

Using Python to Realize Automatic Generation of Geographic Information Data into Graph and Text Table Documents

Ligong Huang Xiangwen Chen Jing Ma Hong Chang

Yunnan Remote Sensing Center, Kunming, Yunnan, 650034, China

Abstract

Python is a kind of simple and easy to learn interpretation of high-level language, according to the need it can both process and object oriented, and do not need to care about the underlying details such as memory allocation, and has the characteristics of across platform and portability, compared with other languages is simple and easy to implement, can let the programmer will focus on the transaction logic rather than the language data structure and some of the underlying algorithm. Due to its versatility and universality, it has been integrated into multiple application platforms. Here is a language application example that realizes the joint use of Osgo, gdal, ogr, matplotlib, openCV, Python-docx, realizes the drawing and annotation of local images and map spots, and automatically generates document tables with the drawing results.

Keywords

Python; geographic information; automation; graph and text table

运用 Python 实现地理信息数据自动化生成图文表格文档

黄立功 陈香稳 马靖 常虹

云南省遥感中心, 中国·云南昆明 650034

摘要

Python是一种简单易学的解释型高级语言, 根据需要它既能面向过程也能面向对象, 又不需要关心诸如内存分配之类的底层细节, 又有跨平台和可移植性的特点, 与其他语言相比简单且容易实现, 可让编程人员将重点放在需要实现的事务逻辑上而不是语言的数据结构和一些底层的算法上。由于其多功能性和广泛性, 已融入多种应用平台。在此介绍一个通过 Osgo、gdal、ogr、matplotlib、openCV、Python-docx 等开源库的联合使用, 实现脱离 ArcGIS 软件读取栅格、地理信息数据库并实现局部影像和图斑的绘制、标注, 并用绘制结果自动生成文档表格的语言应用实例。

关键词

Python; 地理信息; 自动化; 图文表格

1 引言

Python 是一种应用广泛的程序设计语言, 最为关键的是, 它具有相当丰富的标准库和第三方开源库可以调用。这些特点使得编程人员可以事半功倍、节约时间、提高效率, 而其他语言则需要投入大量时间和精力才能完成。因此, Python 特别适合作为数据处理语言使用, 当面对一个特别繁琐需要投入大量人力和耗费大量时间去完成的重复性数据处理流程时, 编程人员应该尝试去掌握和使用这一门语言, 可以用它实现一些前期模型的验证、数据的预处理、后期总结时的数据统计和分析等, 论文主要介绍自动化批量生成图文表格方面的运用。

在地理信息领域, 专业人员经常会遇到需要解决一些通过调用地理信息数据库及遥感影像共同生成大量图文表格的问题。这些表格结构复杂、信息源多、数量庞大、时间受限, 如果采用传统方法制作, 需耗费大量时间和人力, 且难于保证按时按量完成。而这类工作往往也是机械性、重复性、一次性的, 如果使用 C++ 等类的高级语言去开发, 代码工作量以及调试复杂度也会让编程人员望而却步, 但通过 Python 调用相应的库, 用办公自动化的方式去实现自动化批量生成图文表格文档, 就成了实现一次性脚本程序的最佳选择。

2 关键模块主要功能

要实现以上多种功能, 完成从地理信息数据库读取数据、遥感影像的通道合并、影像和矢量数据的绘制、自动化表格批量生成等一系列流程, 就离不开 Python 的支撑。如果换作用其他语言去实现, 海量的代码和调试需耗费大量时

【作者简介】黄立功(1970-), 女, 中国云南昆明人, 本科, 高级工程师, 从事基础测绘、地理信息数据处理研究。

间和精力才能完成,但 Python 众多开源社区提供的第三方库,可以让专业人员轻松就完成自动化批量生成图文表格文档工作,以下就是一个实现地理信息数据的读写、绘制以及办公自动化表格生成等的案例中使用到的关键模块^[1]。

2.1 Osgeo

Osgeo 库是由开源空间信息基金会 (Open Source Geospatial Foundation, 简称 OSGeo) 支持发展的开源空间信息社区,推出在全球使用的开源代码库,通过不同平台与不同语言交流合作,目前有 C/C++、Python、C#、Java 版本的可调用库,本案例主要使用其中的 GDAL 和 OGR 的功能。

