Voss_Kristofor_GEOTiff

July 8, 2018

1 Plotting a NEON RGB Camera Image

1.1 Tutorial Code

```
In [2]: import gdal
```

```
In [3]: import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
        import warnings
        warnings.filterwarnings('ignore')
        def RGBraster2array(RGB_geotif):
            """RGBraster2array reads in a NEON AOP geotif file and returns
            a numpy array, and header containing associated metadata with spatial information.
            --------
            Parameters
                RGB_geotif -- full or relative path and name of reflectance hdf5 file
            --------
            Returns
            --------
            array:
                numpy array of geotif values
            metadata:
                dictionary containing the following metadata (all strings):
                    array_rows
                    array_cols
                    bands
                    driver
                    projection
                    geotransform
                    pixelWidth
                    pixelHeight
                    extent
                    noDataValue
                    scaleFactor
            --------
            Example Execution:
            --------
```

```
    RGB_geotif = '2017_SERC_2_368000_4306000_image.tif'
    RGBcam_array, RGBcam_metadata = RGBraster2array(RGB_geotif) """

    metadata = {}
    dataset = gdal.Open(RGB_geotif)
    metadata['array_rows'] = dataset.RasterYSize
    metadata['array_cols'] = dataset.RasterXSize
    metadata['bands'] = dataset.RasterCount
    metadata['driver'] = dataset.GetDriver().LongName
    metadata['projection'] = dataset.GetProjection()
    metadata['geotransform'] = dataset.GetGeoTransform()

    mapinfo = dataset.GetGeoTransform()
    metadata['pixelWidth'] = mapinfo[1]
    metadata['pixelHeight'] = mapinfo[5]

    metadata['ext_dict'] = {}
    metadata['ext_dict']['xMin'] = mapinfo[0]
    metadata['ext_dict']['xMax'] = mapinfo[0] + dataset.RasterXSize/mapinfo[1]
    metadata['ext_dict']['yMin'] = mapinfo[3] + dataset.RasterYSize/mapinfo[5]
    metadata['ext_dict']['yMax'] = mapinfo[3]

    metadata['extent'] = (metadata['ext_dict']['xMin'],metadata['ext_dict']['xMax'],
                          metadata['ext_dict']['yMin'],metadata['ext_dict']['yMax'])

    raster = dataset.GetRasterBand(1)
    array_shape = raster.ReadAsArray(0,0,metadata['array_cols'],metadata['array_rows']]
    metadata['noDataValue'] = raster.GetNoDataValue()
    metadata['scaleFactor'] = raster.GetScale()

    array = np.zeros((array_shape[0],array_shape[1],dataset.RasterCount),'uint8') #pre
    for i in range(1, dataset.RasterCount+1):
        band = dataset.GetRasterBand(i).ReadAsArray(0,0,metadata['array_cols'],metadata
        band[band==metadata['noDataValue']]=np.nan
        band = band/metadata['scaleFactor']
        array[...,i-1] = band

    return array, metadata
```

```
In [8]: RGB_geotif = 'C:\\Users\\Kris\\Desktop\\NEON Data Institute\\2017_SERC_2_368000_4306000
        SERC_RGBcam_array, SERC_RGBcam_metadata = RGBraster2array(RGB_geotif)
```

```
In [10]: SERC_RGBcam_array.shape
```

