

第 2 节 GEE 的图形用户界面

GEE 是一个主要依靠编码命令进行空间分析和操作的平台，但 GEE 也有适合界面操作的平台：GEE 的图形用户界面(Graphical User Interface, GUI)。GEE 的 GUI 的优点在于符合人的操作直觉，并且对新用户友好，缺点是功能较少，可以实现的空间分析有限。

GEE 的学习重点并不在 GUI 上，可以将本节作为一种过渡，一种从“点击操作”到“命令操作”的适应过程。通过本节学习的，我们能更加深入的体会利用 GEE 进行空间分析时，对任务进行分步运行的思路。

2.1 GUI 的构成

GEE 的 GUI(图 2.1)主要由数据(框 1)、计算(框 2)、分析(框 3)和显示(框 4)四部构成。其中计算和分析功能只有在申请 GEE 资格并且登陆以后才能完全使用。

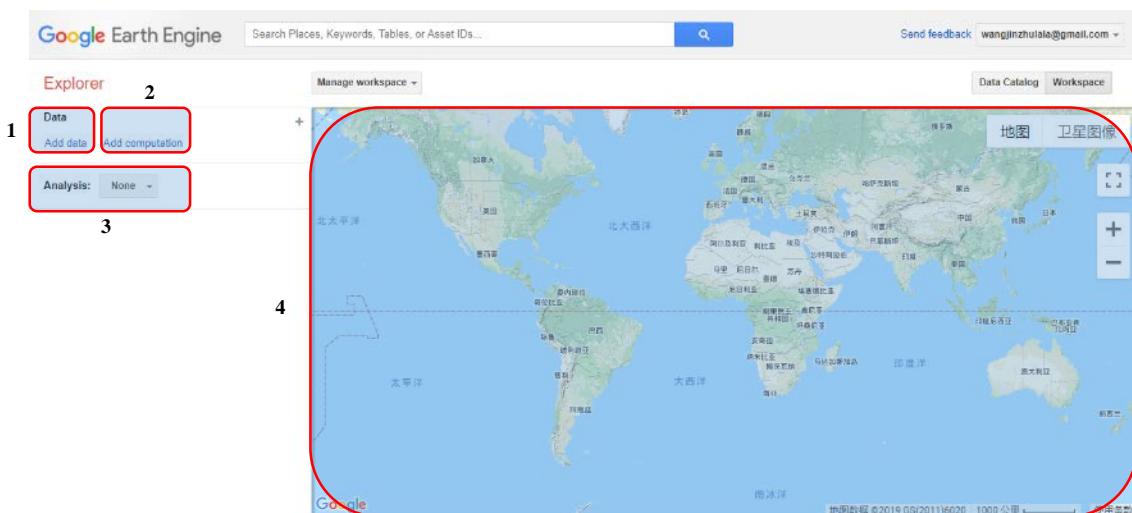


图 2.1 GUI 的基本构成

2.1.1 GUI 中的栅格数据

GUI 的数据(图 2.2)主要为卫星遥感图片、遥感衍生指数数据(8/32 天合成 NDVI/NDWI 数据)以及土地利用分类数据。

在 GEE 的 GUI 中能都调用的遥感卫星图片数据包括 Landsat 4/5/7/8 TOA 系列数据。其中 Landsat 5/7 能通过搜索框中的 Landsat TOA 选项相对简单的进行调用，而 Landsat 4/8 数据(图 2.3)需要在 Data Catalog 中搜索 Landsat 4/8 的 32-day Raw Composite 数据才能加入到 GUI 的工作区中。对于 Sentinel 系列遥感数据时，GUI 仅显示其数据介绍，而不能将其载入到 GUI 的工作环境中。

GUI 中的衍生合成数据(图 2.4)主要包括来自 MODIS，Landsat 4/5/7/8 等的 NDVI, NDWI, NDSI, EVI 以及 BAI 数据等。