GDAL (Geospatial Data Abstraction Library) 全称为“地理空间抽象库”,是 X/MIT 协议下读写栅格数据的开源库,它用抽象数据模型表达各种文件格式,以方便使用者读写,具体说用它来读取作为背景的遥感影像文件。

OGR (OGR Simple Features Library) 其实是 GDAL 的一个分支,它主要提供对地理空间矢量数据及空间参考信息的读写支持及处理,具体说是用它来读取地理信息数据库中的图斑矢量数据及字段信息。

2.2 Matplotlib

Matplotlib 是为 Python 语言提供的一个用于数据绘图的开源模块库,可以用它绘制各种二维图形,简单几行代码就能绘制出高质量图形,极大地简化了编程人员的绘图操作,本案例中通过 Matplotlib 配合 OpenCV 进行遥感影像绘制,同时将矢量数据绘制在读取的遥感影像上。

2.3 OpenCV

OpenCV 是一个跨平台计算机视觉库,提供大量图像处理及机器学习软件库,它原本是一个 C/C++ 库,但提供 Python 接口,在本案例中是用来对影像进行通道合并,并提供合适的数据格式,并提供合适的数据组织格式让 Matplotlib 能绘制到屏幕上。

2.4 Python-docx

Python-docx 是一个用于创建、修改微软办公软件 doc 文档的 Python 开源库,可以提供开发者几乎全部的 Word 操作控制,在此案例中用其来生成表格,形成按格式要求的文档成果。

3 Python 在地理信息数据中的应用

第三次全国国土调查项目和时点更新项目在地理信息数据调查完成过程中就遇到此类处理需求,需要通过调用地理信息矢量数据库并以遥感影像为底,生成一序列图斑并标注特定字段后,以比图斑四至范围稍大的区域显示并截图,最后将该图片与其他字段一起嵌入一个特定要求的表格中,需要进行这样处理并生成的表格数量可能达到数千,如果采用人工操作,作业量大、耗时较长、工序繁复、容易出错,如果采用 Python 来实现自动化流程则可以事半功倍,解决人力不足、重复工序、无法在规定时限完成等问题,具体设

计流程如下:

- ①创建用于盛纳表格的 doc 文档,并保持可修改状态^[2]。
- ②读取地理信息数据库并遍历需要处理的图斑数据,包括需要嵌入表格的字段值、需要在图形上标注的属性值等。
- ③每取出一个图斑,读取该图斑的四至坐标,计算其外接矩形及缓冲区的大小。
- ④按照上一步得到的图斑信息,读取相应位置与大小的影像数据块各通道。
- ⑤合并读取到的影像通道后转成普通图片数据格式并进行绘制。
- ⑥按照影像的绘制比例缩放矢量并在影像上按需要的方式绘制矢量、标注字段,如果需要注记与矢量图斑互不相压盖,还需要采取特定算法进行一定规避。
- ⑦将屏幕绘制内容保存为临时图片文件,生成要求格式的表格插入①步骤所生成的文档中。
- ⑧将保存的临时图片文件插入上一步生成的表格特定位置,将第②步骤读取的字段信息填充到表格中。
- ⑨继续遍历下一个图斑,直到所有图斑处理完成,保存 doc 文档并退出。

整个过程中最重要的就是第 6 个步骤,但因为屏幕坐标系统与直角坐标系统不一致,影像地面分辨率与屏幕绘制之间的关系需要建立严格的数学模型和对应关系,才能做到让矢量数据严密套合到影像之上。

4 应用 Python 语言实现上述流程的关键代码

下面是实现以上流程的关键代码,通过调用 Python 下列三个函数就基本实现了第三次全国国土调查中的地理信息自动化批量生成图文表格,确保了三调地理信息数据成果按照国家要求时限提交。如:“三调”项目中的国家核查抽样图斑核实举证整改情况表、关联图斑修改记录表、国家反馈意见内业整改情况说明、国家督察情况说明、采用收集到的农村土地确权高分辨率正射影像辅助判别地类截图表等图文表格文档;“时点更新”项目中的初始调查成果种植属性修改情况说明表、农用地变未利用地时点更新更正为林地和其他草地截图等图文表格文档。本例以完成“三调”中的国家核查整改难到达举证图斑基本信息表为代表,将完成上述工作的关键代码列出。