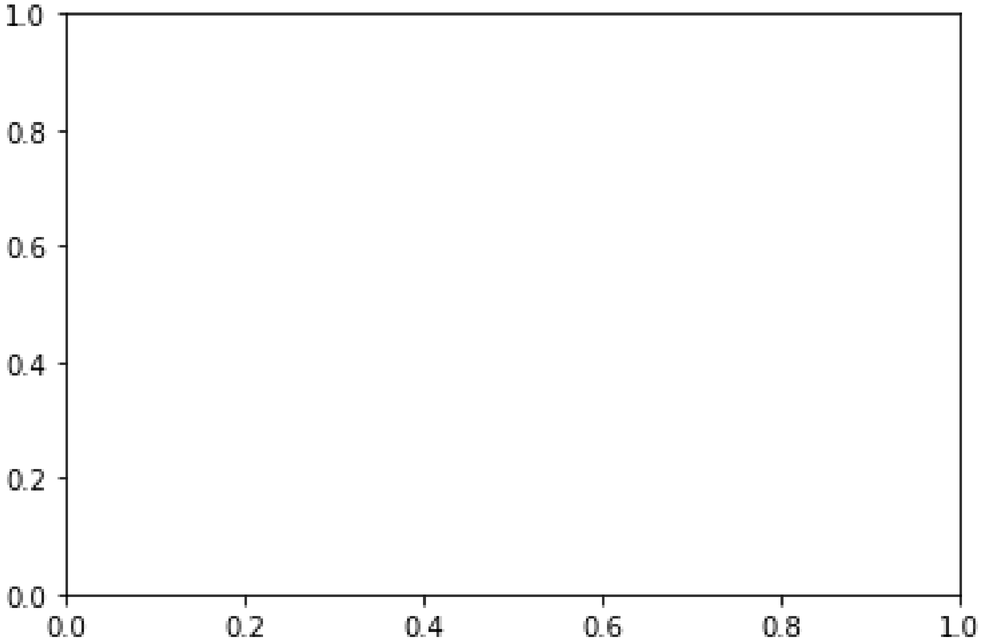
```
Out[10]: (10000, 10000, 3)
```

```
In [11]: #Display information stored in header
         for key in sorted(SERC_RGBcam_metadata.keys()):
           print(key)
```

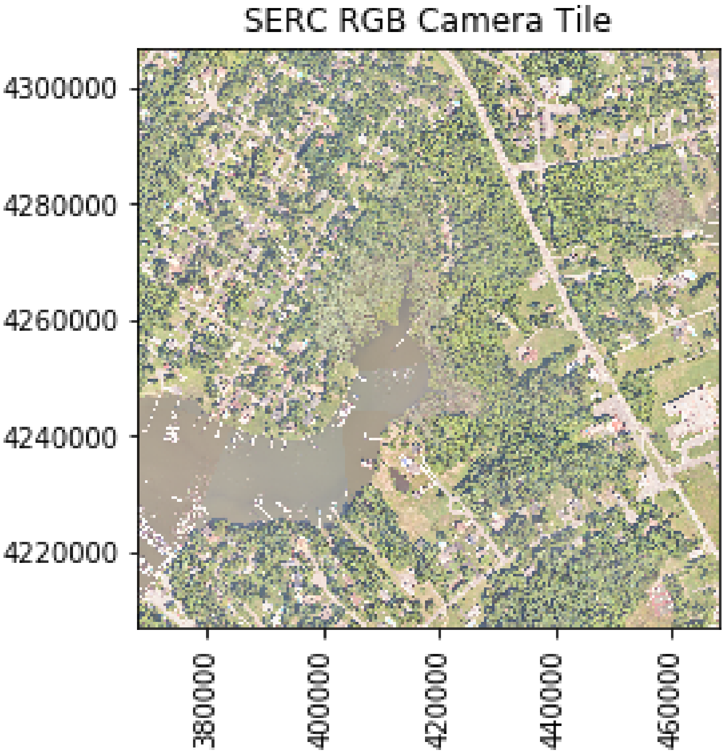
```
array_cols
array_rows
bands
driver
ext_dict
extent
geotransform
noDataValue
pixelHeight
pixelWidth
projection
scaleFactor
```

```
In [12]: def plot_band_array(band_array,
                             refl_extent,
                             colorlimit,
                             ax=plt.gca(),
                             title='',
                             cbar ='on',
                             cmap_title='',
                             colormap='spectral'):

             '''plot_band_array reads in and plots a single band or an rgb band combination of
             --------
             Parameters
             --------
                 band_array: flightline array of reflectance values, created from h5refl2array
                 refl_extent: extent of reflectance data to be plotted (xMin, xMax, yMin, yMax
                 colorlimit: range of values to plot (min,max). Best to look at the histogram
                 ax: optional, default = current axis
                 title: string, optional; plot title
                 cmap_title: string, optional; colorbar title
                 colormap: string, optional; see https://matplotlib.org/examples/color/colorma
             --------
             Returns
                 plots array of single band or RGB if given a 3-band
             --------
             Example:
             --------
             plot_band_array(SERC_RGBcam_array,
                             SERC_RGBcam_metadata['extent'],
                             (1,255),
                             title='SERC RGB Camera Tile',
                             cbar='off')'''

             plot = plt.imshow(band_array,extent=refl_extent,clim=colorlimit);
             if cbar == 'on':
```

