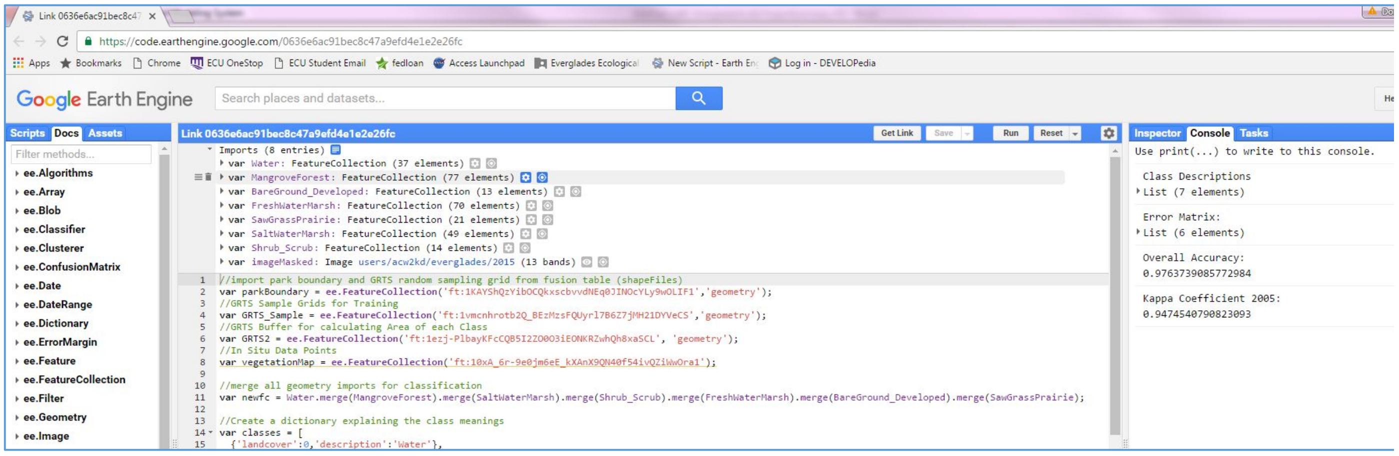
Google Applications for Transitioning Everglades Regions Tutorial

Sign up for GEE account - https://earthengine.google.com/

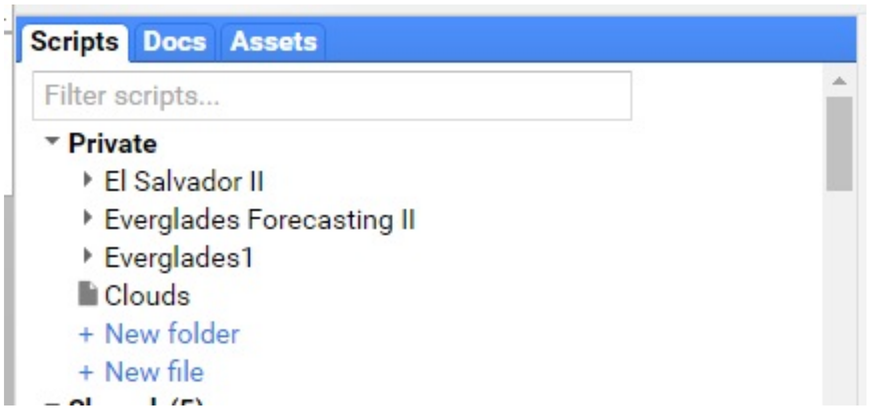
1. Click associated link (s) to this project (see ReadMe.txt). (This will take you to the code editor)