4.1 表格文档生成

生成 doc 文档,插入一个空表格并处于可随时修改状态。

```
def Createdocx(filename,jpgname):
    doc2=docx.Document()# 创建一个 Document 对象
    doc2.add_paragraph('time')# 增加一个 paragraph
    # 插入有序列表,段落的前面会有序号 123
    doc2.add_paragraph('第一段有序列表',style='List
Number')
```

```

doc2.add_paragraph( '第二段顺序列表',style=' List
Number' )
doc2.add_paragraph( '第三段顺序列表',style=' List
Number' )
# 插入无序列表,段落的前面没有序号
doc2.add_paragraph( '第一段无序列表',style=' List
Bullet' )
doc2.add_paragraph( '第二段无序列表',style=' List
Bullet' )
doc2.add_paragraph( '第三段无序列表',style=' List
Bullet' )
# 插入一个6行6列的表格
table=doc2.add_table(rows=6,cols=6,style=' Table
Grid' )
for i in range(0,6):
    for j in range(0,6):
        table.cell(i,j).text=" 第 {i} 行 {j}
列" .format(i=i+1,j=j+1)
doc2.add_picture(jpgname,width=docx.shared.
Inches(5)) # 插入照片
doc2.save(filename)# 保存文档

```

4.2 影像、矢量读取及绘制

读取影像合并通道并绘制到一个临时图像的设备上下文中并插入当前生成的表格中。

```

def Imgsnap(imgname,shpfile,baspt,fbl,rect,FieldStr,jpg_
path,docfilename):
    np.set_printoptions(threshold=np.inf)
    ax=plt.gca() # 坐标轴对象
    ds1 = gdal.Open(imgname) # 打开影像
    cols=(ds1.RasterXSize) # 图像宽度
    rows=(ds1.RasterYSize)# 图像高度
    XX=(rect[0]-baspt[0])/fbl
    YY=rows-(rect[3]-baspt[1])/fbl
    img_wid=int((rect[2]-rect[0])/fbl)
    img_hei=int((rect[3]-rect[1])/fbl)
    plt.xlim(0,img_wid)
    plt.ylim(0,img_hei)
    band1 = ds1.GetRasterBand(1)# 取第1波段
    r=band1.ReadAsArray(XX,YY,img_wid,img_hei)
    band2 = ds1.GetRasterBand(2)# 取第2波段
    g=band2.ReadAsArray(XX,YY,img_wid,img_hei)
    band3 = ds1.GetRasterBand(3)# 取第3波段
    b=band1.ReadAsArray(XX,YY,img_wid,img_hei)
    img=cv2.merge([b,g,r])
    ax.invert_yaxis() # 颠倒y轴
    plt.imshow(img)

```

```

ds2=ogr.Open(shpfile,0)
lyr=ds2.GetLayer(0)
N=lyr.GetFeatureCount()
lyr.SetSpatialFilterRect(rect[0],rect[1],rect[2],rect[3])
oFeature = lyr.GetNextFeature()
while oFeature is not None:
    fd=oFeature.GetFieldAsString(FieldStr)
    geom=oFeature.geometry()
    ext=geom.GetEnvelope()
    if geom==None:
        oFeature = lyr.GetNextFeature()
        continue
    NN=geom.GetGeometryCount()
    anno_ed=False # 该对象是否已经注册,
每对象只注一次
    for i in range(0,NN):
        curve=geom.GetGeometryRef(i)
        coords=curve.GetPoints()
        if coords is not None:
            coords_new=[]
            for coord in coords:
                XX=(coord[0]-rect[0])/fbl
                YY=img_hei-
                (coord[1]-rect[1])/fbl
            coords_new.
            append((XX,YY))
            x,y=zip(*coords_new)
            plt.plot(x,y,' cyan' )
            MX=sum(x)/len(x)
            MY=sum(y)/len(y)
            if anno_ed is not True:
                plt.text(MX,M
                Y,fd,fontsize=10,color=' magenta' )
            anno_ed=True
        oFeature = lyr.GetNextFeature()
    plt.xticks([],plt.yticks([]) # 不显示坐标轴
    jpgname=jpg_path+str(uuid.uuid4()+".jpg"
    plt.savefig(jpgname) # 保存到临时文件
    Createdocx(docfilename,jpgname)
    del ds1
    ds2.Destroy()
    del ds2

