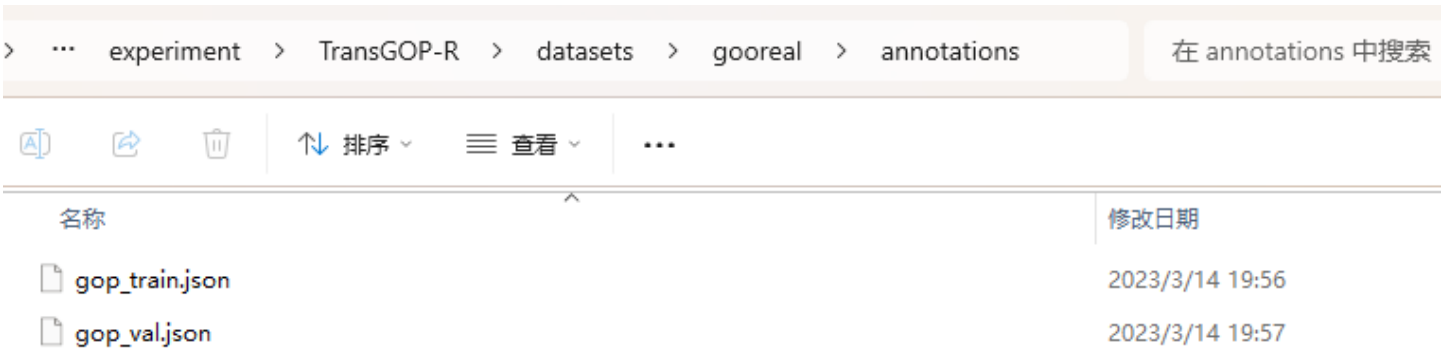
问题：

1.环境配置完后，仍有很多缺少的包，例如yapf，pycocotools，输入命令安装即可。

代码块

```
pip install yapf
pip install pycocotools
pip install opencv-python
pip install timm
pip install thop
```

2.在 datasets/coco.py 文件中，系统期望的是 train.json 和 val.json 文件，但实际存在的是 gop_train.json 和 gop_val.json 。



```python
def build(image_set, args):
    root = Path(args.coco_path)
    mode = 'multi'
    PATHS = {
        "train": (root / "train", root / "annotations" / 'train.json'),
        "train_reg": (root / "train", root / "annotations" / f'{mode}_train.json'),
        "val": (root / "val", root / "annotations" / 'val.json'),
        "eval_debug": (root / "val", root / "annotations" / f'{mode}_val.json'),
        "test": (root / "test", root / "annotations" / 'image_info_test-dev.json' ),
    }
```

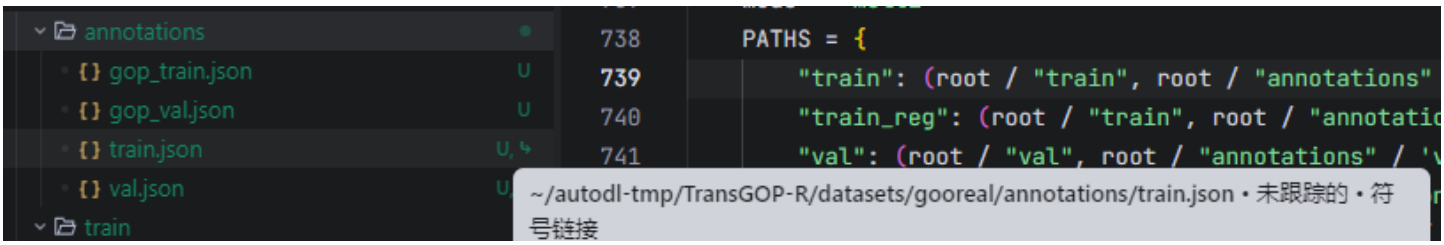
解决：

1.方法一：可以在数据集目录中更改文件名称。

2.方法二：**创建符号链接**来解决这个问题：

代码块

```
cd /root/autodl-tmp/TransGOP-R/datasets/gooreal/annotations && ln -s gop_train.json train.json && ln -s gop_val.json val.json
```