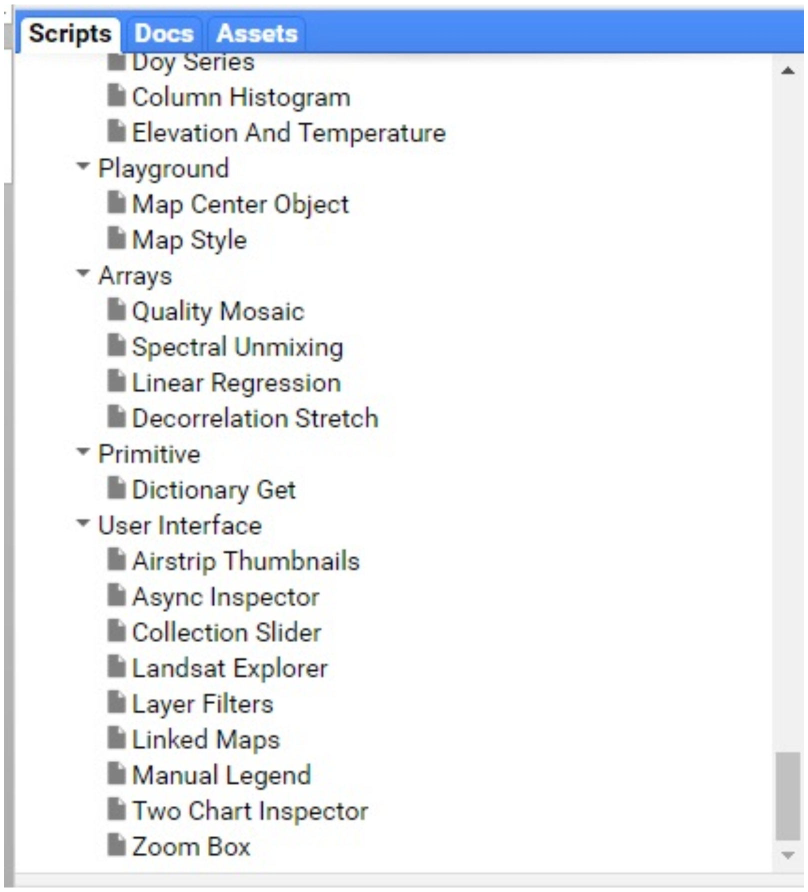
```
Imports (8 entries)
 var Water: FeatureCollection (37 elements)
 var MangroveForest: FeatureCollection (77 elements)
 var BareGround_Developed: FeatureCollection (13 elements)
 var FreshWaterMarsh: FeatureCollection (70 elements)
 var SawGrassPrairie: FeatureCollection (21 elements)
 var SaltWaterMarsh: FeatureCollection (49 elements)
 var Shrub_Scrub: FeatureCollection (14 elements)
 var imageMasked: Image users/acw2kd/everglades/2015 (13 bands)

//import park boundary and GRTS random sampling grid from fusion table (shapeFiles)
var parkBoundary = ee.FeatureCollection('ft:1KAYShQzYibOCQkxscbvvdNEq0JINOcYLy9wOLIF1','geometry');
//GRTS Sample Grids for Training
var GRTS_Sample = ee.FeatureCollection('ft:1vmcnhrotb2Q_BEzMzsFQUyrl7B6Z7jMH21DYVeCS','geometry');
//GRTS Buffer for calculating Area of each Class
var GRTS2 = ee.FeatureCollection('ft:1ezj-PlbayKFcCQB5I2ZO0O3iEONKRZwhQh8xaSCL', 'geometry');
//In Situ Data Points
var vegetationMap = ee.FeatureCollection('ft:10xA_6r-9e0jm6eE_kXAnX9QN40f54ivQZiWwOra1');

//merge all geometry imports for classification
var newfc = Water.merge(MangroveForest).merge(SaltWaterMarsh).merge(Shrub_Scrub).merge(FreshWaterMarsh).merge(BareGround_Developed).merge(SawGrassPrairie);

//Create a dictionary explaining the class meanings
var classes = [
  {'landcover':0,'description':'Water'},
```

```
Use print(...) to write to this console.

Class Descriptions
List (7 elements)

Error Matrix:
List (6 elements)

Overall Accuracy:
0.9763739085772984

Kappa Coefficient 2005:
0.9474540790823093
```

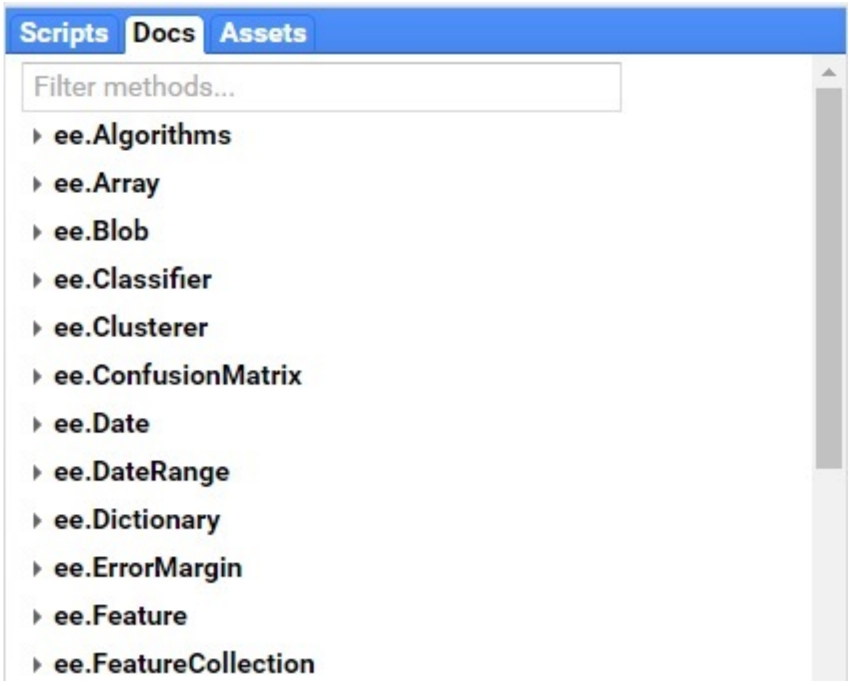
2. The ‘Scripts’ tab is where you will create your own codes. You can place new files into folders for easy organization.



