Disaster Damage Assessment using Deep Learning and Hyperspectral Imaging

Abhishek Batra
Department of Electronics and Communication
Bharati Vidyapeeth's College of Engineering
New Delhi, India

Adhaar Sharma
Department of Electronics and Communication
Bharati Vidyapeeth's College of Engineering
New Delhi, India

Isha Singh
Department of Electronics and Communication
Bharati Vidyapeeth's College of Engineering
New Delhi,India

Sang Pri Singh
Department of Electronics and Communication
Bharati Vidyapeeth's College of Engineering
New Delhi, India

***Abstract*---Every year earthquakes becomes the cause of death for thousands of people, as within few seconds it make buildings to shatter and leave no time for them to react. We humans cannot stop such natural calamities, but in this paper we are trying to prioritize the relief distribution strategies to the worsely affected areas. For that we are using deep learning models to analyze the effects on the land**

Keywords – segments, classification, seismic Hyperspectral, calamities.

1. Introduction

Earthquakes accounted for over 60% of all natural disaster-related deaths from 2001 to 2011. They are mainly caused by the passage of seismic waves through Earth's rocks. That is why analysis of post disaster damage is pivotal to strategize how to recover from it and deal with similar disasters in future.

Traditionally used damage assessment maps created by survey volunteers tend to be time intensive and inaccurate. Using deep learning models with advanced hyperspectral imaging will be invaluable to the volunteers to analyze and assess damage, thus, making the process much more efficient. Our goal is to create a model that could more quickly and accurately identify hardest hit areas and prioritize better relief distribution strategies.

We will learn how to access satellite data and process it for our models. Our focus would be to try to get sufficient data and analyze recently hit Amphan Cyclone in West Bengal. Data which we would be using would be Landsat9, NAIP and Sentinel.

The deep learning techniques that we would be using are Random Forest and U-Net. We will constantly read more and more about other deep learning techniques and compare their efficiency compared to models which we are using. In the end, a heat map can be drawn as per the result to visualize the damage.

2. Related Work

Over the years more and more people have come up with ways to find out ways to save people from suc natural disasters and for same *Austin J. Cooner* at al[2] introduced a method of multilayer feedforward neural networks, radial basis neural networks, and Random forests for detecting the earthquake damage

J. Adams, Charles at al[1] tried to develop techniques for post-earthquake urban damage detection, based on the comparative analysis of remote sensing images acquired before and after the event, and to develop the technological infrastructure for integrating these techniques into field-based reconnaissance activities.

This paper proposes the requirements of a regional damage scale for measuring the effects of earthquakes, floods and windstorms using high-resolution optical and synthetic aperture radar (SAR) data. The basis for this scale comes from recommendations made at two EERI-MCEER-UCI sponsored workshops (2003, 2004) where the focus was on the application of remote sensing technologies for disaster response and recovery. Many of the participants in these two workshops are current members of EERI's Subcommittee on Remote Sensing.

3. METHODOLOGY

In this section of our paper we will walk through segmenting and classifying high resolution imagery using Python.

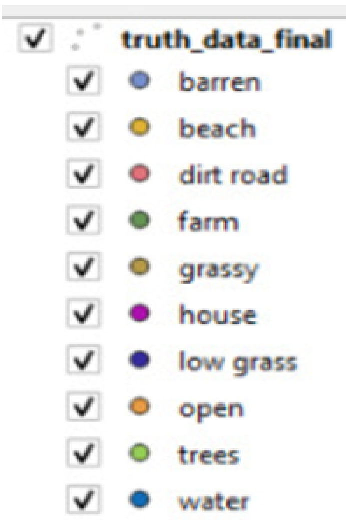
Image segmentation

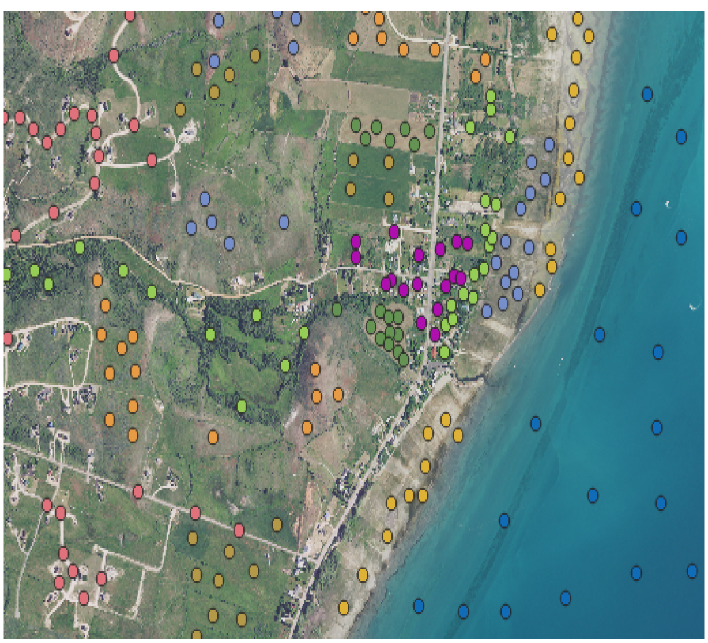
For segmenting and grouping the high resolution imagery we are using the skimage (scikit-image) library. First we read data from the NAIP image into python using gdal and numpy. For that we need to create gdal Dataset with gdal.Open().