GUI 中的土地利用数据(图 2.5)可以方便的加入到工作区中，点击搜索框，选择 CLASSIFIED RASTERS (例如 GlobCover2009)中的数据集，土地利用数据信息的加载框就会显示在工作区中。对每一种地理赋予分类值(Assign Class)，相应的地类就会出现在 GUI 工作区的底图上。

FEATURED
RASTERS
Landsat TOA Percentile Composite
CLASSIFIED RASTERS
MCD12Q1-1 IGBP
MCD12Q1-2 UMD
MCD12Q1-3 LAI/fPAR
MCD12Q1-4 NPP
MCD12Q1-5 PFT
GlobCover 2009
USDA NASS Cropland
VECTORS
Hand-drawn points and polygons
Fusion Table

图 2.2 GEE 的 GUI 主要数据类型

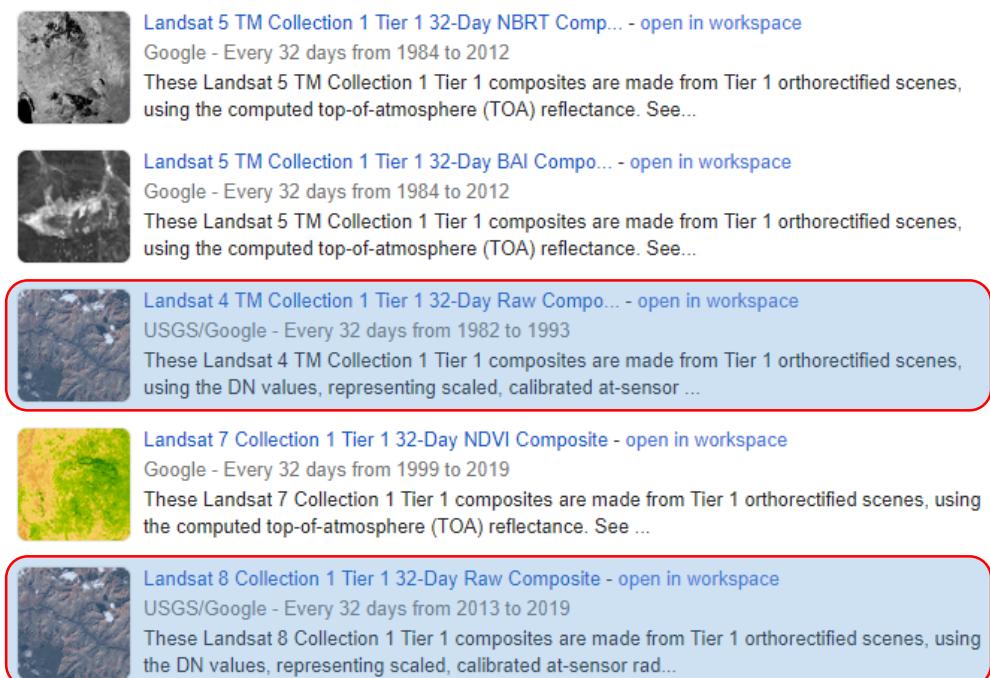


图 2.3 Landsat 4/8 的 32-day Raw Composite 数据

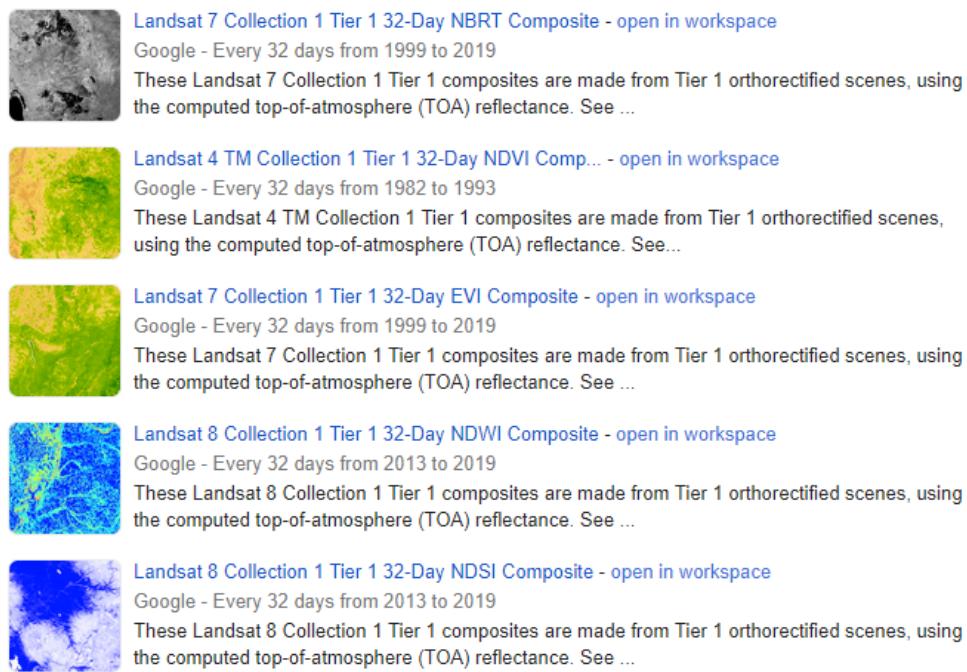


图 2.4 GUI 中的衍生合成数据

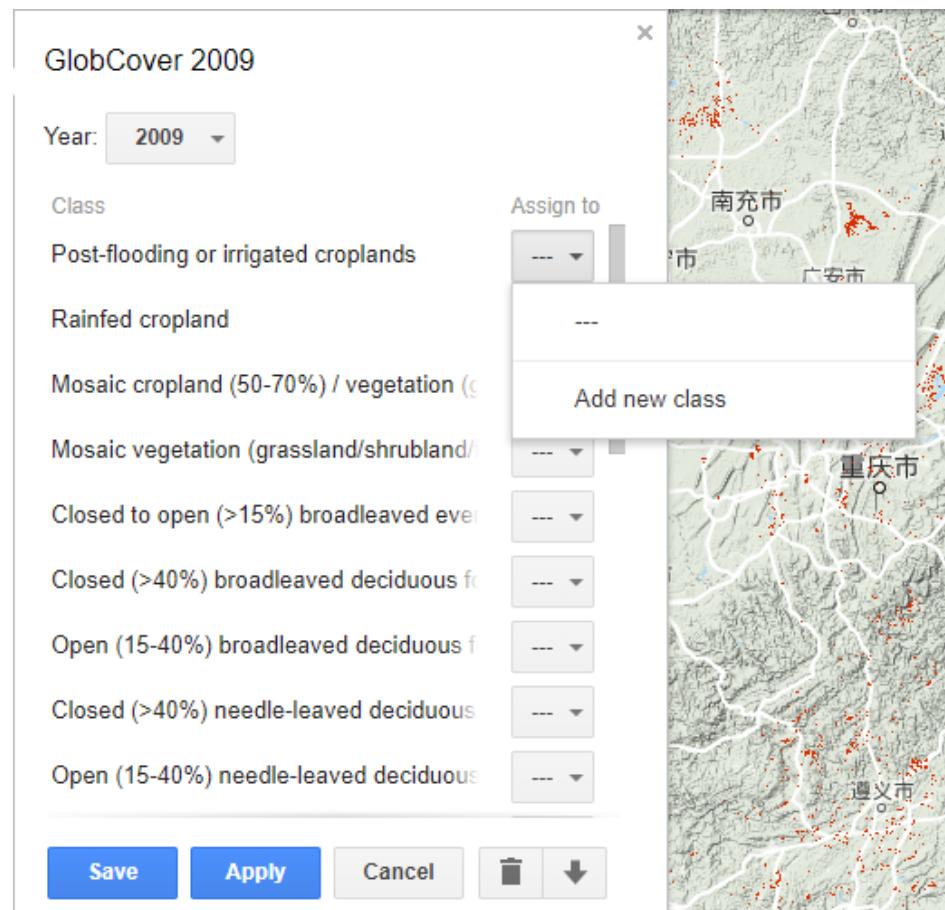


图 2.5 GUI 中的土地利用数据 (GlobCover2009)