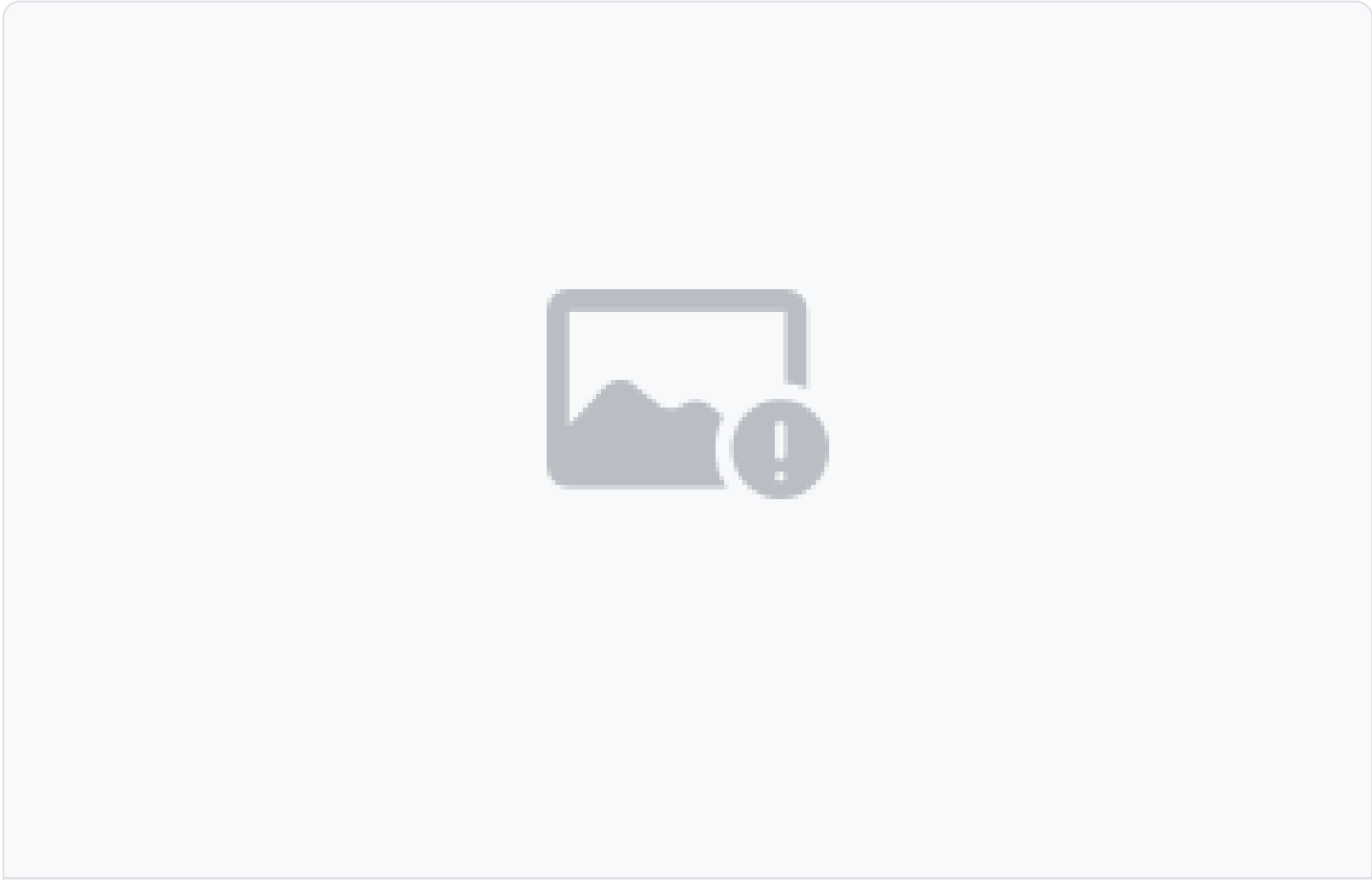


3.UnboundLocalError: local variable 'head_box' referenced before assignment

在代码的第 483-488 行，有一个 try-except 块，如果 head_box 提取失败，它会被跳过，但然后代码仍然尝试在第 489 行使用 head_box 变量。这会导致 UnboundLocalError ，因为如果异常发生， head_box 变量就不会被定义。

解决：修改coco文件

```python
        try:
            head_box = [obj["head_bbox"] for obj in anno_gaze]
            head_box = torch.as_tensor(head_box, dtype=torch.float32).reshape(4)
        except:
            # 如果没有head_bbox，使用默认的head box（整个图像）
            head_box = torch.as_tensor([0, 0, w, h], dtype=torch.float32).reshape(4)
```



coco.py

4.AttributeError: 'Linear' object has no attribute 'total_ops'

这个错误与 thop 库有关，在尝试计算操作数时出现了问题。在engine.py第 23 行，代码使用 thop.profile 来计算 FLOPS，但 thop 库与模型中的某些层不兼容。注释掉这一行（82），因为它不是训练所必需的.