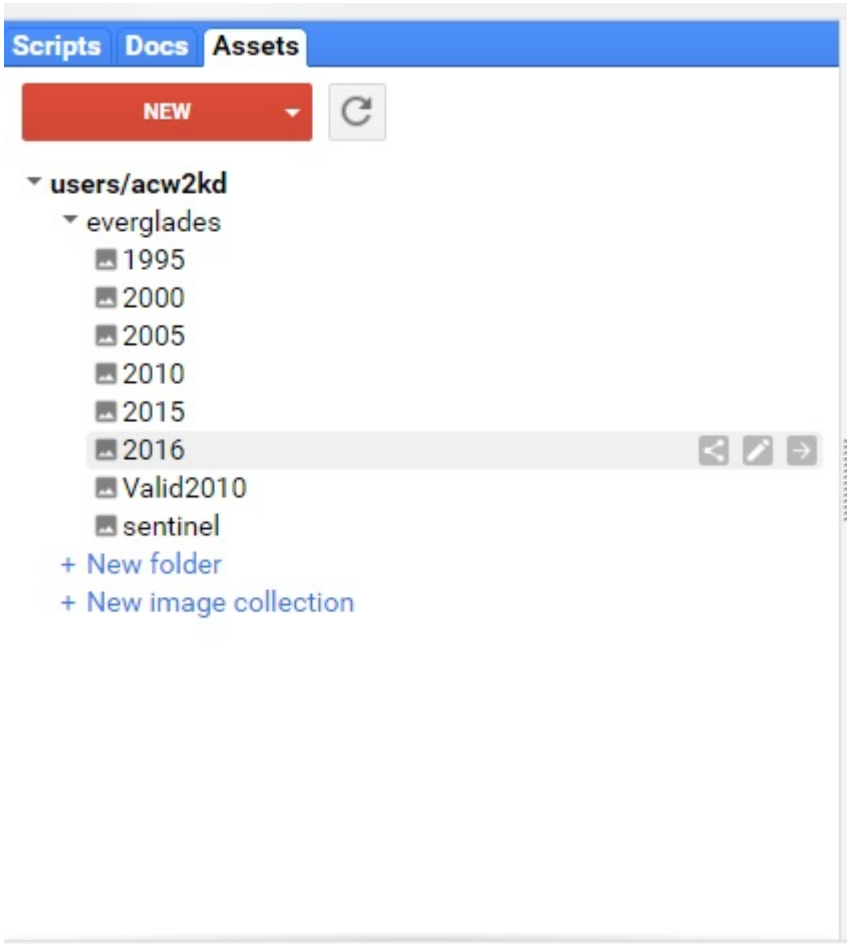
3. GEE also has example codes listed at the bottom of the 'Scripts' tab. Some of these let you customize your user interface.



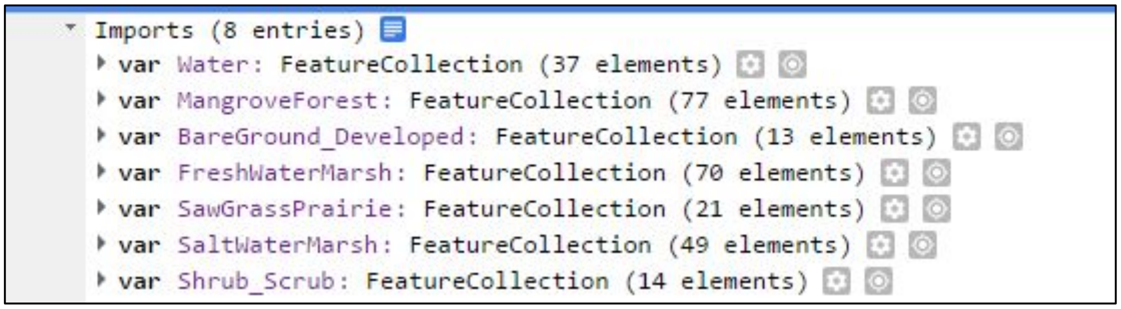
4. The 'Docs' tab is essentially a dictionary of all of the algorithms and functions that you can use in GEE.



5. The 'Assets' tab is where you will find any images that you export to the EE Assets or any images that you import into the Imports section.



6. The Imports section shows imported information. This image displays the different class types for your land change analysis that was created through the Geometry Imports. We used 6 classes. Classes depend on the user preference.

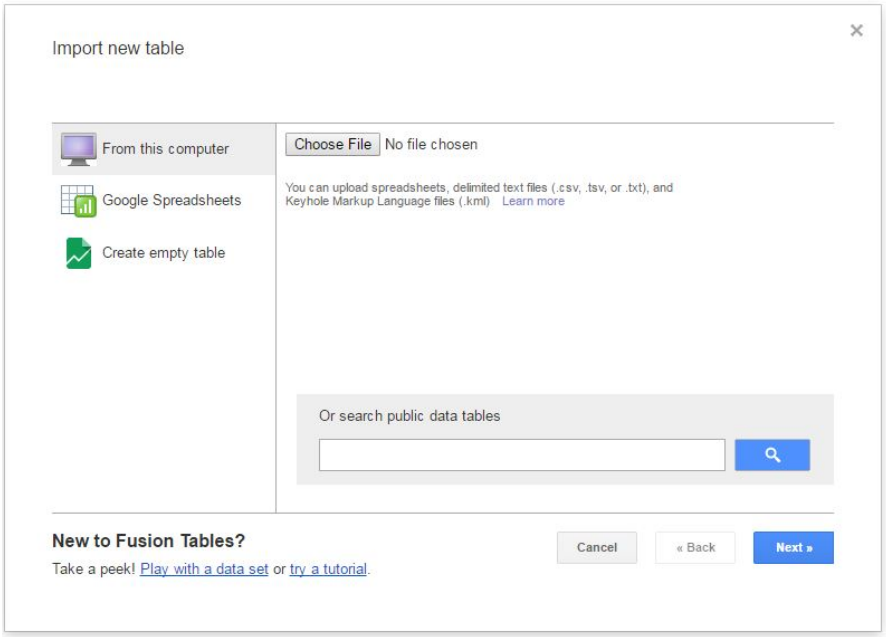


```
Imports (8 entries)
  var Water: FeatureCollection (37 elements)
  var MangroveForest: FeatureCollection (77 elements)
  var BareGround_Developed: FeatureCollection (13 elements)
  var FreshWaterMarsh: FeatureCollection (70 elements)
  var SawGrassPrairie: FeatureCollection (21 elements)
  var SaltWaterMarsh: FeatureCollection (49 elements)
  var Shrub_Scrub: FeatureCollection (14 elements)
```