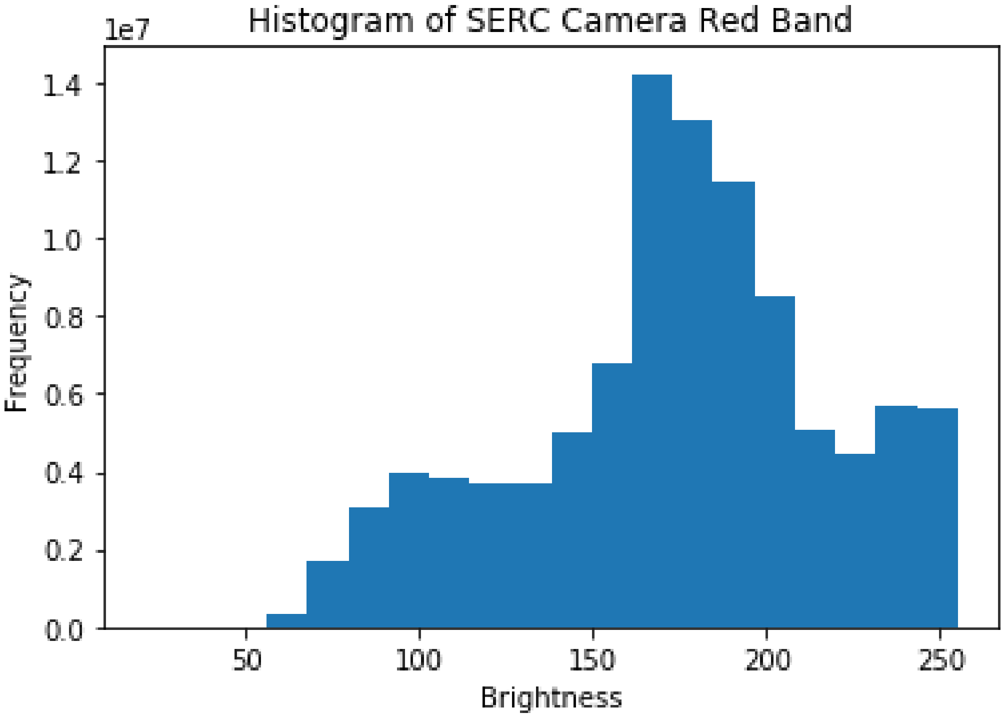
```
        cbar = plt.colorbar(plot,aspect=40); plt.set_cmap(colormap);
        cbar.set_label(cmap_title,rotation=90,labelpad=20)
    plt.title(title); ax = plt.gca();
    ax.ticklabel_format(useOffset=False, style='plain'); #do not use scientific notat
    rotatexlabels = plt.setp(ax.get_xticklabels(),rotation=90); #rotate x tick labels
```



```
In [13]: plot_band_array(SERC_RGBcam_array,
                         SERC_RGBcam_metadata['extent'],
                         (1,255),
                         title='SERC RGB Camera Tile',
                         cbar='off')
```

