

# NBR Mapping using Google Earth Engine

By Vanshika Gupta(16CV245)  
Seminar Report(CV390)  
National Institute of Technology Karnataka, Surathkal

Guided by Ms.Amba Shetty  
Department of Applied Mechanics and Hydraulics  
National Institute of Technology Karnataka, Surathkal

## 1 Abstract

The Objective of the seminar was to illustrate the use of Google Earth Engine to map the Normalised Burnt Ratio(NBR) and the Difference normalised burnt ratio(dNBR) for dataset of the Bandipur and the KM park project in the North Karnataka region. By using Google earth engine, we automate the procedure of importing the dataset, code up the algorithm involved and finally map the NBR. After mapping, we analysed the results using fire points data available.

## 2 Introduction

### 2.1 Google Earth Engine

The Google Earth Engine (GEE) platform contains petabyte-scale data for scientific analysis and visualization. After the Landsat image series were made freely available in 2008, Google consolidated this very large and useful data set and linked it to its cloud computing resources to make available to the scientific community one of the largest datasets for studying the earth's resources. GEE now includes satellite datasets from a number of other platforms, as well as many vector-based datasets.

The easily accessible and user-friendly front-end provides a convenient environment for interactive data and algorithm development. Users are also able to add and curate their own data and collections, while using Google's cloud resources to undertake all the processing. The end result is that this now allows scientists, independent researchers, hobbyists and nations to mine this massive warehouse of data for change detection, map trends and quantify resources on the Earth's surface like never before. One does not need large processing powers of the latest computers or the latest software, meaning that resource poor researchers in the poorest nations of the world have the same ability to undertake analysis as those in the most advanced nations.

Applications of GEE include, but are not limited to: mapping forest cover, detecting deforestation, classifying land cover, estimating forest biomass and carbon, mapping urban area expansion, population mapping, changes in agricultural production and forecasting, and

rangeland dynamics.

This collection calls for example applications of GEE all over the world and in all disciplines. We particularly encourage articles from developing nations on how the availability of GEE data and processing has enabled new research that was difficult or impossible before. We also encourage papers on issues about using GEE, processing shortcomings, programming, and difficulties in handling data in the cloud atmosphere. Anything to do with GEE is suitable for this collection.

The Earth Engine API is available in Python and JavaScript, making it easy to harness the power of Google's cloud for your own geospatial analysis.

## 2.2 Normalised Burnt Ratio

The Normalized Burn Ratio (NBR) was designed to highlight burned areas and estimate fire severity. The formula is similar to NDVI, except that it uses near-infrared (NIR) and shortwave-infrared (SWIR) wavelengths.

$$NBR = \frac{NIR - SWIR}{NIR + SWIR} \quad (1)$$

Pre-fire, healthy vegetation has very high near-infrared reflectance and low reflectance in the shortwave infrared portion of the spectrum. Recently burned areas on the other hand have relatively low reflectance in the near-infrared and high reflectance in the shortwave infrared band. A high NBR value generally indicates healthy vegetation while a low value indicates bare ground and recently burned areas.

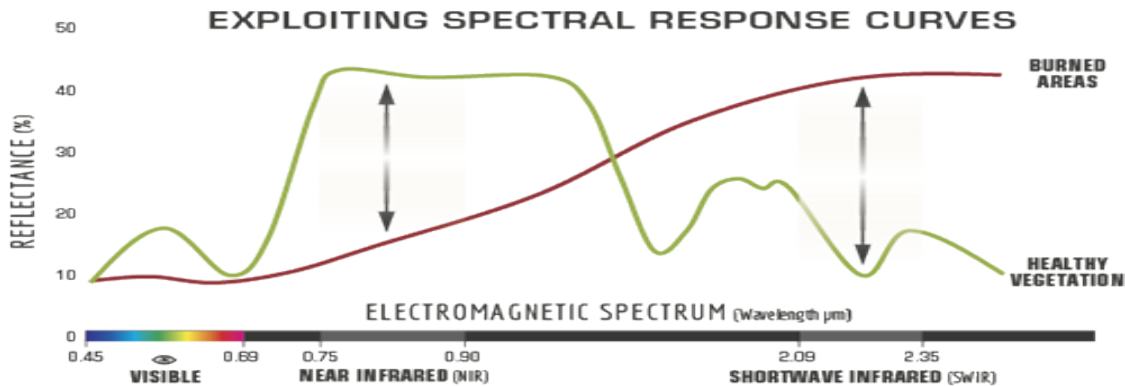


Fig. 1: Image Credit: US Forest Service

## 2.3 Burnt Severity

Normalized Burn Ratio is frequently used to estimate burn severity. Imagery collected before a fire will have very high near infrared band values and very low mid infrared band values and a Imagery collected over a forest after a fire will have very low near infrared band values and very high mid infrared band values. Higher dNBR indicate more severe damage. Areas with negative dNBR values may indicate increased vegetation productivity following a fire.