在搜索框输入 Landsat 7，点击弹出的 Landsat TOA Percentile Composite 数据，该数据会直接加载到工作区，而点击 Landsat 7 Collection 1 Tier 1 TOA Reflectance 数据时，该数据不能被加载到工作区，但是会弹出数据的介绍界面。上述两种数据在搜索框中并没有明显的标识以区别哪一种数据是可以加载的，哪一种是不能被加载的。这说明 GEE 的 GUI 平台的用户逻辑并不完善，反映出 GUI 可能并未得到 GEE 开发团队的重视。

2.1.2 GUI 中的矢量数据

GUI 中的矢量数据的加载方式有两种，一种方式是通过 Fusion Table(图 2.6)进行加载，另一种方式是通过手绘进行加载。

Fusion Table 是谷歌公司推出的一种云存储服务，其主要目的是提供一种方便的文件共享平台。我们可以这样理解，当用户将表格(矢量文件是由点构成的，矢量点文件在存储上以类似于表格文件的组织方式进行存储)上传到 Fusion Table 上之后，谷歌会返回给用户一个数据 ID，用户凭借数据 ID 在任何谷歌服务框架中都可以快速的调用对应的文件。应该指出的是，Fusion Table 的运行模式被认为是云计算数据库的雏形，但由于计算机云计算技术的快速发展，谷歌公司决定在 2019 年 12 月 3 日以后关闭 Fusion Table 的服务。

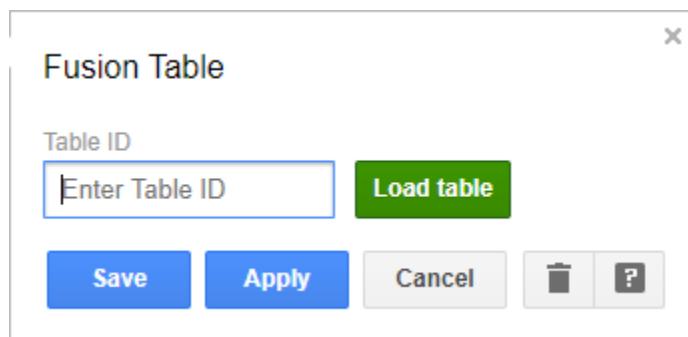


图 2.6 GUI 中的 Fusion Table 加载界面

利用手绘(图 2.7)进行矢量数据加载是 GUI 加载矢量的主要方式。首先在 Classes 栏目中添加标记类别，然后通过工作区左上角的绘图工具进行形状绘制即可得到手绘矢量数据。手绘矢量主要用于遥感图像的土地利用分类：通过选定图像区域进行遥感分类器的训练。



图 2.7 GUI 中通过手绘加载矢量图形

2.2 GUI 的运算与分析

2.2.1 GUI 的运算功能

在工作区加入数据后，利用 Add computation 工具可以对加入的数据进行进一步的处理。GUI 的计算类型(图 2.8)一共有 5 中，分别用于波段添加，掩膜处理，像素运算，邻域处理以及地形处理。

