A Report

ON

Change Detection in Land Usage using Temporal Data from Remote Sensing Imagery

Prepared by

Shaily Bhatt	2017A7PS0040P
Meet Kanani	2017A7PS0128P

AT

Bhaskaracharya Institute for Space Applications and Geo-Informatics, Gandhinagar

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
(JULY 2019)

Change Detection in Land Usage using Temporal Data from Remote Sensing Imagery

Prepared by

Shaily Bhatt	2017A7PS0040P	Computer Science & Engineering
Meet Kanani	2017A7PS0128P	Computer Science & Engineering

Prepared in fulfilment of the
Practice School-I Course NO.
BITS F221/BITS F231/ BITS F241

AT

Bhaskaracharya Institute for Space Applications and Geo-Informatics, Gandhinagar

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
(JULY 2019)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

Practice School Division

Station: Bhaskaracharya Institute of Space Applications and Geo-Informatics

Centre: Gandhinagar, Gujarat

Duration: **From:** May 21, 2019 **To:** July 12, 2019

Date of Submission: July 10, 2019

Title of the Project: **"**Change Detection in Land Usage using Temporal Data from Remote Sensing Imagery"

ID No.	Name	Discipline
2017A7PS0040P	Shaily Bhatt	Computer Science & Engineering
2017A7PS0128P	Meet Kanani	Computer Science & Engineering

Name of expert: Dr. Manoj Pandya **Designation:** Senior Project Manager

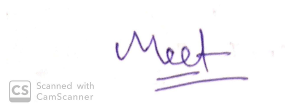
Name of the PS Faculty: Dr. Pratik N. Sheth

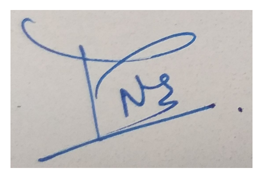
Keywords: Change Detection, Land Usage, Structural Similarity, Temporal data, Sentinel, OpenCV, Image Processing, Texture Recognition, Texture Detection, Local Binary Pattern, Machine Learning, Python, Remote Sensing Imagery

Project Area(s): Image Processing and Machine Learning on Remote Sensing Imagery for Texture Detection and Recognition and Change Detection in Land Usage.

Abstract:

The following report summarises our work on the project "Change Detection in Land Usage using Temporal Data from Remote Sensing Imagery". The report illustrates the use of the 'Structural Similarity Index' for change detection. It details data mining from Sentinel hub for real-time temporal data. This is followed by data pre-processing and the application of the algorithm to detect changes. Finally, we describe the web-app developed to simplify the use of the model for the end-user.





Signatures of Students

Date: 12/07/2019

Signature of PS Faculty

Date:12/07/2019

Acknowledgement

We wish to express our heartfelt gratitude to **Dr. Manoj Pandya** for assigning us this project work and for his guidance and support throughout the project.

We would also like to acknowledge **BISAG, Gandhinagar** for providing us with the opportunity of working on this interesting project.

Our heartfelt thanks are due to instructor-in-charge, **Dr. Pratik N. Sheth**, for his invaluable guidance. We acknowledge with thanks the support rendered by the **PS Division, BITS Pilani** in giving us this exposure through Practice School.

<u>CANDIDATES' DECLARATION</u>

We declare that project report on **"Texture Recognition and Classification in Remote Sensing Imagery"** And "**Change Detection in Land Usage using Temporal Data from Remote Sensing Imagery**" is our own work conducted under the supervision of **Dr. Manoj Pandya** from BISAG (Bhaskaracharya Institute for Space Applications & Geo-informatics).We further declare that to the best of our knowledge the report for Practice School Division, BITS Pilani does not contain any work of other authors without proper citation.

Shaily Bhatt	Meet Kanani
2017A7PS0040P	2017A7PS0128P

Submitted To:

Dr. Pratik N. Sheth	Dr. Manoj Pandya
PS Faculty	Project Mentor
Practice School-I (July 2019)	Senior Project Manager
PS Division, BITS Pilani	BISAG, Gandhinaga

Bhaskaracharya Institute for Space Applications and Geo-informatics
Department of Science & Technology
Government of Gujarat
Phone: 079 - 23213081 Fax: 079-23213091
E-mail: info@bisag.gujarat.gov.in, website: www.bisag.gujarat.gov.in

CERTIFICATE

This is to certify that the project report compiled by ***Ms. Shaily Bhatt*** and ***Mr. Meet Kanani*** students of second year at ***BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI*** have completed their ***PRACTICE SCHOOL-I project*** satisfactorily. To the best of our knowledge this is an original and bonafide work done by them. They have worked on ***"Change Detection in Land Usage using Temporal Data from Remote Sensing Imagery"***, starting from ***May 21, 2019 to July 12, 2019.***

During their tenure at this Institute, they were found to be sincere and meticulous in their work. We appreciate their enthusiasm & dedication towards the work assigned to them. We wish them every success.