$$dNBR(or)\Delta NBR = PrefireNBR - PostfireNBR \quad (2)$$

The meaning of the NBR values can vary by scene, and for best results interpretation in specific instances should always be based on some field assessment. However, the adjacent table from the USGS FireMon program can be useful as a first approximation for interpreting the NBR difference. These guidelines are then used to create a thematic burn severity layer depicting severity as unburned to low, low, moderate, high, and increased greenness (increased post-fire vegetation response).

Typically,  $NBR$  and  $\Delta NBR$  images are generated shortly after a fire burns to get an initial assessment of burn severity and to support field work. During the next growing season, NBR datasets are often calculated again to assess vegetation survival and delayed mortality. Burn severity data and maps can aid in developing emergency rehabilitation and restoration plan post-fire. They can be used to estimate soil burn severity and to estimate the likely future downstream impacts due to flooding, landslides, and soil erosion.

<b><math>\Delta NBR</math></b>	<b>Burn Severity</b>
< -0.25	High post-fire regrowth
-0.25 to -0.1	Low post-fire regrowth
-0.1 to +0.1	Unburned
0.1 to 0.27	Low-severity burn
0.27 to 0.44	Moderate-low severity burn
0.44 to 0.66	Moderate-high severity burn
> 0.66	High-severity burn

Fig. 2: Classification based on NBR values

When uses indices as part of an analysis it is important to make sure data has been appropriately preprocessed. This usually involves radiometric calibration and atmospheric correction. To get the most information out of a spectral index it is optimal to collect ground truth data to calibrate and validate the index.

### 3 Dataset

#### 3.1 Satellite Data

We tried the following procedure for 2 satellite datas and two datasets

- Sentinel-2 MSI: MultiSpectral Instrument, Level-1C

This is a wide-swath, high-resolution, multi-spectral imaging mission supporting Copernicus Land Monitoring studies, including the monitoring of vegetation, soil and water cover, as well as observation of inland waterways and coastal areas.

The Sentinel-2 data contain 13 UTM16 spectral bands representing TOA reflectance scaled by 10000. In addition, three QA bands are present where one (QA60) is a bitmask band with cloud mask information.



Fig. 3: Landsat 8 Image

- USGS Landsat 8 Surface Reflectance Tier 1

This dataset is the atmospherically corrected surface reflectance from the Landsat 8 OLI/TIRS sensors. These images contain 5 visible and near-infrared (VNIR) bands and 2 short-wave infrared (SWIR) bands processed to orthorectified surface reflectance, and two thermal infrared (TIR) bands processed to orthorectified brightness temperature.

These data have been atmospherically corrected using LaSRC and includes a cloud, shadow, water and snow mask produced using CFMASK, as well as a per-pixel saturation mask. Strips of collected data are packaged into overlapping "scenes" covering approximately 170km x 183km using a standardized reference grid.



Fig. 4: Landsat 8 Image

### 3.2 Locations

**Bandipur** On 21 February 2019, wildfire broke out in the Bandipur Tiger Reserve. Unlike in previous years, this is the first time the wildfire in Bandipur flared up earlier due to the sudden climatic change and rapid growth of dry grass and Lantana. Officials reported that there appeared to have been no deaths of the larger mammals in the Park, such as bison, elephants, leopard and tiger.

Over 10,000 acres of forest in Bandipur area was destroyed. With the fire spreading rapidly due to winds, the authorities closed the Gundlupet-Ooty National highway and safaris in Bandipur National Park were also cancelled. Strong winds were making the job of firefighters, forest staff and volunteers more difficult.

The fire that destroyed Kundakere Range, spread to Barakatte and Guddakere, then to the Himavathi Gopalaswamy Hills. It also destroyed forests in Jarkal Kere and Gowri Kalu hills. Many smaller mammals and reptiles were killed, along with thousands of trees. There had been no forest-fire incidents noted in Bandipur National Park in the previous two years.[7]



Fig. 5: Bandipur Fire

## 4 Methodology

### 4.1 Opening and running code in the Code Editor

The steps below demonstrate how to open Earth Engine and execute a custom script that displays an image.