```

4.3 遍历对象依次生成并插入表格

按条件查询数据库遍历满足条件的图斑数据并读取矢量、字段,调用上面两个模块实现整个业务流程。

```

def DocxSnapProcess(imgname,shpfile,basetp,fb1,FieldStr,
jpg_path,doc_path,condstr):
    ds=ogr.Open(shpfile,0)
    lyr=ds.GetLayer(0)
    N=lyr.GetFeatureCount()
    lyr.SetAttributeFilter(condstr)
    N1=lyr.GetFeatureCount()
    if N1==0:
        print(“当前查询条件为查询到对象.”)
        return
    oFeature = lyr.GetNextFeature()
    while oFeature is not None:
        fd=oFeature.GetFieldAsString(FieldStr)
        namestr=oFeature.GetFieldAsString
(“BSM”)
        geom=oFeature.geometry()
        ext=geom.GetEnvelope()
        wid=ext[1]-ext[0]
        hei=ext[3]-ext[2]
        rect=(ext[0]-wid*0.2,ext[2]-hei*0.2,ext[1]+
wid*0.2,ext[3]+hei*0.2)
        docfile=doc_path+namestr+” .docx”
        Imsgsnap(imgname,shpfile,basetp,fb1,rect,’
DLBM’,” d:\\temp\\”,docfile)
        oFeature = lyr.GetNextFeature()
    del ds
    print(“Fine”)
    
```

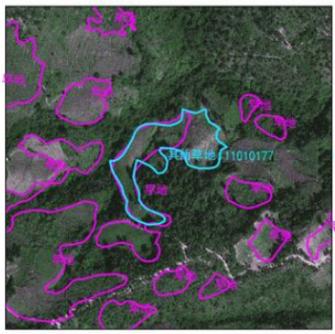
5 结语

在“三调”项目中，运行以上代码对地理信息数据库内外业难以到达的举证图斑，按照“国家核查难到达举证图斑”条件要求进行筛选处理，自动化生成“国家核查整改难以到达举证图斑基本信息表”，见表 1。

表 1 国家核查整改难以到达举证图斑基本信息表

国家核查整改难以到达举证图斑基本信息			
图斑标识码	核查结果	核查结果说明	地类名称
XXXXXXXX11010177XXX	二类	未修改，初始调查错误，属于督察要求排查图斑，按其他草地认定	其他草地

难以到达举证图斑影像截图



Python 语言语法版本之间差别是比较大的，比如 ArcGIS 中就内嵌了 Python2.7，但相对于 Python3.X 已经显得语法老旧，而且大多数比较有价值的第三方库都不再支持 Python 2.7 或者支持不充分^[3]，因此我们要通过 GDAL 库和 OGR 库摆脱对 ArcGIS 的依赖才是正确的开发方向，此例正是这样的应用，虽然这会增加开发难度，但充分利用 Python 庞大的生态圈中丰富而功能强大的开源库，能解决一些地理信息项目中的数据复杂问题和难题，提高工作技术水平和能力，不但提高地理信息数据处理效率、缩短地理信息数据处理时间，还能拓展地理信息数据成果的应用。

参考文献

- [1] 李民录.GDAL源码剖析与开发指南[M].北京:人民邮电出版社,2022.
- [2] Chris Garrard. Geoprocessing with Python[M].北京:人民邮电出版社,2017.
- [3] 鱼弦.Python分析地理空间数据[CP][OL].CSDN网站,2023.