7. To import a specific study area shapefile, you will need to create a fusion table. Google provides an extension to the Chrome browser called Fusion Tables. Import the shapefile as a kml file into the fusion table app. Complete steps are as follows:
   a. To export polygons for use in Google Earth Engine:

ArcMap - add data (training samples shapefile) > toolbox > conversion > Layer to KML > input> (training samples shapefile) > in destination folder right click (training samples shapefile) > 7zip > extract

b. Google Fusion Tables (you'll need to install fusion table in your browser) - https://chrome.google.com/webstore/detail/fusion-tables-experimenta/pfoeakahkgllhkommkfeehmkfcloagkl
c. 'choose file' > upload KML of shapefile > download as .csv > delete unnecessary columns in csv and add the coded class names > repeat upload
d. now you have a fusion table



You can upload spreadsheets, delimited text files (.csv, .tsv, or .txt), and Keyhole Markup Language files (.kml) Learn more

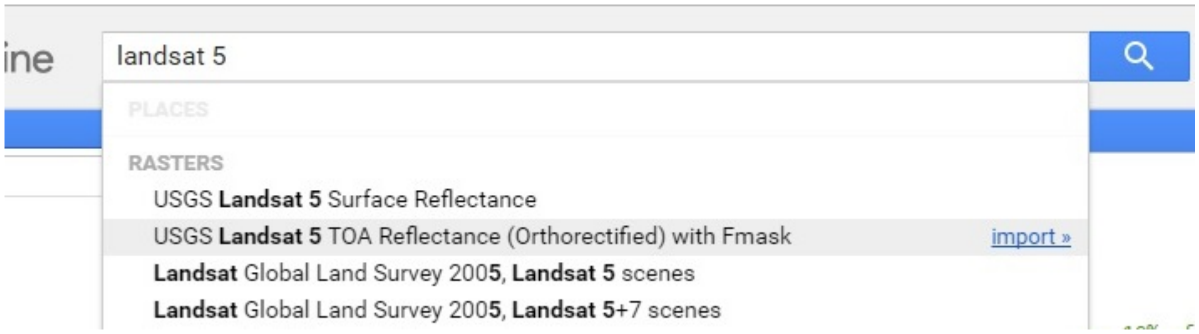
New to Fusion Tables?

Take a peek! Play with a data set or try a tutorial.

8. At the very top of the code, import the shapefile with the ee.FeatureCollection('link-to-fusion-table', 'geometry'); function.
9. Set location/extent. These are coordinates. The '10' represents the zoom level. Change location to your specific study area here.

```
Map.setCenter(-80.67012, 25.15795, 10);
```

10. Since clouds obstruct the land cover, removal of the clouds is imperative and complicated if thresholds are performed manually. Luckily, GEE contains an FMask function that will remove the clouds for us on each image by using the fmask band appended by USGS. The first step is to import the landsat collection that you wish to use. In the search box at the top of the code editor screen, type in Landsat 5. Choose the USGS Landsat 5 TOA Reflectance (Orthorectified) with Fmask and choose 'import'. This will import every scene of the Landsat 5 collection. You may re-name this new import if you wish.



landsat 5

PLACES

RASTERS

USGS **Landsat 5** Surface Reflectance

USGS **Landsat 5** TOA Reflectance (Orthorectified) with Fmask import »

Landsat Global Land Survey 2005, **Landsat 5** scenes

Landsat Global Land Survey 2005, **Landsat 5**+7 scenes

11. To create a cloud masking function, the following code can be used which will select out only clear pixels that are less than '2'. Pixel representations are as follows:

fmask: cloud mask

- 0=clear
- 1=water
- 2=shadow
- 3=snow
- 4=cloud

Cloud masking function code:

```
function applyMask(image) {
        return image.updateMask(image.select('fmask').lt(2));}
```

12. Since we do not want every scene supplied by the Landsat 5 satellite sensor, we can filter the collection to only the dates that we are interested in. The following code is an example for the year 1995

```
var collection95 = ee.ImageCollection(landsat5)
  .filterDate('1995-01-01', '1995-05-30')        //define time frame
  .map(applyMask);                               //applies the mask
```