- Open the Earth Engine Code Editor here: [code.earthengine.google.com](https://code.earthengine.google.com). If you have not already, you will need to enable access by logging in using a registered Google account.
- Navigate to the Scripts tab located on the far left of the Code Editor. There you will find a collection of example scripts that access, display, and analyze Earth Engine data.
- Under “Image Collection,” select the “Filtered Composite” example. You will see a script appear in the center console. Press the Run button to execute the script. The Filtered Composite example selects Landsat 7 images that intersect or are within the boundaries of Colorado and Utah. It then displays a true color composite of the selected images. The samples introduce you to commonly used methods, such as `filter()`, `clip()`, and `Map.addLayer()`.

### 4.2 Importing Geometry

We can either draw a geometry or import a geometry using asset.

```
1 var geometry = Polygon, 4 vertices
```

Listing 1.1: draw geometry

OR

```
1 var Table = /Path to asset/
```

Listing 1.2: import geometry

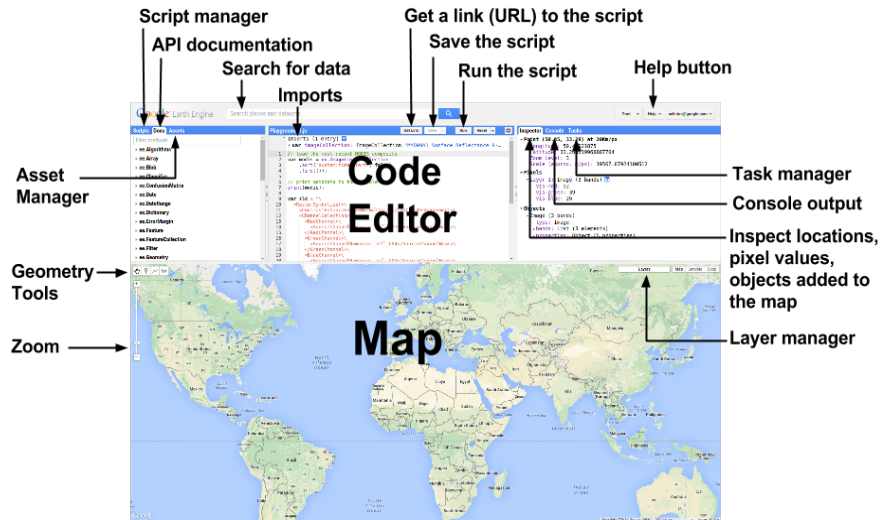


Fig. 6: Earth Engine Code

### 4.3 Importing Image or Image Collection

```
1 var prefire_image = ee.Image( 'LANDSAT/LC08/C01/T1_SR/LC08_145051_20161224 ' );
2 var postfire_image = ee.Image( 'LANDSAT/LC08/C01/T1_SR/LC08_145051_20170330 ' );
```

Listing 1.3: import single image

OR

```
1 var prefire = ee.ImageCollection( 'COPERNICUS/S2 ' )
2   .filterDate( '2019-02-21', '2019-02-22' )
3   .filterBounds(geometry);
4
5 var postfire = ee.ImageCollection( 'COPERNICUS/S2 ' )
6   .filterDate( '2019-03-08', '2019-03-09' )
7   .filterBounds(geometry);
```

Listing 1.4: import geometry

In the above listing, we imported the sentinel 2 dataset from the specified dates and the specific bounds, which is our study area. The dates are set to the prefire dates and suitable interval is given so that image can be taken.

### 4.4 Mapping a layer to the Map

Including the following code add the layer on the map and you can see it in the map displayed below the code editor.

```
1 Map.addLayer(prefire_image,{bands: [ 'B4', 'B3', 'B2'], max:4000}, 'prefire_image');
```

Listing 1.5: Map Layer

### 4.5 Exporting and image

Download the generated tiff file and export to the google drive. The file can be downloaded from the drive,

```

1 Export.image.toDrive({
2   image: prefire_image.visualize(),
3   description: 'prefire_sentinel',
4   scale: 30
5 });