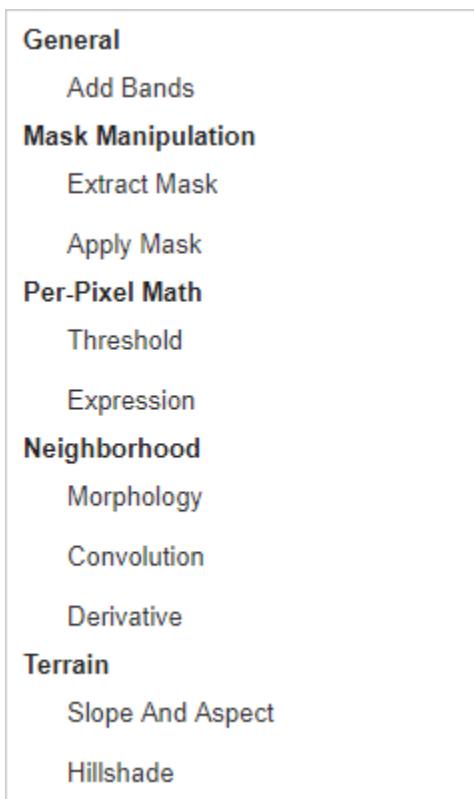


图 2.8 GUI 中的计算类型

波段添加的作用在于将若干栅格数据集合成复合数据，例如可以将 SLOPE 波段加到 DEM 波段上，可以得到一个包含 SLOPE 和 SEM 的数据集。掩膜处理的目的在于告诉 GUI 图像的哪些部分参与运算，哪些部分不参与运算。像素运算的目的在于挑选出合适的像素以及对像素进行数学处理，例如像素运算的阈值筛选可以筛选出高程小于 900m 的像素，而像素运算的表达式运算可以利用公式 $(\text{Band5}-\text{Band4})/(\text{Band5}+\text{Band4})$ 计算出 Landsat 8 系列图像的 NDVI 值。邻域运算可以对图像进行边界提取，卷积以及差分计算。地形计算可以计算出高程图像的坡度、坡向和山体阴影。

2.2.2 GUI 的分析功能

GEE 的 GUI 中的分析功能是完全服务于遥感图像分类的。GUI 分析功能由训练分类器，交叉验证和分类对比三个功能组成，其中训练分类器功能是 GUI 分析功能的核心，在点击训练分类器后，弹出的对话框(图 2.9)中包含了 10 中常见的遥感分类方法(分类器)。

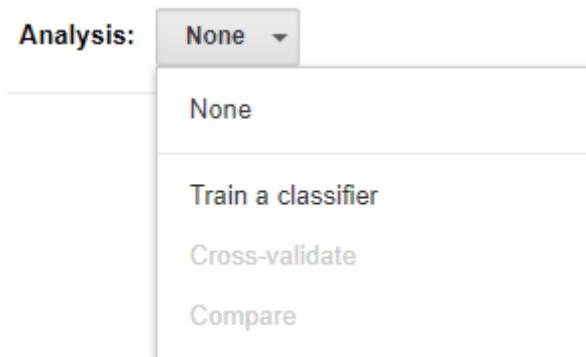


图 2.9 GUI 中的计算类型

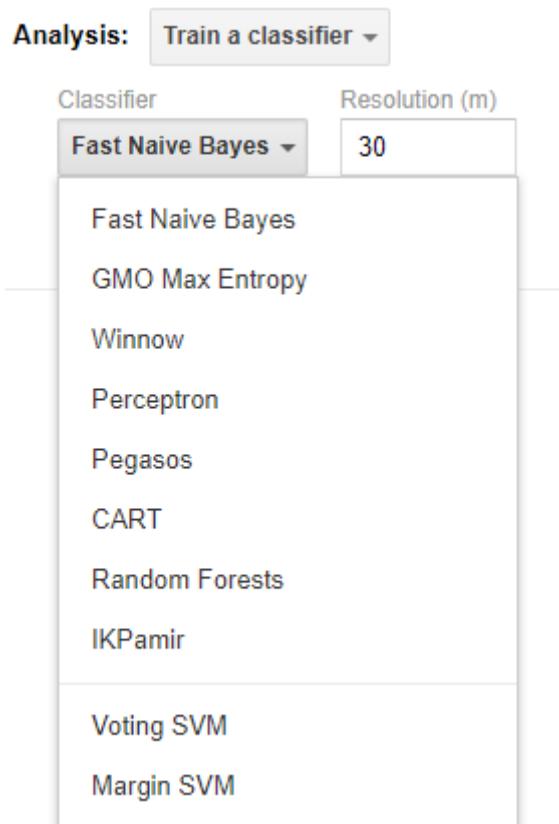


图 2.10 GUI 中的分析类型

2.3 利用 GUI 进行地形分析和遥感图像分类

GUI 与 GEE 的 Code Editor 相比可用数据和分析功能都相对较少，但 GUI 的操作逻辑简单，并且在操作的过程中可以让用户掌握在使用 Code Editor 时的分析思路。下边以地形分析和遥感图像分类为例进行解释。