Now after reading data from each of the four bands in the NAIP image(red, green, blue and near- infrared), rescale the image value so they are between zero and one. Now we can do segmentation.

Truth Data

Below are the Data points which were manually plotted and labelled

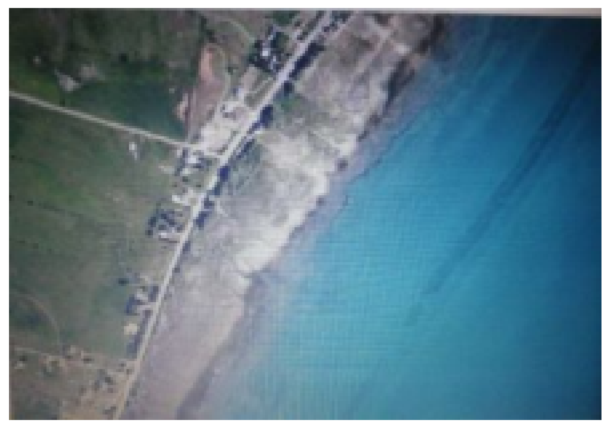




I. Quickshift Segmentation

In this segmentation the segments generated from Quickshift's defaults were too large to accurately represent the roads in the NAIP image therefore we hanged the algorithm in order to get more appropriate segmentation result.

But alas changing the ratio parameters did not produce segments different from the default one. For capturing details in the image we needed smaller segments so we decreased the default value of max_dist parameters, which controls the output size of the segment, from 10 to smaller ones. Below are some segmentation results.



Before Segmentation

After Segmentation

II. SLIC segmentation

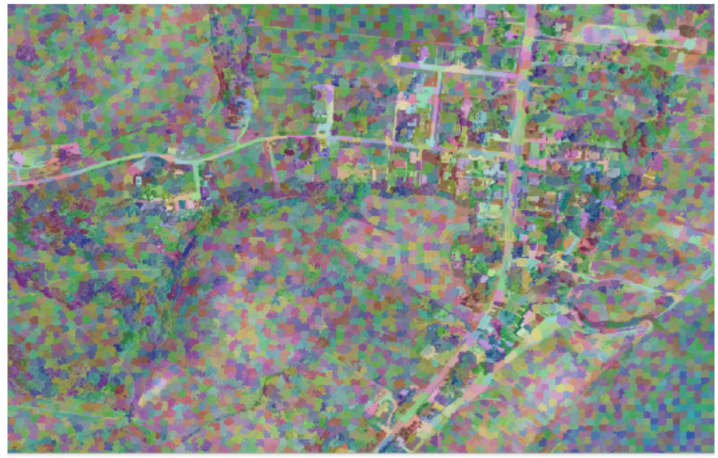
SLIC (Simple Linear Iterative Clustering) knots the pixels into the combined five-dimensional color like min, max, mean, variance, skewness and image plane space to efficiently generate compact and nearly uniform superpixels.

This segmentation is largely controlled by the approximate number of segments (n_segments) and shape (compactness) of the segments. As you can see from the image SLIC seems to do a better job at capturing features in this image than Quickshift. Now we need to describe each segment based on it's spectral properties because the spectral properties are the variables that will classify each segment as a land cover type. First of all, we wrote a function, for that we were given an array of pixel values, which will calculate the min, max, mean, variance, skewness, and kurtosis for each band. Then we got the pixel data and saved the returned features.

In object-based image analysis each segment represents an object. Objects represent buildings, roads, trees, fields or pieces of those features, depending on how the segmentation is done. Then sets up a list for the statistics describing each object (i.e. segment ID) returned from the segment_features function. The pixels for each segment are identified and passed to segment_features, which returns the statistics describing the spectral properties of the segment/object.

On increasing the number of segments and decreasing the compactness resulted in very elongated segments, but unable to match the image features very well (compactness) of the segments. Therefore, we started with 100,000 segments because the Quickshift algorithm was producing 50,000-75,000 segments and not capturing. Below is the best segmentation which we created with the limited runs

1.) When the segmentation is overlaid on the satellite image with transparency.



2.) When Zoomed - in the version of the overlay

By observing the image one can see, how important is segmentation for creating accurate classification results.

III Training And Test Dataset:

This is a supervised classification workflow, so we'll need to have some truth data describing the land cover types The code below uses GEOPANDAS to read the truth data as a EODATAFRAMES. Randomly, 70% of the truth observations are assigned to a training data set and the remaining 30% to a testing data set. The training and test data sets are each saved to a new shapefile.