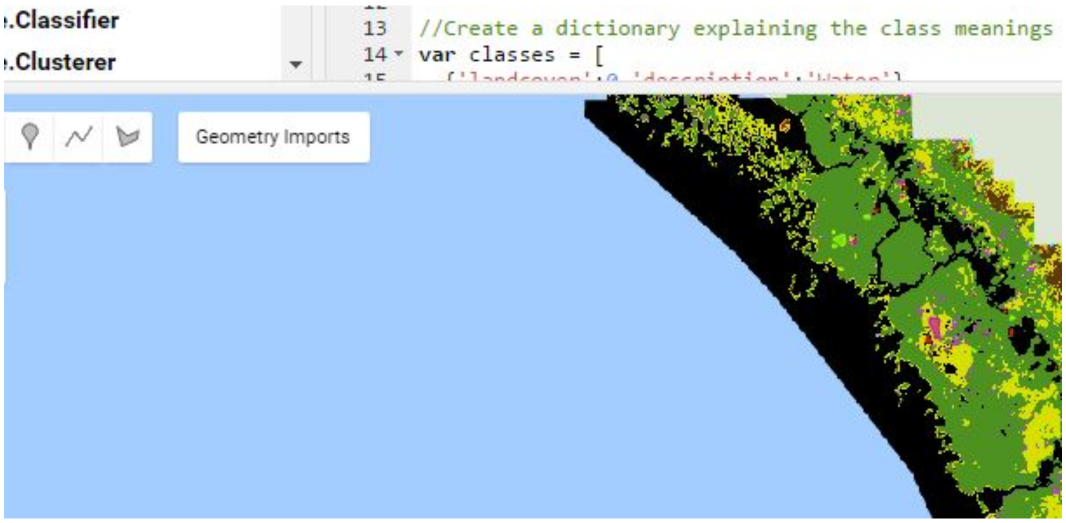
13. We also only want one cloud free image clipped to the area of interest. In order to do this, we will use the reducer to aggregate all of the cloud-masked scenes for the time period specified and then clip it to the Everglades National Park boundary.

```
var merged95 = combine95.reduce(ee.Reducer.median())
   .clip(parkBoundary);
```

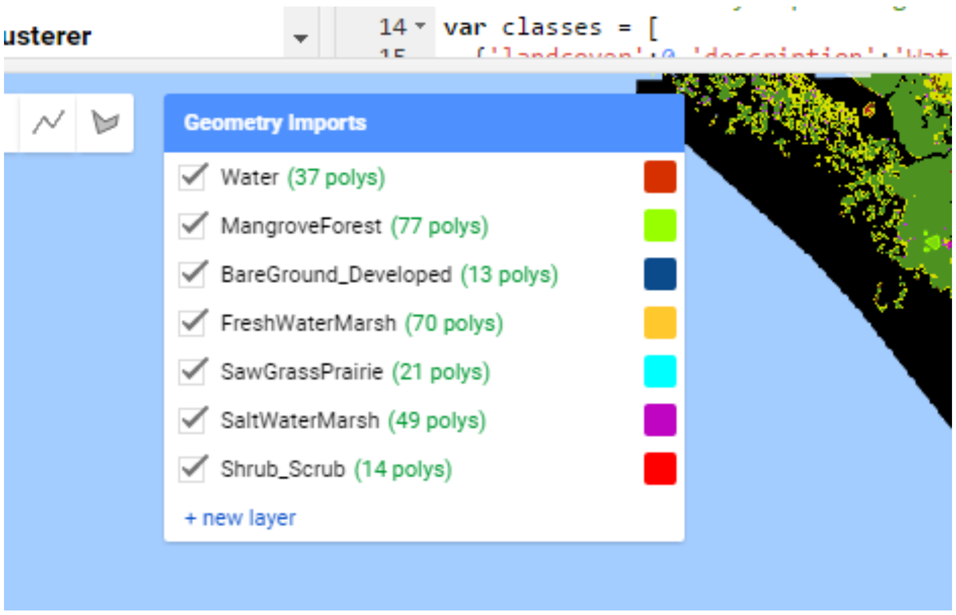
14. Add it to the GEE user interface to see the newly created cloud free image on the map using false color composite

    Map.addLayer(masked, {bands: ['B5_median', 'B4_median', 'B3_median'], max: 0.3}, 'masked image');

15. Now that we have a cloud free image, we can start creating our training polygons. To create classes: Click on Geometry Imports.



```
//Create a dictionary explaining the class meanings
var classes = [
```



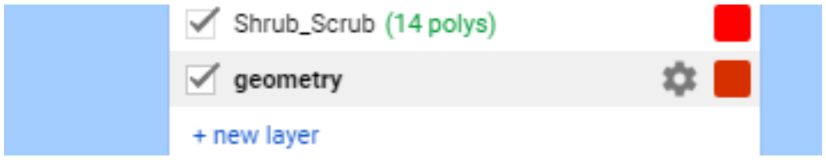
```
var classes = [
```

Geometry Imports

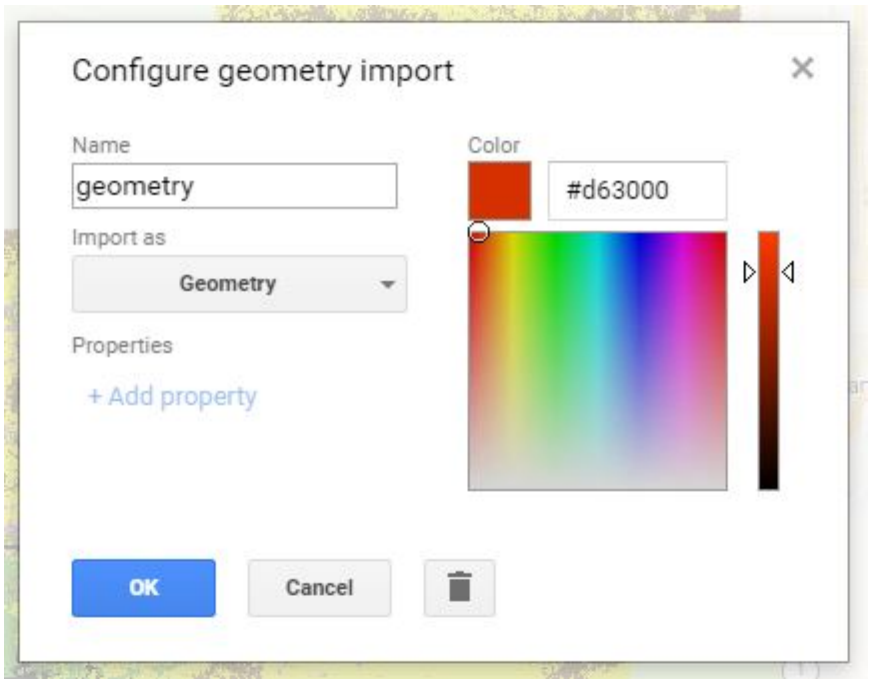
- ✓ Water (37 polys) 🟥
- ✓ MangroveForest (77 polys) 🟩
- ✓ BareGround_Developed (13 polys) 🟦
- ✓ FreshWaterMarsh (70 polys) 🟨
- ✓ SawGrassPrairie (21 polys) 🟦
- ✓ SaltWaterMarsh (49 polys) 🟪
- ✓ Shrub_Scrub (14 polys) 🟥