Manoj Pandya
Senior Project Manager
BISAG, Gandhinagar

T. P. Singh
Director
BISAG, Gandhinagar

INDEX

1. About BISAG, Gandhinagar

1.1. About the Institute

Modern day planning for inclusive development and growth calls for transparent, efficient, effective, responsive and low cost decision making systems involving multi-disciplinary information such that it not only encourages people's participation, ensuring equitable development but also takes into account the sustainability of natural resources. The applications of space technology and Geo–informatics have contributed significantly towards the socio-economic development. Taking cognizance of the need of geo-spatial information for developmental planning and management of resources, the Department of Science and Technology, Government of Gujarat established "Bhaskaracharya Institute for Space Applications and Geo-informatics" (BISAG). BISAG is an ISO 9001:2008, ISO 27001:2005 and CMMI: 5 certified institute. BISAG which was initially set up to carryout space technology applications, has evolved into a centre of excellence, where research and innovations are combined with the requirements of users and thus acts as a value added service provider, a technology developer and as a facilitator for providing direct benefits of space technologies to the grass root level functions/functionaries.

1.2. BISAG’s Enduring Growth

Since its foundation, the Institute has experienced extensive growth in the sphere of Space technology and Geo-informatics. The objective with which BISAG was established is manifested in the extent of the services it renders to almost all departments of the State. Year after year the institute has been endeavoring to increase its outreach to disseminate the use of geo-informatics up to grassroots level. In this span of nine years, BISAG has assumed multi-dimensional roles and achieved several milestones to become an integral part of the development process of the Gujarat State.

1.3. Activities at BISAG

Satellite Communication..

for promotion and facilitation of the use of broadcast and teleconferencing networks for distant interactive training, education and extension.

Remote Sensing..

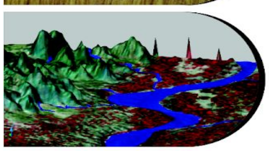
for Inventory, Mapping, Developmental planning and Monitoring of natural & man-made resources.

Geographic Information System..

for conceptualization, creation and organization of multi purpose common digital database for sectoral/integrated decision support systems.

Global Navigation Satellite System..

for Location based Services, Geo-referencing, Engineering Applications and Research.



Photogrammetry..

for Creation of Digital Elevation Model, Terrain Characteristic, Resource planning.

Cartography..

for thematic mapping, value added maps.

Software Development..

for wider usage of Geo-spatial applications, Decision Support Systems (desktop as well as web based), ERP solutions.

Education, Research and Training..

for providing Education, Research, Training & Technology Transfer to large number of students, end users & collaborators.

1.4. BISAG's Journey

2003-04	Gujarat SATCOM Network
2007-08	Centre for Geo-Informatics Applications
2010-11	Academy of Geo-informatics for Sustainable Development
2012-13	A Full Fledged Campus

1.5. Organisational Setup at BISAG

The Institute is responsible for providing information and technical support to different Departments and Organizations. The Governing Body and the Empowered Executive Committee govern the functioning of BISAG. The Institute is registered under the Societies Registration Act 1860. Considering the scope and extent of the activities of BISAG, its organizational structure has been charted out with defined functions.

The functional units include:

1. Academy of Geo-Informatics for sustainable development
2. Center for Geo-Informatics Applications
3. GUJSAT
4. Center for Informatics and Training
5. Quality Control and Standardisation cell
6. Administration

Our Project was carried out under the *Academy of Geo-Informatics for sustainable development.*

1.6. Academy of Geo-Informatics for sustainable development

Considering the requirement of high end research and development in the areas having relevance of geo-informatics technology for sustainable development, a separate infrastructure has been established. In collaboration with different institutes in the state as well as in the country, R&D activities are being carried out in the areas of climate change, environment, disaster management, natural resources management, infrastructure development, resources planning, coastal hazard and coastal zone management studies, etc. under the guidance of eminent scientists. Various innovative methodologies/models developed in this academy through the research process have helped in development of various applications. This unit also provides training to more than 600 students every year in the field of Geo-informatics to the students from various backgrounds like water resources, urban planning, computer Engineering, IT, Agriculture in the areas of Remote sensing, GIS and their applications.

