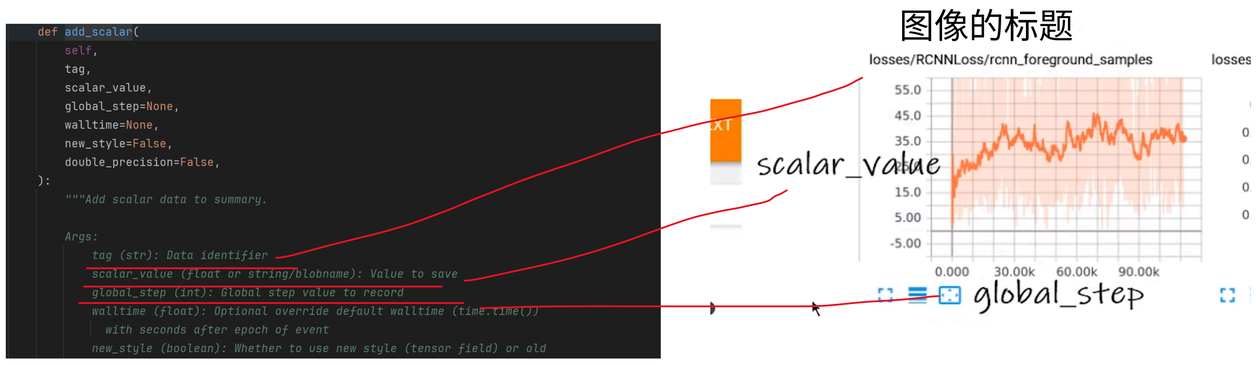
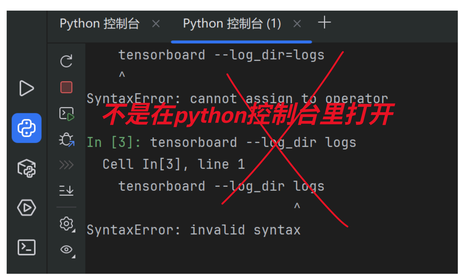
tensorboard

主要用于模型的训练中，可以可视化模型的训练过程，帮助我们更好得去调试和分析模型。



数据存放的路径　　在指定的端口打开

```
(pytorch) PS D:\pycharm_space\PycharmProjects\PythonProject1>  tensorboard  --logdir tensorboard/logs  --port 6008
D:\anaconda3\envs\pytorch\lib\site-packages\tensorboard\default.py:30: UserWarning: pkg_resources is deprecated as an API. See ht
pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to Setuptools<81.
  import pkg_resources
TensorFlow installation not found - running with reduced feature set.
Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.20.0 at http://localhost:6008/ (Press CTRL+C to quit)
W1114 15:19:29.200887 146132 application.py:559] path /hybridaction/zybTrackerStatisticsAction not found, sending 404
W1114 15:19:29.216076 129576 application.py:559] path /hybridaction/zybTrackerStatisticsAction not found, sending 404
W1114 15:41:13.669266 144112 application.py:559] path /hybridaction/zybTrackerStatisticsAction not found, sending 404
W1114 15:41:13.670267 124960 application.py:559] path /hybridaction/zybTrackerStatisticsAction not found, sending 404
W1114 15:42:20.933279 144852 application.py:559] path /hybridaction/zybTrackerStatisticsAction not found, sending 404
W1114 15:42:20.953834 143336 application.py:559] path /hybridaction/zybTrackerStatisticsAction not found, sending 404
```



绝对路径时反斜杠"\"
相对路径是正斜杠"/"
反斜杠会被当做转义

加个r防止转义

中间直接用空格就行了，用等号也行，但是等号前后就不能加空格了

```python
from torch.utils.tensorboard import SummaryWriter
import cv2

writer = SummaryWriter("logs")#对SummaryWriter创建实例  文件存储在logs文件夹下
image_path = r"D:\pycharm_space\PycharmProjects\PythonProject1\dataset\train\bees\21399619_3e61e5bb6f.jpg"
img = cv2.imread(image_path,cv2.IMREAD_COLOR)
#常用的两个方法
print(type(img))
print(img.shape)
writer.add_images( tag: "test",img, global_step: 2,dataformats="HWC")
for i in range(100):
    writer.add_scalar( tag: 'y = 3x',3*i,i)

writer.close()
```