+ new layer

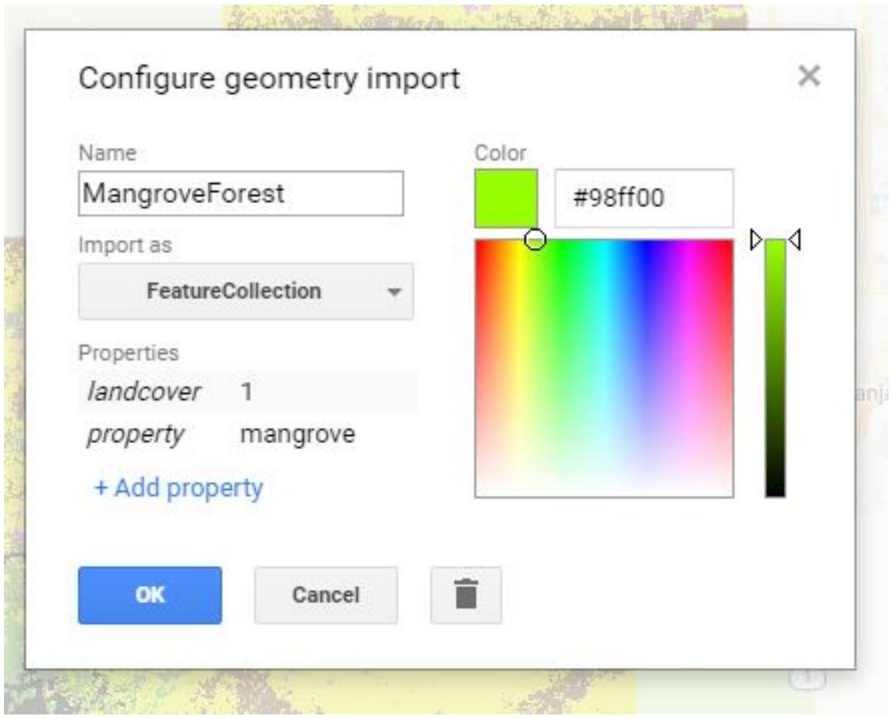
16. Click on + new layer. A geometry class shows up. Click on settings tab.



17. This is the setting menu for delegating a class name. Text placed within this box ***is case sensitive*** and needs to be in the ***same format*** as the other class types.



18. The class type should be formatted like below. Change the drop down menu to FeatureCollection. Properties should include