```

Listing 1.6: Export to Drive

## 4.6 Compute NBR

The normalised difference can be computed as follows:

```

1 //prefire NBR
2 var pre_nbr = prefire_image.normalizedDifference(['B5', 'B7']).rename('NBR');
3 var prefire_nbr = prefire_image.addBands(pre_nbr);
4
5 //postfire NBR
6 var post_nbr = postfire_image.normalizedDifference(['B5', 'B7']).rename('NBR');
7 var postfire_nbr = postfire_image.addBands(post_nbr);

```

Listing 1.7: Calculate NBR

## 4.7 Compute dNBR

The difference normalised difference can be computed as follows:

```

1 var delta = prefire_nbr.select('NBR').subtract(postfire_nbr.select('NBR'));
2 var dNBR = delta.multiply(1000);

```

Listing 1.8: Calculate dNBR

## 4.8 Classify the Map

The difference classes based on the dNBR value can be classified using colors to indicate the severity of the burn.

```

1 //map classified using color classifiers
2 var sld_intervals =
3   '<RasterSymbolizer>' +
4   '<ColorMap type="intervals" extended="false" >' +
5     '<ColorMapEntry color="#ffffff" quantity="-500" label="-500"/>' +
6     '<ColorMapEntry color="#7a8737" quantity="-250" label="-250" />' +
7     '<ColorMapEntry color="#acbe4d" quantity="-100" label="-100" />' +
8     '<ColorMapEntry color="#0ae042" quantity="100" label="100" />' +
9     '<ColorMapEntry color="#fff70b" quantity="270" label="270" />' +
10    '<ColorMapEntry color="#ffaf38" quantity="440" label="440" />' +
11    '<ColorMapEntry color="#ff641b" quantity="660" label="660" />' +
12    '<ColorMapEntry color="#a41fd6" quantity="2000" label="2000" />' +
13    '</ColorMap>' +
14    '</RasterSymbolizer>';
15
16 Map.addLayer(dNBR.sldStyle(sld_intervals), {}, 'dNBR classified');