根据add_image描述文件排查错误

是 NCHW 格式（4 维：批量数 × 通道 × 高度 × 宽度），两者维度长度不匹配（3 维 vs 4 维）导致断言失败。

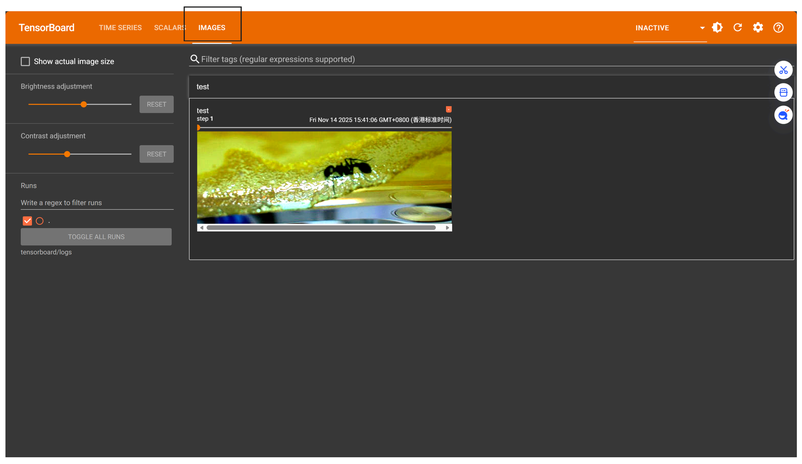
最直接的解决方法：明确指定数据格式

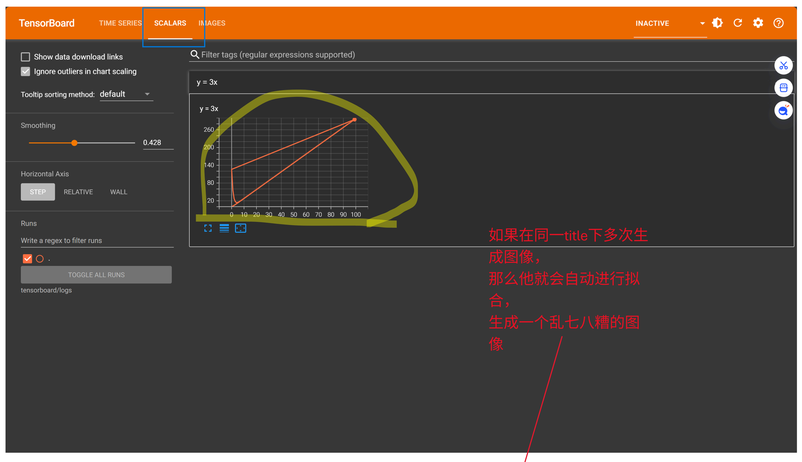
既然你的图像是 `(H, W, C)` 格式的 numpy 数组，只需在 add_images 中用 dataformats 参数明确告诉 TensorBoard 当前格式，无需复杂转换：

```python
import cv2
from torch.utils.tensorboard import SummaryWriter

# 1. 读取图像（确保路径正确，此处假设已读取成功）
img = cv2.imread("dataset/train/ants/5650366_e22b7e1065.jpg")  # 形状为 (H, W, 3)，BGR格式

# 2. 转换通道为RGB（否则显示颜色异常）
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  # 仍为 (H, W, 3)

# 3. 写入TensorBoard，明确指定数据格式为"HWC"
writer = SummaryWriter("logs")
# 关键：添加 dataformats='HWC'，告诉函数为前维度是（高度，宽度，通道）
writer.add_images("test", img_rgb, global_step=1, dataformats='HWC')
writer.close()
```

为什么这样能解决？

- `add_images` 函数通过 `dataformats` 参数识别输入维度，默认是 `NCHW`（4 维），但你的图像是 `HWC`（3 维）。
- 显式指定 `dataformats='HWC'` 后，函数会按（高度，宽度，通道）解析你的 3 维数组，无需额外增加批量维度（单张图像会自动处理）。





把存放数据的文件夹删除（logs）

control
+c关闭终端的运行
再重新打开 网页
（直接刷新时不行的）