数据集获取技术文档

导入所需模块

导入用于处理文件和图像的必要模块 `os` ， `Dataset` 和 `Image` 。

```
import os  # 导入操作系统模块，用于处理文件路径
from torch.utils.data import Dataset  # 从 PyTorch 导入 Dataset 基类
from PIL import Image  # 导入PIL库中的Image模块，用于图像处理
```

定义数据集类

定义一个名为 `MyDataset` 的类，继承自 `torch.utils.data.Dataset` 。

```
def __init__(self, root_dir: str, label_dir: str, transform=None):
```

参数说明

`root_dir` (str): 包含图像子目录的根目录路径。 `label_dir` (str): 指定的子目录，包含相应的图像。
`transform` (callable, optional): 用于对图像进行处理的函数或变换。

初始化过程

```
self.root_dir = root_dir  # 设置根目录
self.label_dir = label_dir  # 设置标签目录
self.path = os.path.join(self.root_dir, self.label_dir)  # 组合根目录和标签目录以形成完整
self.image_path = os.listdir(self.path)  # 列出标签目录中的所有图像文件名
self.transform = transform  # 设置图像变换
```

`self.path` ： 通过 `os.path.join` 合并 `root_dir` 和 `label_dir` ，形成完整的路径。

`self.image_path` ： 使用 `os.listdir` 列出指定目录中的所有图像文件名，并保存到列表中。

获取数据项

定义 `__getitem__` 方法，根据索引获取图像及其对应标签。使用索引从 `self.image_path` 获取图像文件名，构造完整的图像路径，再使用 `Image.open` 打开图像文件，如果有指定的变换（如图像缩放、归一化等），则对图像进行处理，根据 `label_dir` 赋予图像相应的标签（0、1 或 2），返回处理后的图像和其对应的标签。

```
def __getitem__(self, idx: int):
    image_name = self.image_path[idx]  # 获取指定索引的图像文件名
    image_item_path = os.path.join(self.path, image_name)  # 构造完整的图像路径
    image = Image.open(image_item_path)  # 使用PIL打开图像
    if self.transform is not None:  # 如果提供了变换
        image = self.transform(image)  # 对图像应用变换

    if self.label_dir in ["Fake", "fake"]:
        label = 0  # "Fake" 图像对应标签 0
    elif self.label_dir in ["Real", "real"]:
        label = 1  # "Real" 图像对应标签 1
    else:
        label = 2  # 其他情况，默认标签为 2
    return image, label  # 返回图像和标签
```

获取数据集大小

定义 `__len__` 方法，返回数据集中图像的总数量。使用 `len()` 函数获取并返回 `self.image_path` 列表的长度，表示数据集中图像的数量。

```
def __len__(self):
    return len(self.image_path)  # 返回图像文件名列表的长度
```