```

Listing 1.9: Calculate dNBR



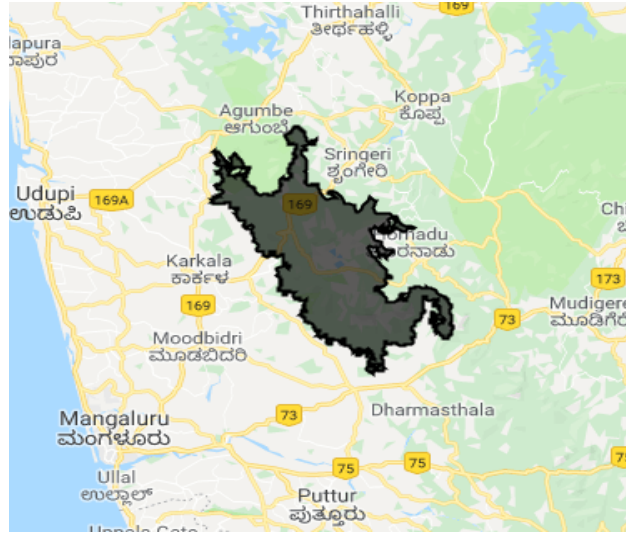


Fig. 7: Geometry of the region



Fig. 8: Prefire RGB image

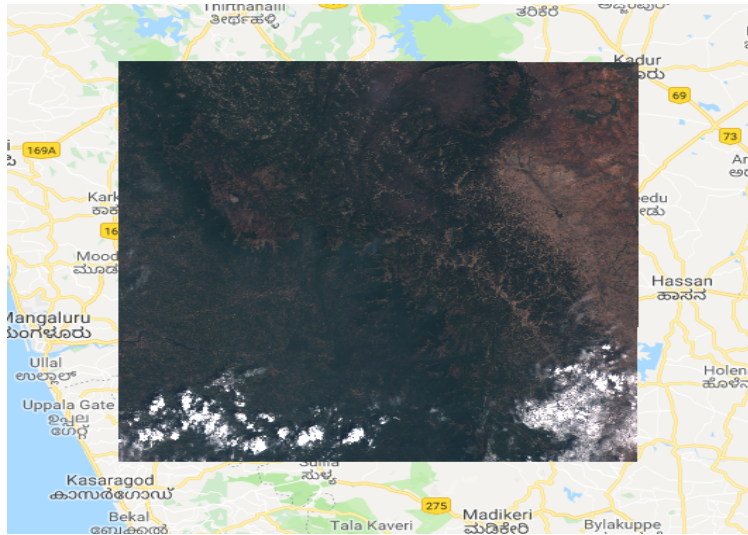


Fig. 9: Postfire Image

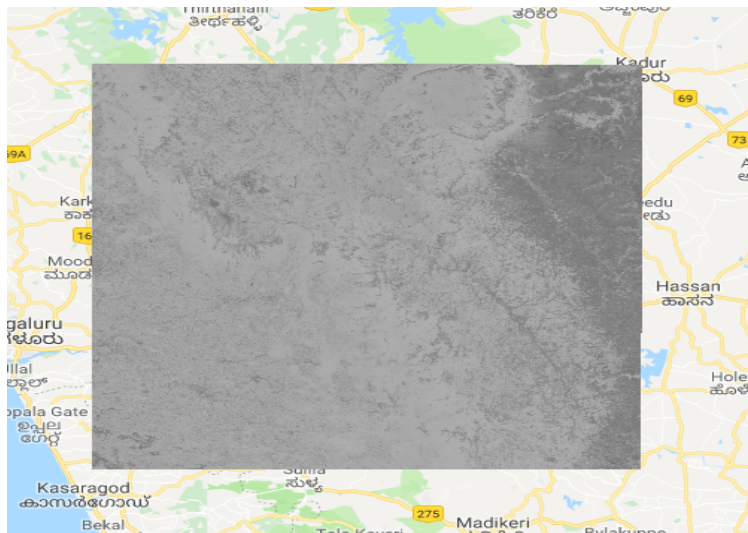


Fig. 10: Prefire NBR



Fig. 11: Postfire NBR

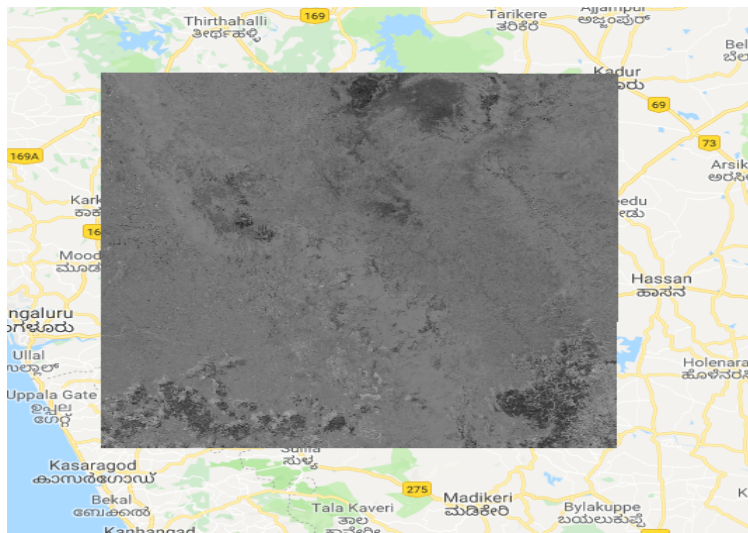


Fig. 12: dNBR unclassified

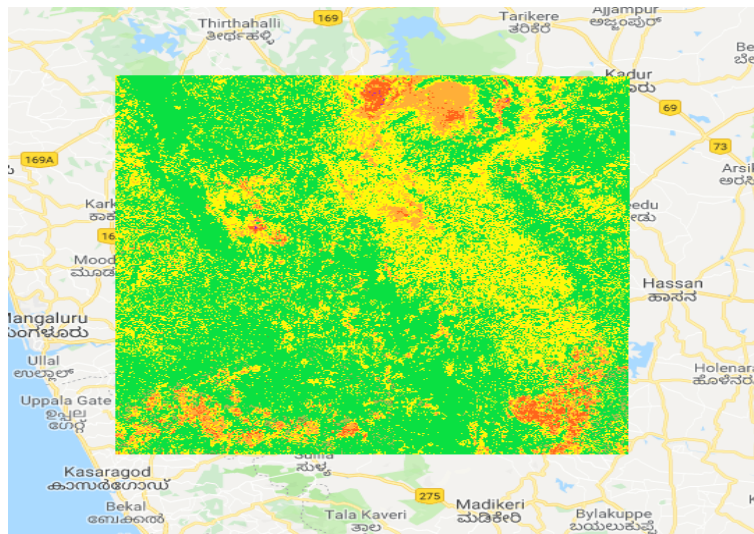


Fig. 13: dNBR classified

### dNBR Classes

- Enhanced Regrowth, High
- Enhanced Regrowth, Low
- Unburned
- Low Severity
- Moderate-low Severity
- Moderate-high Severity
- High Severity
- NA