IV Land Cover Classification

This is the meat of the analysis. The classification algorithm. First,identify and label the training objects (lines 1–20). This process involves associating a label (landcover type) with the statistics describing each spectral band within the image segment. Now,everything is set up to train a classifier and use it to predict across all segments in the image.

Here we are using random forests, a popular classification algorithm. The random forest is a tree based classifier and consists of a combination of tree classifiers where each classifier is generated using a random vector sampled independently from the input vector, and each tree casts a unit vote for the most popular class to classify an input vector.

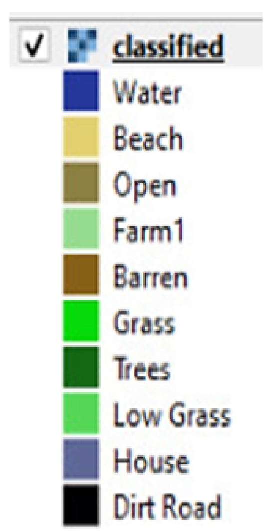
The code to train (fit) the algorithm and make predictions is quite simple (lines 22–24). Simply pass the training objects (containing the spectral properties) and the associated land cover label to the classifier. classifier is trained (fitted) predictions can be made for non-training segments based on their spectral properties. After the predictions are made, save them to raster for display in a GIS (lines 26–43).

V CONCLUSION

In this paper we did Geographic Object-Based Image Analysis (GeOBIA). For that first we segmented the image captured using the SLIC and Quickshift segmentation. On performing both we found out that SLIC segmentation was found to capture more details than the quick shift algorithm. Then we classified our segmented images into land cover categories, using the Random Forests algorithm.

But still we have to work on the accuracy assessment with a confusion matrix. As accuracy assessment is a crucial aspect of any classification. If your classification doesn't represent what it's supposed to, it's not worth much. The represented in your classification. We generated the points in QGIS to represent seven different land cover classes. We had collected data in a more organized and statistically rigorous manner. The land cover truth data needs to be split into training and test data sets. The training data set will train the Random Forests classification algorithm. We will compare the classification results to the test data set to assess classification accuracy.

Test dataset which we had created earlier now we convert it to the raster format so it is compatible with the generated predictions. Then simply query the predicted values from the locations where test data exist. Finally, generate the confusion matrix from the corresponding values



VI Accuracy Results

Accuracy can be determined by dividing the diagonal element with the sum of all elements in a particular column.

$$accuracy = \frac{cm.diagonal(\quad)}{cm.sum(axis = 0)}$$

Accuracy formula

```
[7 3 3 6 4 6 7 5 4 6]
[7 6 6 8 5 6 9 6 4 8]
[1.         0.5        0.5        0.75       0.8        1.
 0.77777778 0.83333333 1.         0.75      ]
```

Accuracy Result

VII Accuracy Assessment

A confusion matrix is a table that is often used to describe the performance of the classification model on a set of test data for which the true values are known. It is also used to assess the accuracy of the model

PREDICTED

ACTUAL

	WATER	BEACH	OPEN	FARM	BARREN	GRASSY	TREES	LOW-GRASS	HOUSE	DIRT-ROAD
WATER	7	0	0	0	0	0	0	0	0	0
BEACH	0	3	1	0	0	0	0	0	0	1
OPEN	0	2	3	1	1	0	1	1	0	0
FARM	0	0	0	6	0	0	0	0	0	0
BARREN	0	1	1	1	4	0	0	0	0	0
GRASSY	0	0	1	0	0	6	0	0	0	0
TREES	0	0	0	0	0	0	7	0	0	0
LOW-GRASS	0	0	0	0	0	0	0	5	0	0
HOUSE	0	0	0	0	0	0	1	0	4	1
DIRT-ROAD	0	0	0	0	0	0	0	0	0	6

Confusion Matrix

REFERENCES

[1] *J. Adams, Charles K. Huyck(2003) "*Application of High-Resolution Optical Satellite Imagery for Post-Earthquake Damage Assessment"

[2] Austin J. (2010)Detection of Urban Damage Using Remote Sensing and Machine Learning Algorithms:Revisiting the 2010 Haiti Earthquake

[3] S. Avidan and A. Shamir, "Seam Carving for Content-Aware Image Resizing," ACM Trans. Graphics, vol. 26, no. 3, Article 10, 2007.

[4] A. Ayvaci and S. Soatto, "Motion Segmentation with Occlusions on the Superpixel Graph," Proc. Workshop Dynamical Vision, Oct. 2009. [5] Y. Boykov and M. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images," Proc. IEEE Int'l Conf. Computer Vision, 2001.

[6] Dong, L.; Shan, J.(2013) A Comprehensive review of earthquake-induced building damage detection with r
emote sensing techniques.

[7] Ural, S.; Hussain, E.; Kim, K.; Fu, C.; Shan, J. (2011)Building extraction and rubble mapping for city Port-au-Prince post-2010 earthquake with GeoEye-1 imagery and Lidar Data.

[8] BREIMAN, L., 1999, Random forests—random features. Technical Report 567, Statistics Department, University of California, Berkeley..