

# 常用函数和方法

## Tensorboard: 可视化工具

**SummaryWriter 类:** 训练日志写入

```
writer = SummaryWriter("logs") # 指定日志目录
```

**参数:** log\_dir 为日志目录名 (输出结果的文件夹名)

启动 TensorBoard 的指令: tensorboard--logdir= "目录名"

**add\_image 方法:** 将图像数据添加到 TensorBoard 的日志目录中

```
writer.add_image("test", img_array, 1, dataformats="HWC")
```

```
def add_image(  
    self, tag, img_tensor, global_step=None, walltime=None, dataformats="CHW"  
):
```

**参数:** tag 为图片名称; img\_tensor 为数据类型, 格式只能为 tensor (pytorch) 或 array (numpy) ; global\_step 为训练次数; walltime 为现实训练时间; dataformats 为图像的数据格式

**add\_scalar 方法:** 记录数据, 常记录 loss 值

```
writer.add_scalar("y=3x", 3 * i, i)
```

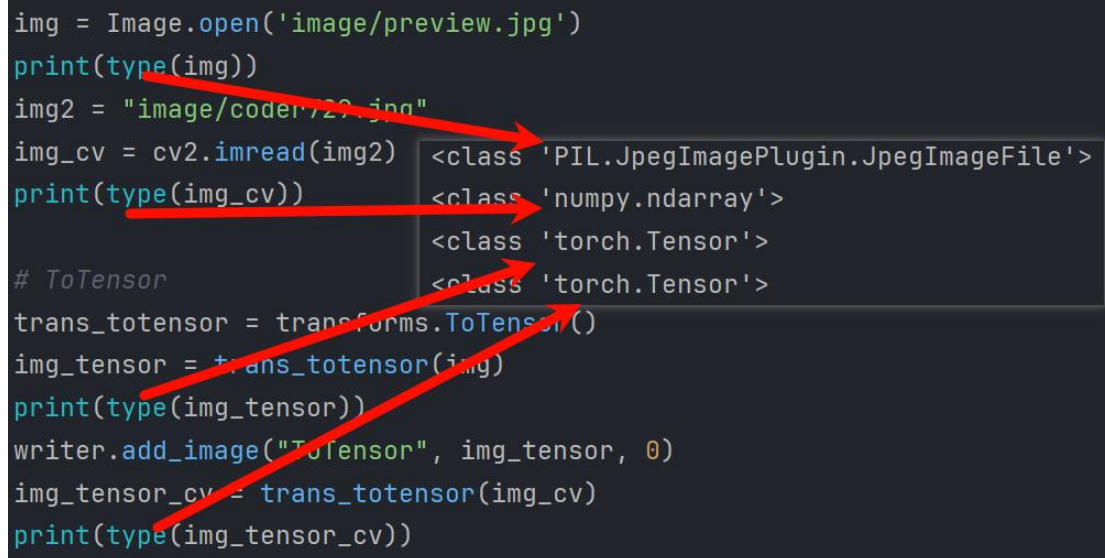
```
def add_scalar(  
    self, tag, scalar_value, global_step=None, walltime=None, new_style=False,
```

**参数:** tag 为图片名称; value 为数值, step 为步骤

# Transforms: 可视化工具

**Totensor:** 将 PIL 或 numpy 格式图像转换为 tensor 型格式 (opencv 创造的图形都是 numpy 格式)

```
img = Image.open('image/preview.jpg')
print(type(img))
img2 = "image/coder/22.jpg"
img_cv = cv2.imread(img2)
print(type(img_cv))
# ToTensor
trans_totensor = transforms.ToTensor()
img_tensor = trans_totensor(img)
print(type(img_tensor))
writer.add_image("ToTensor", img_tensor, 0)
img_tensor_cv = trans_totensor(img_cv)
print(type(img_tensor_cv))
```



**Normalize:** 将数据归一化 (归一化可提高训练速度)

```
class Normalize(torch.nn.Module):
    """Normalize a tensor image with mean and standard deviation.
    This transform does not support PIL Image.
```

**参数:** 需要传入 torch 格式, 不支持 PIL

**Resize:** 重设图片大小

**Compose:** 将多个操作组合成一个单一操作, 传入数据需是列表

**RandomCrop:** 随机裁剪

**Torchvision\_Dataset:** Torchvision 是一个 PyTorch 的附加库，它专门用于处理计算机视觉任务。torchvision 提供了一系列工具和功能，方便用户在图像数据集上进行操作，并简化深度学习模型的开发和应用。torchvision 包含四个主要模块：datasets、transforms、models 和 utils，每个模块都有其特定的功能。

```
# CIFAR-10 是一个更接近普通物体的彩色图像的小型数据集。  
train_set = torchvision.datasets.CIFAR10(root='./dataset', train=True, transform=dataset_transform, download=True)  
test_set = torchvision.datasets.CIFAR10(root='./dataset', train=False, transform=dataset_transform, download=True)
```

**参数：** root 为下载路径；train 为判断是否为训练数据集参数设置；transform 为图像需要进行的变换操作，一般使用 compose 把所需的 transforms 结合起来。