Fig. 14: Classes made for segregation

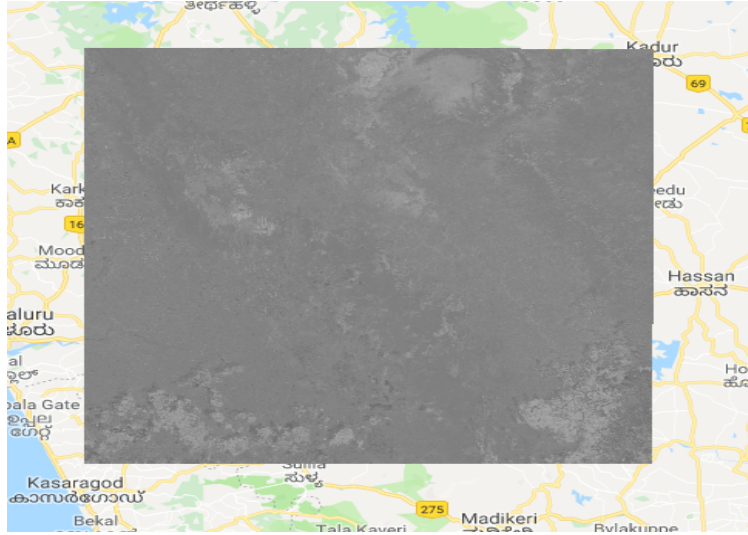


Fig. 15: the dnbr grayscale original single band image

## 4.9 Results and Inference

The above results are generated when the code is run. These images were exported and then checked using software and confirmed with the available data for firepoints. It was found that the firepoints matched exactly with the locations that were reported as burnt by the dnbr classification.

Very few of the points were shifted due to the image selected and the dates chosen as the fire might have shifted in that span from one point to another.

## 4.10 Conclusion

Earth Engine is not like any GIS you have ever used before! Because Earth Engine is a cloud-based platform, it is incredibly powerful. It is also strange and wondrous. The purpose of the docs in the How Earth Engine works section is to demystify some potentially surprising behavior you may encounter when running scripts in the Code Editor. This includes:

Distinguishing between JavaScript objects in the client and Earth Engine objects on the server. The lazy computation model. How Earth Engine handles scale (pixel resolution). How Earth Engine handles map projections.

## 4.11 Acknowledgement

I would like to acknowledge Ms. Amba Shetty for guiding me throughout the course of the project. Also, I would like to thank her for being a constant source of motivation for me to continue with this topic, which was a challenging and hard task. I would also like to thank our faculty advisor, Ms. B R Jayalekshmi, for being supportive in proceeding with the course.

## 5 References



1. <https://developers.google.com/earth-engine/>
2. [https://servirglobal.net/Portals/0/Documents/Articles/ChangeDetectionTraining/Module2\\_Intro\\_Google\\_Earth\\_Engine\\_Exercise.pdf](https://servirglobal.net/Portals/0/Documents/Articles/ChangeDetectionTraining/Module2_Intro_Google_Earth_Engine_Exercise.pdf)
3. [http://gsp.humboldt.edu/olm\\_2015/Courses/GSP\\_216\\_Online/lesson5-1/NBR.html](http://gsp.humboldt.edu/olm_2015/Courses/GSP_216_Online/lesson5-1/NBR.html)
4. Bioucas-Dias, José M., et al. "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches." *IEEE journal of selected topics in applied earth observations and remote sensing* 5.2 (2012): 354-379.
5. Parra, Lucas C., Clay Spence, Paul Sajda, Andreas Ziehe, and Klaus-Robert Müller. "Unmixing hyperspectral data." In *Advances in neural information processing systems*, pp. 942-948. 2000.
6. Drumetz, Lucas, et al. "Blind hyperspectral unmixing using an extended linear mixing model to address spectral variability." *IEEE Transactions on Image Processing* 25.8 (2016): 3890-3905.
7. Code Source for the greedy sparse approximation algorithm: (<http://staffhome.ecm.uwa.edu.au/~00053650/code.html>)
8. Sources: <http://openremotesensing.net/knowledgebase/code-and-demo-for-hyperspectral-unmixing-2>
9. Sources: <http://openremotesensing.net/?s=Hyperspectral+Unmixing+Overview%3A+Geometrical%2C+Statistical%2C+and+Sparse+Regression-Based+Approachesposttype=>