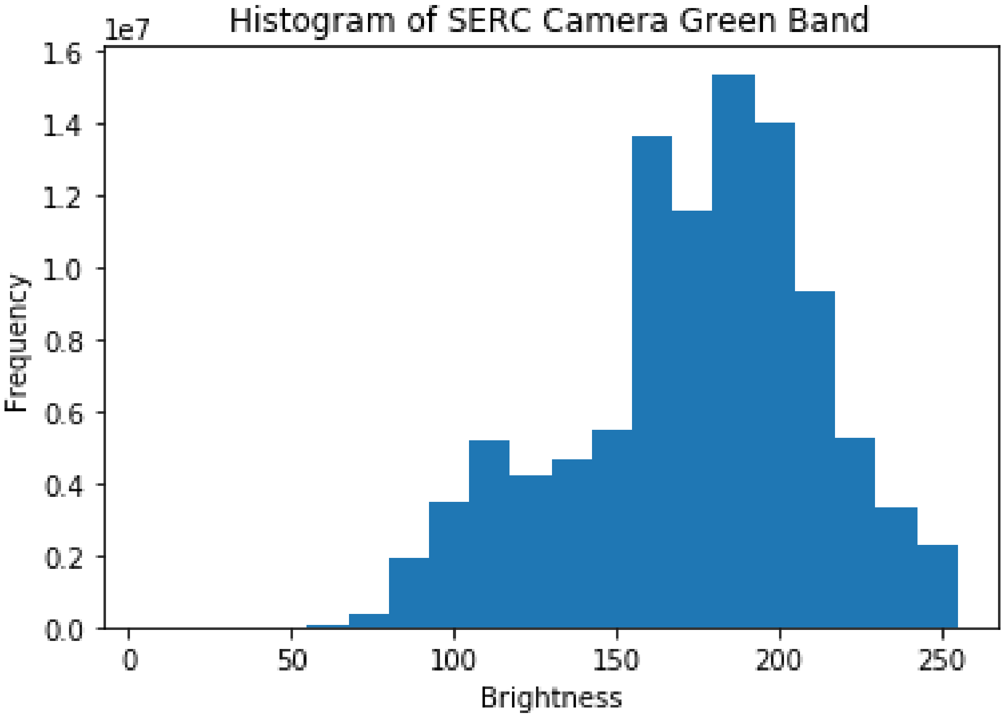


```
In [15]: plt.hist(np.ravel(SERC_RGBcam_array[:,:,0]),20);
         plt.title('Histogram of SERC Camera Red Band')
         plt.xlabel('Brightness'); plt.ylabel('Frequency')

Out[15]: Text(0,0.5,'Frequency')
```