Configure geometry import

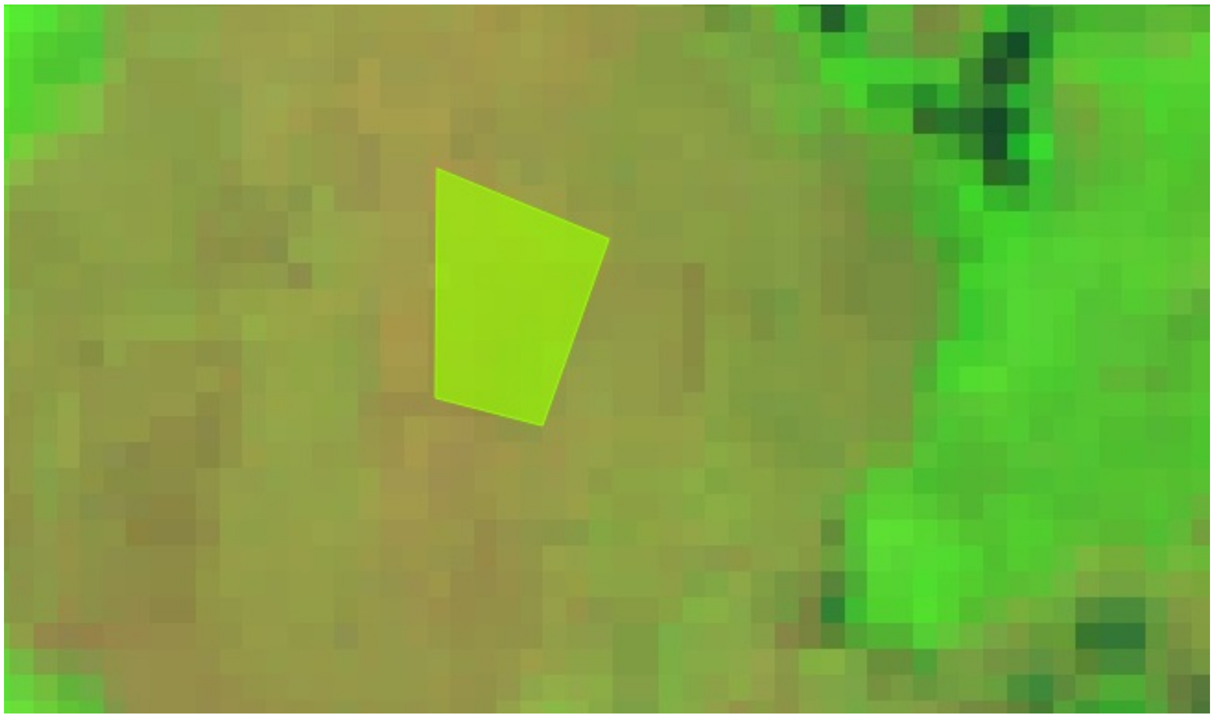
Name: MangroveForest

Import as: FeatureCollection

Properties

landcover	1
property	mangrove

19. You are now ready to create your first training site. Simply click each corner of the polygon that you wish to draw that covers the pixels which ***only contain the class***



20. Continue with steps 10 – 15 until you have created all of your classes and training sites.

21. GEE offers multiple algorithms to classify images. This example with use the Random Forest Classifier with 100 trees. In order to begin, all of the training sites need to be merged into one variable.

```
//merge all geometry imports for classification
var newfc =
Water.merge(MangroveForest).merge(SaltWaterMarsh).merge(Shrub_Scrub).merge(Fre
shWaterMarsh).merge(BareGround_Developed).merge(SawGrassPrairie);
```

22. A random number column will be used on the merged training sites. Different seeds can be used to obtain different results (1, 2, 3, etc.)

```
var newfc2 = newfc.randomColumn('random', 2);
```

23. Define the classification samples to include newfc2 and the properties we want the program to consider. Set the scale to 30 to match the pixel resolution of the Landsat 5 image – in this case, 30 meters

```
var samples = masked.sampleRegions({
  collection: newfc2,
  properties: ['landcover', 'random'],
  scale: 30 });
```

24. Split the training points into 90% and 10% for classification and the accuracy assessment

```
var training1995 = samples.filterMetadata('random', 'less_than', 0.9);
var testing1995 = samples.filterMetadata('random', 'not_less_than', 0.9);
```

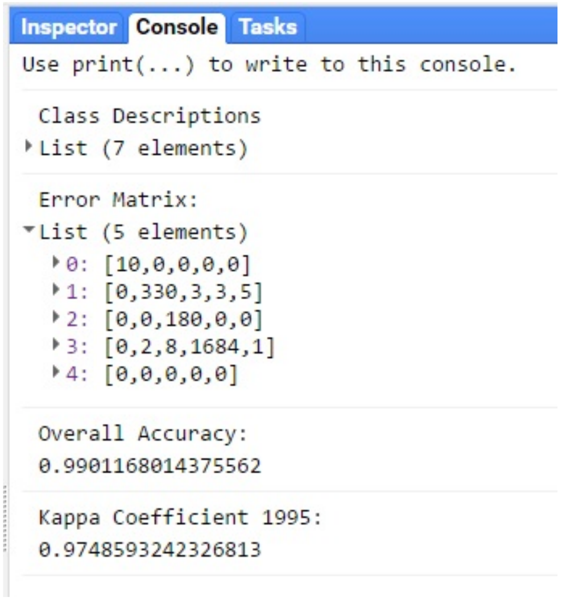
25. Only use 90% of the training samples for classification.

```
var classifier = ee.Classifier.randomForest(100).train({
 features: training1995,
 classProperty: 'landcover'});

//apply classifier
var classified = masked.classify(classifier);
```

26. To validate the classification and produce an error matrix, the following example code can be used and modified where 10% of the polygons are used for testing:

```
var validation1995 = testing1995.classify(classifier);
var errorMatrix1995 = validation1995.errorMatrix('landcover', 'classification');
print('Error Matrix:', errorMatrix1995);
print('Overall Accuracy:', errorMatrix1995.accuracy());
print('Kappa Coefficient 1995: ', errorMatrix1995.kappa());
```

27. The Console (located on the right side of the user interface) will display the results of the analysis.



```
Use print(...) to write to this console.

Class Descriptions
▸List (7 elements)

Error Matrix:
▾List (5 elements)
  ▸0: [10,0,0,0,0]
  ▸1: [0,330,3,3,5]
  ▸2: [0,0,180,0,0]
  ▸3: [0,2,8,1684,1]
  ▸4: [0,0,0,0,0]

Overall Accuracy:
0.9901168014375562

Kappa Coefficient 1995:
0.9748593242326813
```

**Clicking on the inspector tab will allow you to see the properties of each pixel. Your cursor will turn into a cross-hair and information will display in the inspector tab once you click on the map. The Tasks tab is where any of your exports will be located.