解决：

```
with torch.cuda.amp.autocast(enabled=args.amp):
    if need_tgt_for_training:
        # enter network #
        # flops, params = profile(model, inputs=(samples, faces, head_channels, targets, head_boxes))
    else:
        outputs, gaze_outputs, gaze_cone = model(samples, faces, head_channels)

    loss_dict = criterion(outputs, targets)
    weight_dict = criterion.weight_dict
```

Training & Inference

To carry out experiments on the GOO dataset, please follow these commands:

Experiments on GOO-Synth:

代码块

```
1   bash scripts/TransGOP-R_train.sh /Dateses/goosynth/
```

Experiments on GOO-Real:

代码块

```
cd /root/autodl-tmp/TransGOP-R
bash scripts/TransGOP-R_train.sh
```

auc	dist	ang
0.7093690	0.2271640	51.35272536998765
0.7786373	0.1887684	37.838435667418466
0.8075693	0.1705246	32.31180879578748
0.8303673	0.1538598	29.467184912543047
0.8381767	0.1540835	27.63683831571677
0.8424807	0.1388124	25.217450158410433
0.8528880	0.1376710	24.608612683343836
0.8611818	0.1306325	22.43652242529776
0.8727574	0.1301469	23.671387077917466
0.8778564	0.1250509	22.973680820800993
0.8853718	0.1194948	20.939172433309643
0.8890077	0.1225657	22.72101710206762
0.9362334	0.1205975	21.886516853807727
0.9359310	0.1203366	22.072035136605933
0.9400129	0.1153691	21.02236467193879
0.9331285	0.1189732	22.970578448786064
0.9370974	0.1133061	20.815312729590918
0.9396606	0.1094351	20.152287237612963
0.9381695	0.1105360	20.36789477895278
0.9444854	0.1082339	19.725600831052503
0.9404804	0.1106576	20.375231729254853
0.9451776	0.1056983	18.639594470972682
0.9401667	0.1155066	21.213599669847333
0.9423734	0.1075984	19.090657736669883
0.9433676	0.1058612	19.715014531866526
0.9403722	0.1056052	19.42298609156187
0.9352853	0.1074552	19.947174062471312
0.9416465	0.1067917	19.252558955135633
0.9427477	0.1043040	18.205238194516937
0.9413786	0.1069369	18.7031485282932
0.9428356	0.1040047	18.837376717682552
0.9403128	0.1057118	20.31055612299156
0.9435965	0.1014069	18.583641080794333
0.9428091	0.1055560	19.22034045453975
0.9379513	0.1054051	19.21110999169612
0.9434226	0.1015534	18.64695524709366
0.9417993	0.1016330	18.478531480805263
0.9393025	0.0993609	18.184064138148642
0.9408896	0.1015820	18.428454546029837
0.9420187	0.1068971	19.686078835912888
0.9403584	0.1073223	18.844498851891125
0.9442317	0.1032168	18.657790633304412

0.9412673	0.0991806	17.73465116258971				
0.9441491	0.1010508	18.652864993792626				
0.9428409	0.1008798	18.789697520350995				
0.9425795	0.1014252	18.333705948189234				
0.9426316	0.1006322	18.585059723802637				
0.9428468	0.1009889	18.333688781052086				
0.9442735	0.0993334	17.68685800321504				
0.9426529	0.0978190	17.61617660191997				
0.9401770	0.1001539	18.44602948083909				

R50-MS4score.csv

Get_Result

Test on the GOO-Synth:

```
代码块
1   bash scripts/TransGOP-R_eval.sh /Dateses/goosynth/ /path/to/your/checkpoint
```

Test on the GOO-Real:

```
代码块
1   bash scripts/TransGOP-R_eval.sh
```

分隔符: ,

	0.9401770768071108	0.10015395025764845	18.44602948083909

评估分析

关键指标解读：

- `class_error: 0.00` - 分类错误率为0，表现很好
- `loss_ce: 0.0082 (0.0079)` - 交叉熵损失，数值较低表示分类效果良好
- `loss_bbox: 0.0132 (0.0150)` - 边界框回归损失
- `loss_giou: 0.1052 (0.1190)` - GIoU损失，用于边界框优化
- `loss_gaze: 11.7614 (16.6662)` - 注视点预测损失
- `loss_energy: 2.0594 (2.4052)` - 能量损失

模型表现：

- 各种损失值都保持在合理范围内
- 注视点预测损失(loss_gaze)在逐渐降低
- 模型在测试集上表现稳定

评估已经完成超过60% (570/904)，预计还有约45秒完成。从损失值来看，模型在目标检测和注视点预测任务上都有不错的表现。