2.3.1 利用 GUI 进行地形分析

本节的目的是提取高程大于 500m， 并且坡度小于 10°的地区。要达到这样的目标，相应的思路是：

- 1) 加入高程数据
- 2) 加入坡度数据
- 3) 提取高程大于 500m 的区域“DEM_提取”
- 4) 提取坡度小于 10°的区域“SLOPE_提取”
- 5) 利用工具将“DEM_提取”和 SLOPE_提取”叠加
- 6) 得到结果

GUI 中的操作步骤如下：

- 1) 在搜索框中输入“SRTM”，点击“SRTM Digital Elevation Data 30m”，然后点击弹出对话框最上部的“SRTM Digital Elevation Data 30m”并重命名为“DEM”后点击“Save”
- 2) 点击“Add computation”按钮，选择“Slope And Aspect”，然后点击弹出对话框最上部的“Slope And Aspect”并重命名为“SLOPE”，然后打开“Visualization”，将显示方式更改为“1 Band (Grayscale)”后点击“Save”
- 3) 点击“Add computation”按钮，选择“Threshold”，然后点击弹出对话框最上部的“Computed layer: Threshold”并重命名为“DEM_Reclass”，然后在 Threshold 栏目中将 Image 选择为“DEM”，同时通过 Add Threshold 按钮增加两个分类框，分别输入 0,500,0 和 500,9000,1。最后打开“Visualization”，将 Rang 更改为 0-1，然后点击“Save”
- 4) 点击“Add computation”按钮，选择“Threshold”，然后点击弹出对话框最上部的“Computed layer: Threshold”并重命名为“SLOPE_Reclass”，然后在 Threshold 栏目中将 Image 选择为“SLOPE”，注意点击 SLOPE 右侧的图层按钮，将 Aspect 前的√去掉，同时通过 Add Threshold 按钮增加两个分类框，分别输入 0,10,1 和 10,90,0。最后打开“Visualization”，将 Rang 更改为 0-1，然后点击“Save”
- 5) 点击“Add computation”按钮，选择“Expression”，然后点击弹出对话框最上部的“Computed layer: Expression”并重命名为“DEM_SLOPE_Intersect”，通过 Select Image 按钮分别选择“DEM_Reclass”和“SLOPE_Reclass”，保持默认名 img1 和 img2 不变，然后在 Expression 栏目中输入 img1 * img2。最后打开“Visualization”，将 Rang 更改为 0-1，然后点击“Save”
- 6) 观察结果，地图中白色区域就是符合要求的“高程大于 500m， 并且坡度小于 10°的地区”操作结果可以参考(<https://explorer.earthengine.google.com/#workspace/q6HdA8g7qPT>)。

在 Code Editor 中，同样也可以实现提取“高程大于 500m， 并且坡度小于 10°的地区”的目标。在编写代码命令之前，首先需要确定代码思路，可以将思路先以代码注释(图 2.10)的形式写入代码框中，然后再在相应的位置具体的进行命令编写。



The screenshot shows a 'New Script *' window in a Code Editor. The interface includes a toolbar with 'Get Link', 'Save', 'Run', 'Reset', and other icons. The code area contains the following pseudocode:

```
1 // 加入高程数据
2 // 加入坡度数据
3
4 // 提取高程大于500m的区域“DEM_提取”
5
6 // 提取坡度小于10°的区域“SLOPE_提取”
7
8 // 利用工具将“DEM_提取”和“SLOPE_提取”叠加
9
10 // 得到结果
```