2. Change Detection in Land Usage using Temporal Data from Remote Sensing Imagery

2.1. Introduction

Bhaskaracharya Institute for Space Applications and Geoinformatics (BISAG) is a State level agency by Government of Gujarat to facilitate to provide services and solutions in implementing map-based GeoSpatial Information Systems. Currently BISAG is working to implement geo-spatial technologies for the planning and developmental activities pertaining to agriculture, land and water resource management, wasteland/watershed development, forestry, disaster management, infrastructure, urban growth, land usage and education.

Automated change detection in images of acquired at different times is one of the most image processing's interesting topics. Such images are known as multi temporal images. Change detection involves the analysis of two multi temporal satellite images to find any changes that might have occurred between the two timestamps. It is one of the major utilizations of remote sensing and finds application in a wide range of tasks like defence inspections, deforestation assessment, land use analysis, disaster assessment and monitoring many other environmental/man-made changes.

Our project "Change Detection in Land Usage using Temporal Data from Remote Sensing Imagery" aims to develop sophisticated computational techniques that can monitor the changes in marked plots of land. The algorithm is than put into the backend of a user friendly web application that allows the user to detect the changes in the marked plots by selecting the duration of monitoring.

2.1.1. Project Profile

Project Title	Change Detection in Land Usage using Temporal Data from Remote Sensing Imagery
Developed for	Bhaskaracharya Institute for Space Applications and Geoinformatics (BISAG)
Description	Developing a sophisticated computational technique that can monitor the changes in marked plots of land. And to create user friendly web application that allows the user to detect the changes in the marked plots by selecting the duration of monitoring.
Type of Application	Web Application
Tools	Python, OpenCV, SentinelHub, Other python libraries and Flask

2.1.2. Project Objectives

1. Obtain Requisite data from sentinel hub for the given coordinates
2. Developing a sophisticated computational technique that can monitor the changes in plot in question
3. Create user friendly web application that allows the user to select the duration of monitoring
4. Display change maps of the images and generate an alert if changes are detected

2.2. Details of Tools Used

2.2.1. Python

Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000[100] packages with a wide range of functionality, including:

- Graphical user interfaces
- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping
- Documentation
- System administration
- Scientific computing
- Text processing
- Image processing

2.2.2 OpenCV

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. The library is cross-platform and free for use under the open-source BSD license. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. OpenCV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. OpenCV runs on the following mobile operating systems: Android, iOS, Maemo, BlackBerry 10

OpenCV's application areas include:

- 2D and 3D feature toolkits

- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion tracking
- Augmented reality

To support some of the above areas, OpenCV includes a statistical machine learning library that contains:

- Boosting
- Decision tree learning
- Gradient boosting trees
- Expectation-maximization algorithm
- k-nearest neighbor algorithm
- Naive Bayes classifier
- Artificial neural networks
- Random forest
- Support vector machine (SVM)
- Deep neural networks (DNN)

2.2.3 Sentinel Hub

Sentinel-2 is an Earth observation mission from the EU Copernicus Programme that systematically acquires optical imagery at high spatial resolution (10 m to 60 m) over land and coastal waters. The mission is a constellation with two twin satellites (Sentinel-2A and Sentinel-2B). The mission supports a broad range of services and applications such as agricultural monitoring, emergency management, land cover classification or water quality.

Sentinel Hub is a cloud based GIS platform for distribution, management and analysis of satellite data. Earth Observation imagery data is currently provided from our open data sources. The Copernicus Open Access Hub (previously known as Sentinels Scientific Data Hub) provides complete, free and open access to Sentinel-1, Sentinel-2, Sentinel-3 and Sentinel-5P user products, starting from the In-Orbit Commissioning Review (IOCR).

Sentinel Data are also available via the Copernicus Data and Information Access Services (DIAS) through several platforms .

The sentinelhub Python package allows users to make OGC (WMS and WCS) web requests to download and process satellite images within your Python scripts. It supports Sentinel-2 L1C and L2A, Sentinel-1, Landsat 8, MODIS and DEM data source.

Some of the major features are linked to one's Sentinel Hub account:

- support for Web Map Service (WMS) and Web Coverage Service (WCS) requests using your Sentinel Hub account;
- support for standard and custom multi-spectra layers, such as unprocessed bands, true color imagery, or NDVI;
- support for multi-temporal requests;
- support for cloud coverage filtering;
- support for different Coordinate Reference Systems;
- support to read and write downloaded data to disk in the most common image and data formats;
- support for various data sources:
    - Sentinel-2 L1C,
    - Sentinel-2 L2A,
    - Sentinel-1,
    - Landsat 8,
    - MODIS,

- DEM.

