CSci 5715 – LAB 4 Part C

This Part contains two sections: A – Google Earth Engine; B – Visual change identification.

A. Google Earth Engine (GEE) (https://code.earthengine.google.com/)
   For all the steps, if you need additional detailed references, please visit:
   https://developers.google.com/earth-engine/tutorials
   OR https://developers.google.com/earth-engine/changelog
   For all the question, include both of your code and result in the lab report.

1. Getting started
   a. Open your GEE playground, and the (Google Chrome) page should be visible as follows,

   ![Google Earth Engine Interface]

   var image = ee.Image('srtm90_v4'); // Note. The quotes (') can cause error, please retype the quotes in the script.
   Map.setCenter(-110, 40, 5); // Last parameter is the zoom level
   Map.addLayer(image);

   Show the map you see in the interface. Can you easily distinguish different elevation values?

   b. Getting started by adding an elevation raster (map layer) to the map:

   var image = ee.Image('srtm90_v4'); // Note. The quotes (') can cause error, please retype the quotes in the script.
   Map.setCenter(-110, 40, 5); // Last parameter is the zoom level
   Map.addLayer(image);

   Show the map you see in the interface. Can you easily distinguish different elevation values?

   c. Change the last line of code to:

   Map.addLayer(image, {min: 0, max: 3000});

   This line sets a range of values to be stretched. All values below min will get the lowest luminance while that above max will get the highest. Show the new visualization.

   Can you see a better representation after the change? Create your own min-max value pairs to get the visualization you think is the best and show it.
2. Map algebra – focal analysis
Focal analysis on an image is a computation involving all elements within a spatial window (e.g., 3 by 3, 5 by 5 in size). Terrain indexes, such as slope and aspect of digital elevation models can be calculated using focal analysis and this function is available in GEE.

a. On the left panel, switch to “Docs” tab, and scroll down to find ee.Terrain as shown in the following figure. All the functions needed to process terrains in this set of questions are available here.

b. Compute the slope of the terrain and show it for the whole United States. Use the visualization enhancing method you just tried in 1 to make sure differences can be easily distinguished. If you want to know the current min-max range of visualization, you check expand layers on the top-right corner of the map and view the properties.

Note. You may find the following link helpful, [https://developers.google.com/earth-engine/tutorial_api_03](https://developers.google.com/earth-engine/tutorial_api_03)
c. Do the same task in 2b for terrain aspect and hillshade.
d. For better visualization purpose, color-code can be added as a property of map layers. For example:

```javascript
var colorcodes = ['000000', 'F0F0F0', '111111'];
Map.addLayer(map1, { min: 0, max: 255, palette: colorcodes });
'F0F0F0', etc., used here is the RGB color code and they can be found at:
http://www.rapidtables.com/web/color/RGB_Color.htm
For slope, create a new color code array with three colors in the following order: Green, Yellow, Red. Apply this to the slope result with appropriate min-max stretch (values of your choice) and show the result.
```

3. Local
Local analysis is operated on multiple images (e.g., two or three). The output for each pixel location is computed using the pixel values at the exactly same geographical location in the inputs.

a. Between bands
Firstly, we will perform local analysis on different bands of the same image. Our inputs are the red band (R) and the near infrared band (NIR). The result to compute here is the Normalized Difference Vegetation Index (NDVI), which is an index of plant “greenness” or photosynthetic activity. NDVI is computed as: $\text{NDVI} = \frac{(\text{NIR} - \text{R})}{(\text{NIR} + \text{R})}$

The following code is provided in GEE tutorials showing how to compute another index called Enhanced Vegetation Index (EVI). The red colored line shows the mathematical expression used and the green lines show the bands used from the image. Use the same image in this code snippet: 'LANDSAT/LC8_L1T_TOA/LC80440342014077LGN00' and compute the NDVI with image.expression. Show the modified code and the result.

```javascript
// Load a Landsat 8 image.
var image = ee.Image('LANDSAT/LC8_L1T_TOA/LC80440342014077LGN00');
// Compute the EVI using an expression. var evi = image.expression(
"2.5 * ((NIR - RED) / (NIR + 6 * RED - 7.5 * BLUE + 1))", {
  'NIR': image.select('B5'),
  'RED': image.select('B4'),
  'BLUE': image.select('B2')
});
Map.setCenter(-122, 37, 7);
Map.addLayer(evi);
```

4. Metadata
Metadata contains critical information of a GIS dataset, such as coordinate system, projection, date created, precision and owner. Here we are interested in retrieving the projection and time information of a dataset.

The following code snippet is provided by GEE. It demonstrates how to extract projection and timestamp information out of a raster or image.

```javascript
var image = ee.Image('LANDSAT/LC8_L1T/LC80440342014077LGN00');

// Get projection information from b and 1.
var b1proj = image.select('B1').projection();
print('Band 1 projection: ', b1proj);

// Get the timestamp and convert it to a date.
var date = ee.Date(image.get('system:time_start'));
print('Timestamp: ', date);
```

Modify the code above and print the projection and timestamp information of the elevation image used at the beginning of the exercise: 'srtm90_v4'. More details can be found at https://developers.google.com/earth-engine/image_info

B. Visual change identification
Visit Time-Lapse at: [http://world.time.com/timelapse/](http://world.time.com/timelapse/) Click “Explore the world” at the lower-right corner of the interface and search for the corresponding time-lapse for the city assigned to you as shown in the following table.

Pick two scenes as well as the two scenes at the very beginning and the very end. Your chosen scenes should well reflect the changes happened over the years. Show the scenes in your report and demonstrate what the major changes you observe. (In case you cannot observe any major changes, consider zooming in one or more levels but not too many)

<table>
<thead>
<tr>
<th>Team Number</th>
<th>City Name</th>
<th>Team Number</th>
<th>City Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Minneapolis</td>
<td>14</td>
<td>San Diego</td>
</tr>
<tr>
<td>2</td>
<td>Duluth</td>
<td>15</td>
<td>Phoenix</td>
</tr>
<tr>
<td>3</td>
<td>Buffalo</td>
<td>16</td>
<td>Saint Paul</td>
</tr>
<tr>
<td>4</td>
<td>San Antonio</td>
<td>17</td>
<td>Redlands</td>
</tr>
<tr>
<td>5</td>
<td>Ontario (CA)</td>
<td>18</td>
<td>Seattle</td>
</tr>
<tr>
<td>6</td>
<td>Honolulu</td>
<td>19</td>
<td>Palm Springs</td>
</tr>
<tr>
<td>7</td>
<td>Anchorage</td>
<td>20</td>
<td>Houston</td>
</tr>
<tr>
<td>8</td>
<td>Orlando</td>
<td>21</td>
<td>Nashville</td>
</tr>
<tr>
<td>9</td>
<td>Las Vegas</td>
<td>22</td>
<td>Pittsburgh</td>
</tr>
<tr>
<td>10</td>
<td>Austin (TX)</td>
<td>23</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>11</td>
<td>Oklahoma City</td>
<td>24</td>
<td>Kansas City</td>
</tr>
<tr>
<td>12</td>
<td>Long Beach</td>
<td>25</td>
<td>Memphis</td>
</tr>
<tr>
<td>13</td>
<td>Miami</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>