图 2.11 首先进行思路构建

```
// 加入高程数据
var DEM = ee.Image("USGS/SRTMGL1_003");

// 加入坡度数据
var SLOPE = ee.Terrain.slope(DEM)

// 提取高程大于 500m 的区域“DEM_提取”
var DEM_Reclass = DEM.gt(500)

// 提取坡度小于 10°的区域“SLOPE_提取”
var SLOPE_Reclass = SLOPE.lt(10)

// 利用工具将“DEM_提取”和“SLOPE_提取”叠加
var DEM_SLOPE_Intersect = DEM_Reclass.and(SLOPE_Reclass)

// 得到结果
Map.setCenter(107.09,29.43,10)
Map.addLayer(DEM_SLOPE_Intersect)
```

在 Code Editor 中实现提取“高程大于 500m， 并且坡度小于 10°的地区”的方式与 GUI 中的稍有不同。在 GUI 中，筛选符合条件像素的方式是重分类，即设定一个区间，符合条件的像素被重新赋值(上例中通过 Add Threshold 按钮增加两个分类框，分别输入 0,500,0 和 500,9000,1 的目的就是将符合条件的像素重新赋值为 1，不符合条件的像素重新赋值为 0)，

而在 Code Editor 中，筛选符合条件像素的操作是“lt”和“gt”，它们的含义分别是“less than”和“greater than”，这样就比较容易理解相应代码的含义了。

可以看出，GUI 与 Code Editor 中的操作思路也是完全一致的。通过本例，可以看出命令操作的方式是“分布进行，逐步细化”的，这种思维方式对 GEE 的学习非常重要。

2.3.2 利用 GUI 进行遥感图像分类

本节的目的是利用 GUI 的 Analysis 功能实现遥感图像的的土地分类，目的是区分目标区域的建设用地，绿地和水体。相应的思路是：

- 1) 加载遥感图像
- 2) 确定训练样区
- 3) 训练分类器
- 4) 进行遥感解译
- 5) 得到结果

GUI 中的操作步骤如下：

- 1) 点击搜索框，选择“Landsat TOA Percentile Composite”，然后点击弹出对话框最上部的“Landsat TOA Percentile Composite”并重命名为“L5”，然后点击对话框中部的“Custom”将时间范围调整为 2010-01-01 至 2012-12-31，并将 Collection 选择为“Landsat 5”，最后打开 Visualization，将波段组合调整为“B4, B3, B2”的形式后，点击 Save。
- 2) 点击搜索框，选择“Hand-drawn points and polygons”，点击左侧的 Add Class 按钮，添加三个矢量种类，并分别命名为“Urban”，“Vegetation”和“Water”。然后在遥感图像的合适位置利用工作区左上角的 分别给每个种类添加相应的矢量训练区(图 2.11)。
- 3) 点击 Analysis，选择“Train a classifier”，选择“CART”作为训练函数。
- 4) 点击“Train classifier and display results”，进行训练并展示训练结果。具体操作可参考下边的连接 (<https://explorer.earthengine.google.com/#workspace/lWtI2Sy7sIf>)。