2.2.4 Other Python Libraries

NumPy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays

Scipy: SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.

PIL: Python Imaging Library is a free library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

Skimage: The scikit-image SciKit (toolkit for SciPy) extends scipy.ndimage to provide a versatile set of image processing routines.

OS: The *OS module* in *python* provides functions for interacting with the *operating system*.

2.2.5 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. Despite the lack of a major release, Flask has become popular among Python enthusiasts. As of mid 2016, it was the most popular Python web development framework on GitHub. Applications that use the Flask framework include Pinterest, LinkedIn, and the community web page for Flask itself

2.3. System Analysis

2.3.1 Existing system

No existing system for direct application was available for the required problem statement. However multiple APIs and libraries that could be used to obtain data and perform the image processing were available.

2.3.2 Problem Identification

1. Availability of temporal data is limited: The Google Earth engine API which can provide high quality data is paid whereas the other open source data providers do not give very highly zoomed data,
2. Algorithm for change detection: Multiple change detection algorithms are available and the best one has to be identified on the basis of results obtained by testing them.

2.4 System Design

2.4.1 Data Collection

The data to be collected id obtained from sentinel data available through sentinel hub. A python package, sentinelhub has been used for this purpose. Firstly, the coordinates of the plot are passed. Using these coordinates, a bounding box of 2km radius is calculated. Note: The radius of the bounding box to be calculated can be changed as per requirement by altering the argument to the function parameter. This bounding box is then used to shoot a WMS request to the sentinel hub.

The Web Map Service Interface Standard (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc) that can be displayed in a browser application. The interface also supports the ability to specify whether the returned images should be transparent so that layers from multiple servers can be combined or not.

The image ID number along with the coordinated and the vertices if bounding box is stored in a file. The required coordinates are obtained from this file as per ID no. Now, two WMS requests are sent using the same coordinate bounding box. Each of the requests obtains images corresponding to the time interval of the initial and final years of the duration of the monitoring. The obtained images from the WMS request are in the form of a NumPy array, whose each entry contains an image from the particular timespan. The last image from this image is used, which corresponds to the latest image in that timespan in question.

2.4.2 Data Pre-Processing

The image obtained in NumPy format is firstly converted to the OpenCV format. This is done by splitting the RGB arrays of the image and then remerging them as BGR.
Using the coordinated, the are of interest is marked and sharpened using opencv functionality. This image is then processed using change detection algorithms as described subsequently.

2.4.3 Algorithm for change detection

Once the two images have been obtained as cv2 arrays, the program proceeds to perform change detection on it. Automatic change detection in images of a region acquired at different times is one of the most interesting topics of image processing. Multiple Algorithms can be used to perform this task. These include traditional methods like MSE (Mean Squared Error) and Peak Signal to Noise Ratio (PNSR). More sophisticated and improved methods include:

1. Principal Component Analysis (PCA) and K-Means Algorithm
2. Structural Similarity Index (SSIM) Algorithm

Both of these techniques gave results with good accuracy. The implementation of both has been discussed in detail in the subsequent sections. The SSIM algorithm outperformed the PCA and K-Means Algorithm. The K-means clustering change images gave more noise in the results as compared to SSIM results. As a result, the final implemented model contains the latter.

2.4.4 Displaying the results

Once the change map has been calculated, an OpenCV function is used to draw contours for the change. The change map and the changes on the old and new image are then displayed for the user.

2.5 PCA & K-Means

2.5.1 Overview of Method

This is an unsupervised method for change detection. It involves the automatic analysis of the change data, i.e. the difference image, constructed using the multi temporal images. A difference image is the pixel-by-pixel subtraction of the 2 images.

Eigenvectors of pixel blocks from the difference image will then be extracted by Principal Component Analysis (PCA). Subsequently, a feature vector is constructed for each pixel in the difference image by projecting that pixel's neighbourhood onto the Eigenvectors. The feature vector space, which is the collection of the feature vectors for all the pixels, upon clustering by K-means algorithm gives us two clusters – one representing pixels belonging to the changed class, and other representing pixels belonging to the unchanged class. Each pixel will belong to either of the clusters and hence a change map can be generated.