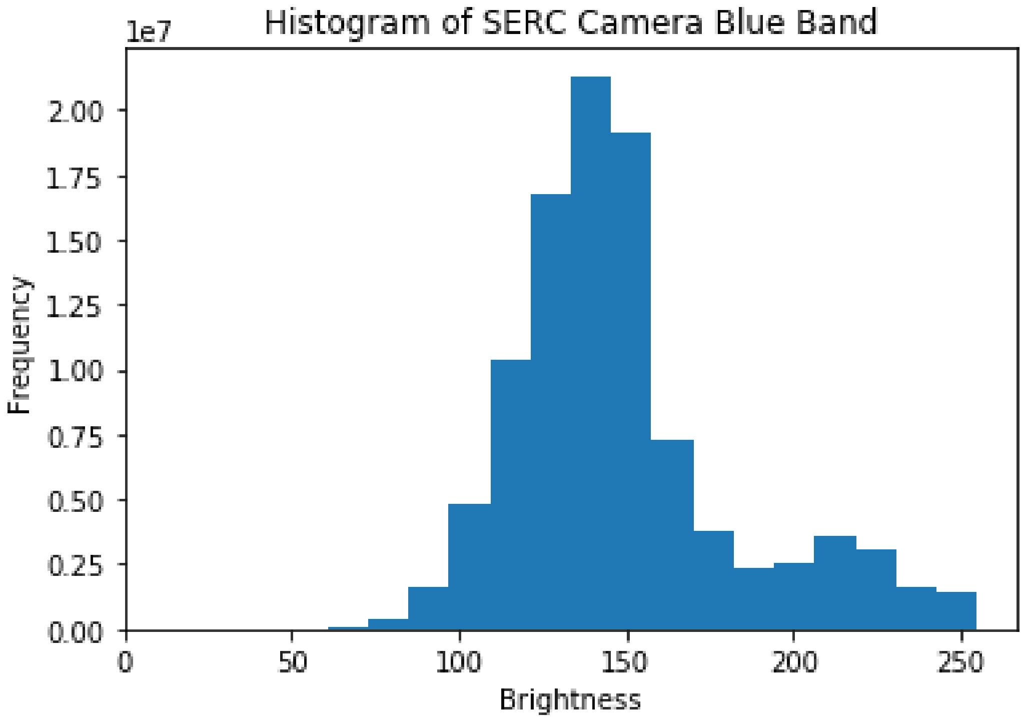


1.2 Challenge 1: Plot Blue & Green Band Histograms

```
In [19]: plt.hist(np.ravel(SERC_RGBcam_array[:,:,1]),20);
         plt.title('Histogram of SERC Camera Green Band')
         plt.xlabel('Brightness'); plt.ylabel('Frequency')

Out[19]: Text(0,0.5,'Frequency')
```



```
In [20]: plt.hist(np.ravel(SERC_RGBcam_array[:,:,2]),20);
         plt.title('Histogram of SERC Camera Blue Band')
         plt.xlabel('Brightness'); plt.ylabel('Frequency')

Out[20]: Text(0,0.5,'Frequency')
```



1.3 Challenge 2a: Min & Max of Each Band

```
In [28]: red_min = np.amin(SERC_RGBcam_array[:,:,0])
         print("The minimum value in the red band is", red_min)
         red_max = np.amax(SERC_RGBcam_array[:,:,0])
         print("The maximum value in the red band is", red_max)
```

```
The minimum value in the red band is 21
The maximum value in the red band is 255
```

```
In [29]: green_min = np.amin(SERC_RGBcam_array[:,:,1])
         print("The minimum value in the green band is", green_min)
         green_max = np.amax(SERC_RGBcam_array[:,:,1])
         print("The maximum value in the green band is", green_max)
```

```
The minimum value in the green band is 5
The maximum value in the green band is 255
```

```
In [30]: blue_min = np.amin(SERC_RGBcam_array[:,:,2])
         print("The minimum value in the blue band is", blue_min)
         blue_max = np.amax(SERC_RGBcam_array[:,:,1])
         print("The maximum value in the blue band is", blue_max)
```

```
The minimum value in the blue band is 12
The maximum value in the blue band is 255
```

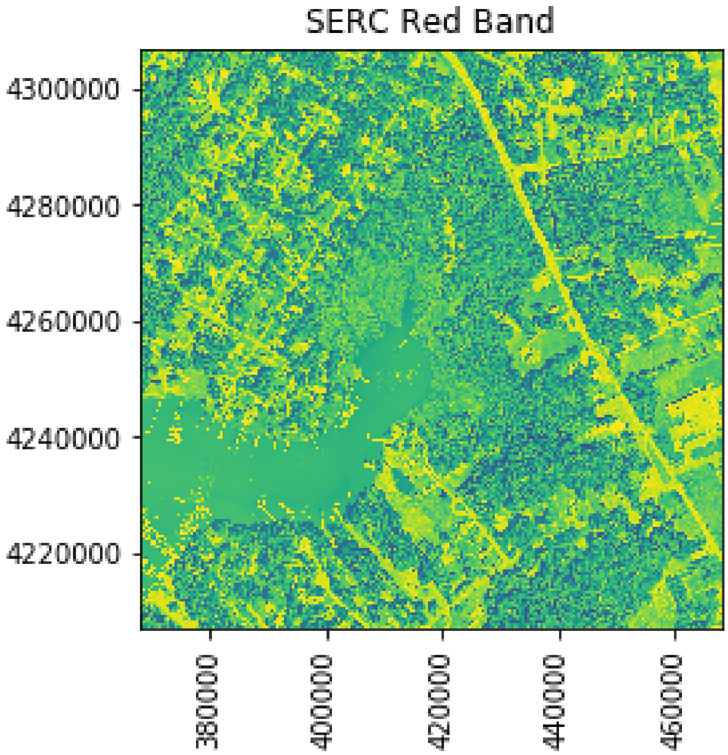
1.4 Challenge 2b: Print UTM Zone

```
In [31]: print("The UTM Zone for this image is", SERC_RGBcam_metadata['projection'])
```