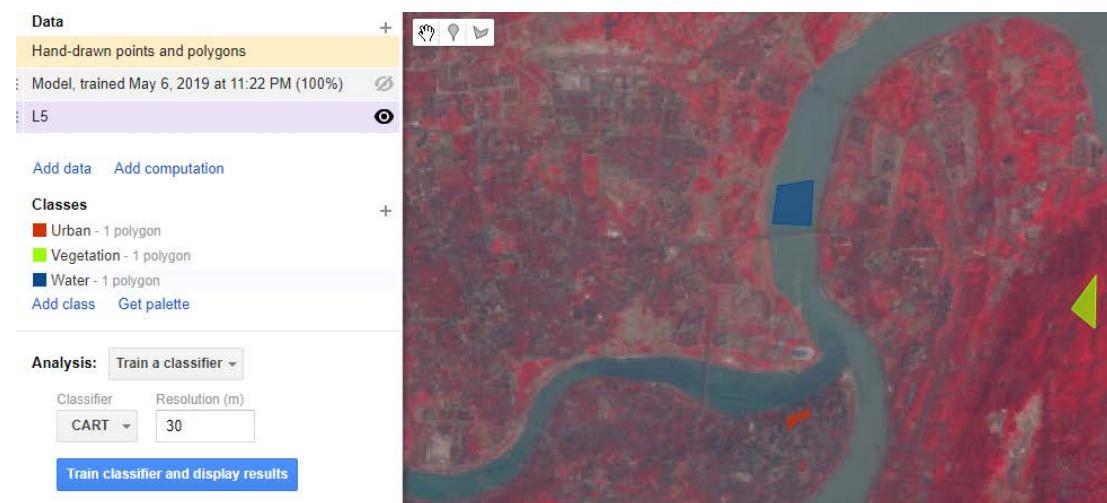


图 2.12 训练区选择

同样的，在 Code Editor 中也可以实现相应的操作，根据上个小节的方法，同样首先做出代码注释，然后分别对每个注释进行填充，具体代码如下：

```
// 定义默认值
var bands = ['B2', 'B3', 'B4', 'B5', 'B6', 'B7'];

// 加载遥感图像
var L5_Original = ee.ImageCollection("LANDSAT/LT05/C01/T1")

var L5_Filtered = L5_Original.filterDate('2010-01-01','2012-12-31')
    .filterBounds(ee.Geometry.Point(106.6229, 29.5723))

var L5_No_Cloud = ee.Algorithms.Landsat.simpleComposite(L5_Filtered).select(bands)

// 确定训练样区
var Urban = ee.Feature(ee.Geometry.Polygon(
    [[[106.5690, 29.5586],[106.5690, 29.5569],
      [106.5710, 29.5569],[106.5710, 29.5586]]], null, false),
    {"class": 1,"system:index": "0"}),
Water = ee.Feature(ee.Geometry.Polygon(
    [[[106.5849, 29.5593],[106.5849, 29.5571],
      [106.5877, 29.5571],[106.5877, 29.5593]]], null, false),
    {"class": 2,"system:index": "0"}),
Vegetation = ee.Feature(ee.Geometry.Polygon(
    [[[106.6217, 29.5704],[106.6217, 29.5684],
      [106.6243, 29.5684],[106.6243, 29.5704]]], null, false),
    {"class": 3,"system:index": "0"});

var sample_zone = ee.FeatureCollection([Urban,Water,Vegetation])

var training = L5_No_Cloud.sampleRegions({
  collection: sample_zone,
  properties: ['class'],
  scale: 30
});

// 训练分类器
var trained = ee.Classifier.cart().train(training, 'class', bands);

// 进行遥感解译
var classified = L5_No_Cloud.select(bands).classify(trained);

// 得到结果
Map.setCenter(106.61, 29.58, 13)
Map.addLayer(classified,{"min":1,"max":3,"palette":["ff4734","1e0fff","3cff0b"]});
```

本例相对 GUI 的操作显得复杂，这是因为在确定训练样区的代码中，有很大一部分是为了实现训练区的选取。在后边的学习中可以发现，训练样区的选取可以通过鼠标操作，因此，除去训练样区的选择，上述代码命令仍然是一行代码就能实现一个分步骤的功能。

2.4 小结

通过本节的学习，可以更加清晰的感受到“点击命令”与“代码命令”之间的关联，尤其是两者在进行空间分析时，都采用分步进行的思路特点。在具体实现功能上，“点击命令”用图形界面直观的引导用户进行操作(例如上述例子中，GUI通过点击 Add Computation 中的 Slope and Aspect 来计算地形要素)，而命令方式则用代码来实现同样的空间分析目的(例如在 Code Editor 中利用 ee.Terrain.slope() 来计算坡度)。

在充分理解 GEE 的数据构成以及命令思路后，就可以开始开始学习 GEE 中各种命令代码的具体语法和格式。从下一节开始，我们将开始 GEE 的编程学习，学习参数类型时，不能过分依赖前两节的基于直觉的学习思路，而应该更加注重细节的应用和体会，同时还要注意养成规范的代码编写习惯。



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#).

本材料由王金柱（西南大学&迪肯大学）创作。如有需要请与我联系。

This doc contributed by Jinzhu Wang of Southwest University & Deakin University.

Email: wangjinzhlala@gmail.com