from osgeo import gdal

import numpy as np

import os

def process\_image(input\_path, output\_path):

    # 打开输入图像

    dataset = gdal.Open(input\_path, gdal.GA\_ReadOnly)

    if dataset is None:

        print(f"无法打开图像: {input\_path}")

        return

    # 获取图像的宽度、高度和波段数

    width = dataset.RasterXSize

    height = dataset.RasterYSize

    bands = dataset.RasterCount

    # 确保图像有3个通道

    if bands != 3:

        print(f"图像通道数不符合要求，需要3通道，实际为{bands}通道")

        return

    # 获取数据类型

    band1 = dataset.GetRasterBand(1)

    gdal\_datatype = band1.DataType

    numpy\_datatype = gdal.GetDataTypeName(gdal\_datatype)

    # 确定输出数据类型

    if numpy\_datatype in ['Byte', 'UInt16', 'Int16', 'UInt32', 'Int32']:

        is\_integer = True

        output\_datatype = gdal.GDT\_Float32  # 转换为浮点型以保留小数

    else:

        is\_integer = False

        output\_datatype = gdal\_datatype

    # 读取三个通道的数据

    r\_band = dataset.GetRasterBand(1).ReadAsArray()  # R通道

    g\_band = dataset.GetRasterBand(2).ReadAsArray()  # G通道

    b\_band = dataset.GetRasterBand(3).ReadAsArray()  # B通道

    # 如果是整数类型，转换为浮点数

    if is\_integer:

        r\_band = r\_band.astype(np.float32)

    # 创建新的G、B通道，全部设为0，保持相同数据类型

    new\_g\_band = np.zeros\_like(g\_band, dtype=r\_band.dtype)

    new\_b\_band = np.zeros\_like(b\_band, dtype=r\_band.dtype)

    # 创建输出图像

    driver = gdal.GetDriverByName('GTiff')

    output\_dataset = driver.Create(

        output\_path, width, height, 3, output\_datatype

    )

    # 设置地理参考信息和投影信息

    output\_dataset.SetGeoTransform(dataset.GetGeoTransform())

    output\_dataset.SetProjection(dataset.GetProjection())

    # 写入三个通道的数据

    output\_dataset.GetRasterBand(1).WriteArray(r\_band)  # R通道保持不变

    output\_dataset.GetRasterBand(2).WriteArray(new\_g\_band)  # G通道全部为0

    output\_dataset.GetRasterBand(3).WriteArray(new\_b\_band)  # B通道全部为0

    # 关闭数据集，确保数据写入磁盘

    output\_dataset = None

    dataset = None

    print(f"图像处理完成，输出路径: {output\_path}")

def process\_folder(input\_folder, output\_folder):

    # 确保输出文件夹存在

    if not os.path.exists(output\_folder):

        os.makedirs(output\_folder)

    # 遍历输入文件夹中的所有文件

    for filename in os.listdir(input\_folder):

        # 构建完整的文件路径

        input\_path = os.path.join(input\_folder, filename)

        # 跳过子文件夹，只处理文件

        if not os.path.isfile(input\_path):

            continue

        # 检查文件扩展名，只处理常见的图像格式

        ext = os.path.splitext(filename)[1].lower()

        if ext not in ['.tif', '.tiff']:

            print(f"跳过非tif文件: {filename}")

            continue

        # 构建输出文件路径

        output\_filename = f"{os.path.splitext(filename)[0]}\_R\_only.tif"

        output\_path = os.path.join(output\_folder, output\_filename)

        # 处理图像

        print(f"正在处理: {filename}")

        process\_image(input\_path, output\_path)

# 使用示例

input\_folder\_path = r"D:\one\deep\_learning\snerf-main\data\0681"  # 替换为你的输入文件夹路径

output\_folder\_path = r"D:\one\deep\_learning\snerf-main\data\0681"  # 替换为你的输出文件夹路径

process\_folder(input\_folder\_path, output\_folder\_path)