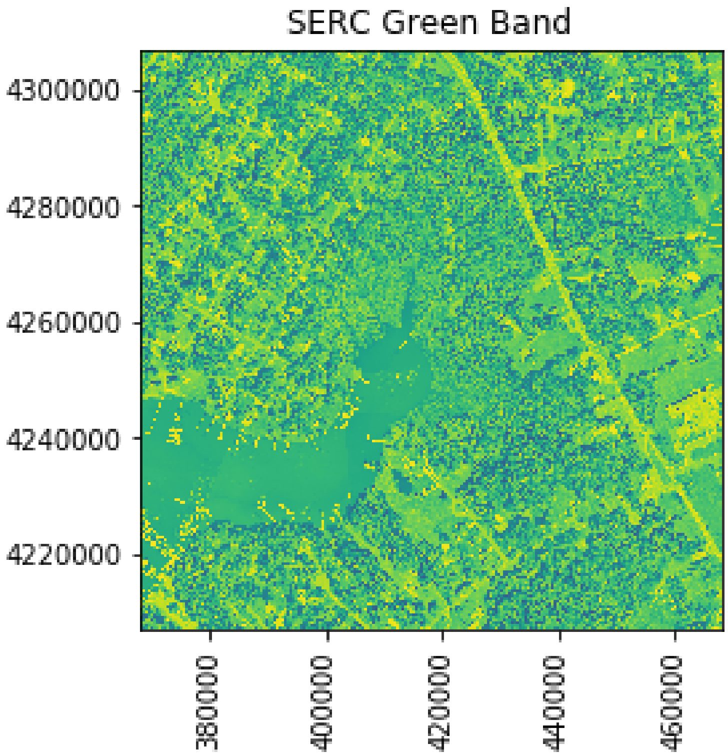
```
The UTM Zone for this image is PROJCS["WGS 84 / UTM zone 18N",GEOGCS["WGS 84",DATUM["WGS_1984"
```

1.5 Challenge 2c: Plot 3 Bands Separately

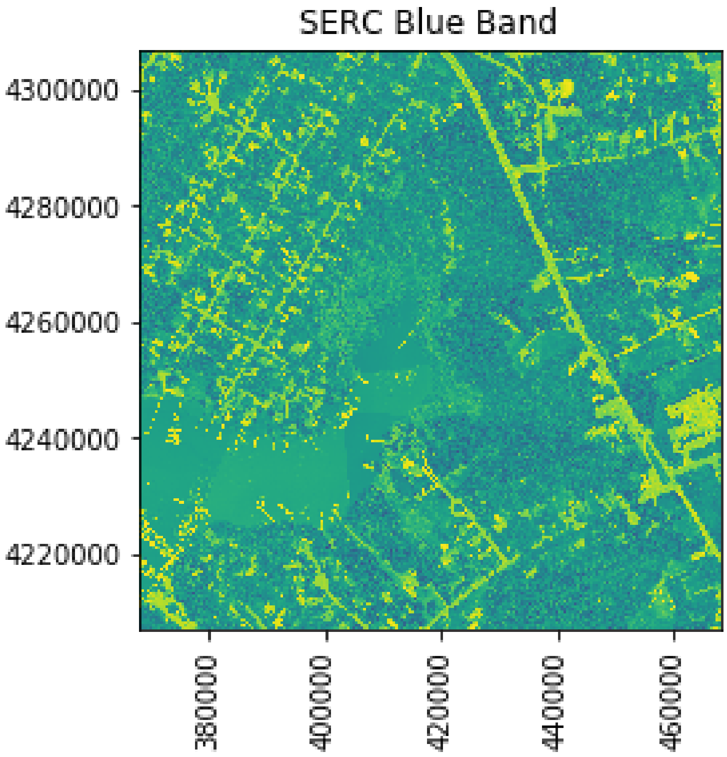
```
In [32]: plot_band_array(SERC_RGBcam_array[:,:,0],
                         SERC_RGBcam_metadata['extent'],
                         (1,255),
                         title='SERC Red Band',
                         cbar='off')
```



```
In [36]: plot_band_array(SERC_RGBcam_array[:,:,1],
                         SERC_RGBcam_metadata['extent'],
                         (1,255),
                         title='SERC Green Band',
                         cbar='off')
```



```
In [37]: plot_band_array(SERC_RGBcam_array[:,:,2],
                         SERC_RGBcam_metadata['extent'],
                         (1,255),
                         title='SERC Blue Band',
                         cbar='off')
```


SERC Blue Band
4300000
4280000
4260000
4240000
4220000
380000
400000
420000
440000
460000