对hdf数据进行to tiff

# 1.hdf各波段数据的to tiff  
inputHdf\_path = (r"E:/zuoye/202106280700.hdf")  
with h5py.File(path\_hdf,'r') as hdf\_file:  
 extend = hdf\_file.attrs["exrend"]  
 band\_name =list(hdf\_file.keys())  
 #data = hdf\_file[band\_name[-1]][()]  
 for i in band\_name:  
 data = hdf\_file[i][()]  
 gtiff\_driver = gdal.GetDriverByName('GTiff')  
 name\_tif = i + '.tif'  
 path\_tif = os.path.jion(path\_tem,name\_tif)  
 path\_tiflist\_tem.append(path\_tif)  
 out \_ds = gtiff\_driver.Create(path\_tif,datd.shape[1],data.shape[0],1,gdal.GDT\_Float32)  
 x = (exrend[1] - extend[0] / data.shape[1])  
 y = - (extend[3] - extend[2] / data.shape[0])  
 geoTrans = [extend[0],a,0,extend[3],0,b]  
 sr = osr.SpatialReference()  
 sr.ImportFromEPSG()  
 out\_ds.SetProjection(sr.ExportTowkt())  
 out\_ds.SetGeoTransform(geoTrans)  
 out\_band = out\_ds.GetRasterBand(1)  
 out\_band.WriteArray(data)  
 del out \_ds  
return path\_tiflist\_tem

TIFF的重采样

outputfilePath = 'D:/studyprojects/gdal/GdalStudy/Files/images/202106280700.tif'

inputfilePath='D:/studyprojects/gdal/GdalStudy/Files/images/202106280700.tif'

referencefilefilePath='D:/studyprojects/gdal/GdalStudy/Files/images/202106280700.tif'

def ReprojectImages():

# 获取输出影像信息

inputrasfile = gdal.Open(inputfilePath, gdal.GA\_ReadOnly)

inputProj = inputrasfile.GetProjection()

# 获取参考影像信息

referencefile = gdal.Open(referencefilefilePath, gdal.GA\_ReadOnly)

referencefileProj = referencefile.GetProjection()

referencefileTrans = referencefile.GetGeoTransform()

bandreferencefile = referencefile.GetRasterBand(1)

Width= referencefile.RasterXSize

Height = referencefile.RasterYSize

nbands = referencefile.RasterCount

# 创建重采样输出文件（设置投影及六参数）

driver = gdal.GetDriverByName('GTiff')

output = driver.Create(outputfilePath, Width,Height, nbands, bandreferencefile.DataType)

output.SetGeoTransform(referencefileTrans)

output.SetProjection(referencefileProj)

# 参数说明 输入数据集、输出文件、输入投影、参考投影、重采样方法(最邻近内插\双线性内插\三次卷积等)、回调函数

gdal.ReprojectImage(inputrasfile, output, inputProj, referencefileProj, gdalconst.GRA\_Bilinear,0.0,0.0,)

基于shp的裁剪

import rasterio as rio band = rio.open(path)

from rasterio.warp import (reproject,RESAMPLING, transform\_bounds,calculate\_default\_transform as calcdt) affine, width, height = calcdt(src.crs, dst\_crs, src.width, src.height, \*src.bounds) kwargs = src.meta.copy() kwargs.update({ 'crs': dst\_crs, 'transform': affine, 'affine': affine, 'width': width, 'height': height, 'geotransform':(0,1,0,0,0,-1) , 'driver': 'GTiff' }) dst = rio.open(newtiffname, 'w', \*\*kwargs) for i in range(1, src.count + 1): reproject( source = rio.band(src, i), destination = rio.band(dst, i), src\_transform = src.affine, src\_crs = src.crs, dst\_transform = affine, dst\_crs = dst\_crs, dst\_nodata = src.nodata, resampling = RESAMPLING.bilinear)

from geopandas import GeoSeries features = [shpdata.geometry.\_\_geo\_interface\_\_]

from geopandas import GeoSeries features = [GeoSeries(shpdata.geometry[i]).\_\_geo\_interface\_\_]

import rasterio.mask out\_image, out\_transform = rio.mask.mask(src, features, crop=True, nodata=src.nodata) out\_meta = src.meta.copy() out\_meta.update({"driver": "GTiff", "height": out\_image.shape[1], "width": out\_image.shape[2], "transform": out\_transform}) band\_mask = rasterio.open(newtiffname, "w", \*\*out\_meta) band\_mask.write(out\_image)