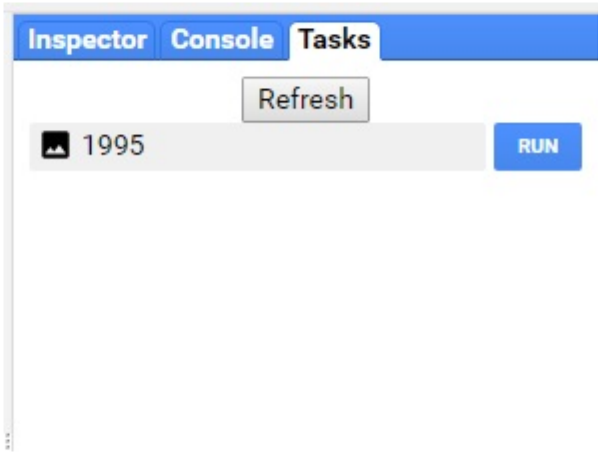
28. Display the new classified image on the map

```
//Assign a palette scheme
var palette = ['000000', //Water - Black
        '4d9221', //Mangrove Forest - dark green
        '61380B', //Freshwater Marsh - brown
        'D7DF01', //Saltwater Marsh - Yellow
        'CC2EFA', //Shrub/Scrub - Magenta
        'b2182b', //Bare Ground_Developed - Red
        //'2E9AFE' //Sawgrass Prairie - Yellow
        ];

//add classification based on training points with the assigned palette color

Map.addLayer(classified,

          {min: 0, max: 5, palette: palette}, 'classification');
```

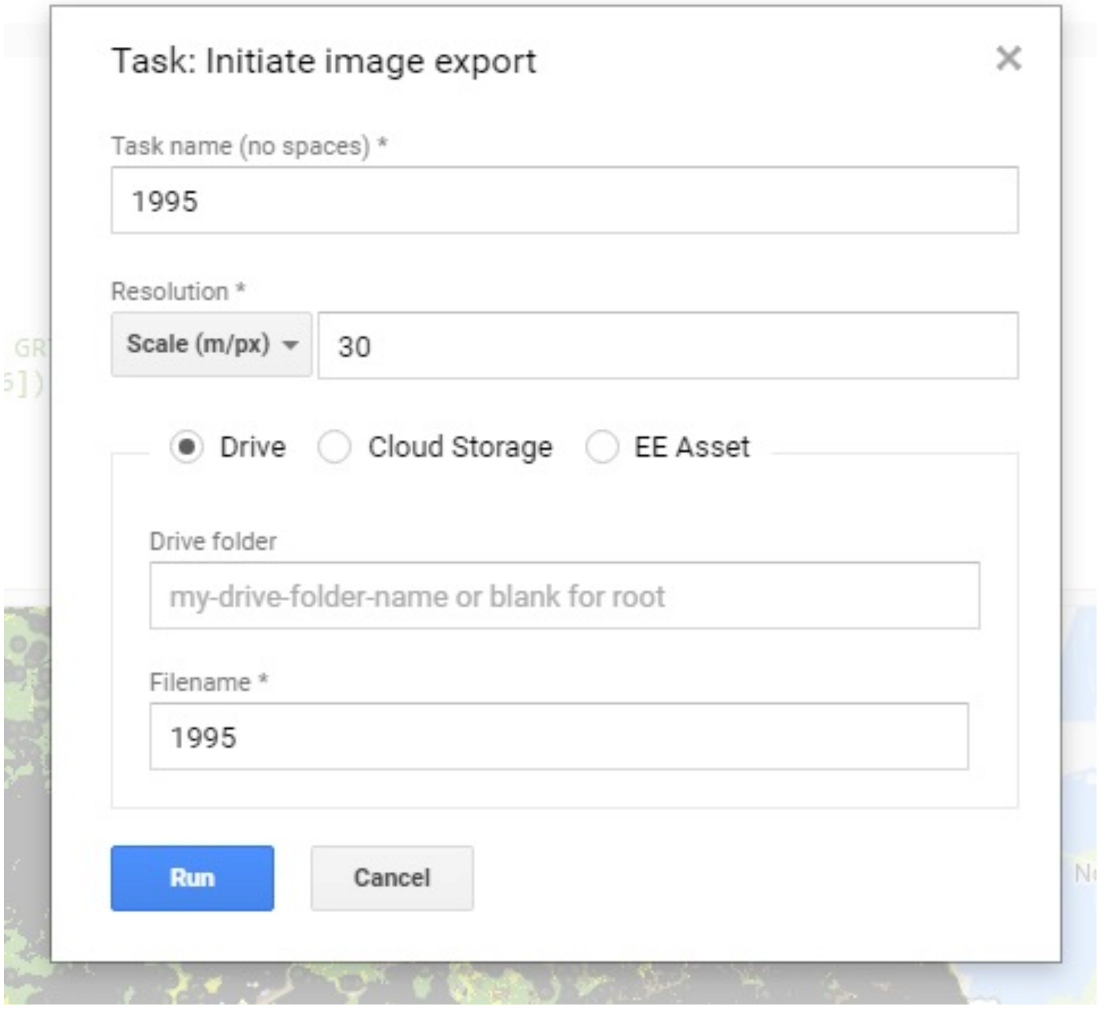
29. Export the map if desired and clip to the region

```
//export image to google drive
Export.image.toDrive({
 image: classified,
 description: '1995',
 scale: 30,
 region: parkBoundary});
```

30. Click 'Run'. Assign the file a task name, resolution, and a filename. You can export it as a .tif file to your personal google drive, cloud storage, or as an Earth Engine Asset (which can be imported into a different code).



1995



Task name (no spaces) *: 1995

Resolution *: Scale (m/px) 30

Drive [✓] Cloud Storage EE Asset

Drive folder: my-drive-folder-name or blank for root

Filename *: 1995