So, the steps towards implementing this application are:

1. **Difference image generation and Eigenvector space (EVS)**

   The difference image has the absolute valued differences of the intensity values of the corresponding pixels of the 2 grayscale images. The computed difference image would hence be such that the values of the pixels associated with land changes will have values significantly different from those of the pixels associated with unchanged areas.

   PCA takes a data set and determines its covariance matrix after performing mean normalisation on it. The Eigenvectors and Eigenvalues of the covariance matrix are computed (giving us the EVS) and then the Eigenvectors are sorted in the descending order of Eigenvalues. This sorting step is the actual revelation of the PCA algorithm. The Eigenvectors have been sorted in decreasing order of the Eigenvalues because the Eigenvector with the highest Eigenvalue is the principal component of the data set. That vector shows along which direction the majority of the data is inclined. Thus by PCA, we have been able to extract the lines that characterise the data.

PCA is then applied on this vector set to get the Eigenvector space. The Eigen vector space will be a 25 x 25 matrix; its each column is an Eigenvector of 25 dimensions.In Python, from sklearn.decomposition, we can simply import the PCA module and use it to perform PCA on vector_set variable to get the variable EVS.

2. **Building the feature vector space (FVS)**

Building the FVS involves again taking 5 x 5 blocks from the difference image, flattening them, and lastly projecting them onto the EVS, only this time, the blocks will be overlapping. A vector space (VS) is first made by constructing one vector for each pixel of the difference image in such a way that one 5 x 5 block is actually a pixel's 5 x 5 neighborhood. It is to be noted here that by this logic, 4 boundary rows and 4 boundary columns pixels won't get any feature vectors since they won't have a 5 x 5 neighborhood. (We can manage with this exclusion of these pixels, since it is safe to assume here that any changes occurring would be concentrated in the middle regions of the images, rather than the edges). So, we will have (m x n)- 8 feature vectors in the FVS, all 25 dimensional. Projecting the FVS to the 25 dimensional EVS simply means to perform the following matrix multiplication

$$(VS)_{((m \times n - 8) \times 25)} \cdot (EVS)_{(25 \times 25)} = (FVS)_{(m \times n - 8) \times 25}$$

Figure below summarises the steps that have been followed so far in the method.

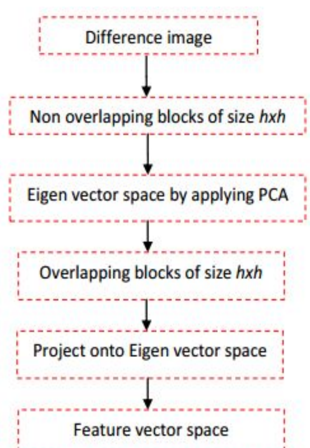


Fig: Flowchart for building the feature vector space

Source: https://i2.wp.com/appliedmachinelearning.blog/wp-content/uploads/2017/11/figure1.png?resize=247%2C335&ssl=1

The feature vectors for the pixels now lie in a space where their variance has been maximized. This will help the subsequent step of clustering to better categorize the pixels into the 2 classes – changed pixels (pix_c) and unchanged pixels (pix_u).

3. Clustering of the feature vector space and change map

The feature vectors for the pixels carry information whether the pixels have characteristics of a changed pixel or an unchanged one. Having constructed the feature vector space, we now need to cluster it so that the pixels can be grouped into two disjoint classes. The K-means algorithm will be used to do that. Thus each pixel will get assigned to a cluster in such a way that the distance between the cluster's mean vector and the pixel's feature vector is the least. Each pixel gets a label from 1 to K, which denotes the cluster number that they belong to.

The best results are obtained with K = 3. Thus the argument components in clustering() will be 3. Remember, even though we have to divide the pixels into 2 categories, we have chosen K = 3, instead of 2.

It can be postulated that the cluster which contains the lowest number of pixels (denoted by variable least_index) is the cluster denoting the changed class, since the background remains more or less the same in satellite images and the changes occurred are comparatively less. Also, the mean of this cluster will be the highest. The reason behind the highest value of mean for that cluster is that the values of the difference image pixels in a region where some changes have occurred are higher than the values of pixels in the regions where there is no change.

Thus, in conclusion, the cluster with the ***lowest number of pixels***, and also the ***highest mean*** is the cluster belonging to the changed class.

With this information, we will now build a change map – a binary image to show the output of change detection.

$$change\_map(i,j) = \begin{cases} 255, & \text{if (i, j)} \in pix_u \\ 0, & \text{otherwise} \end{cases}$$

2.5.2 System Testing and Results

<u>BISAG Plot 1:</u>

Fig: Old and New Image of Bisag plot 1

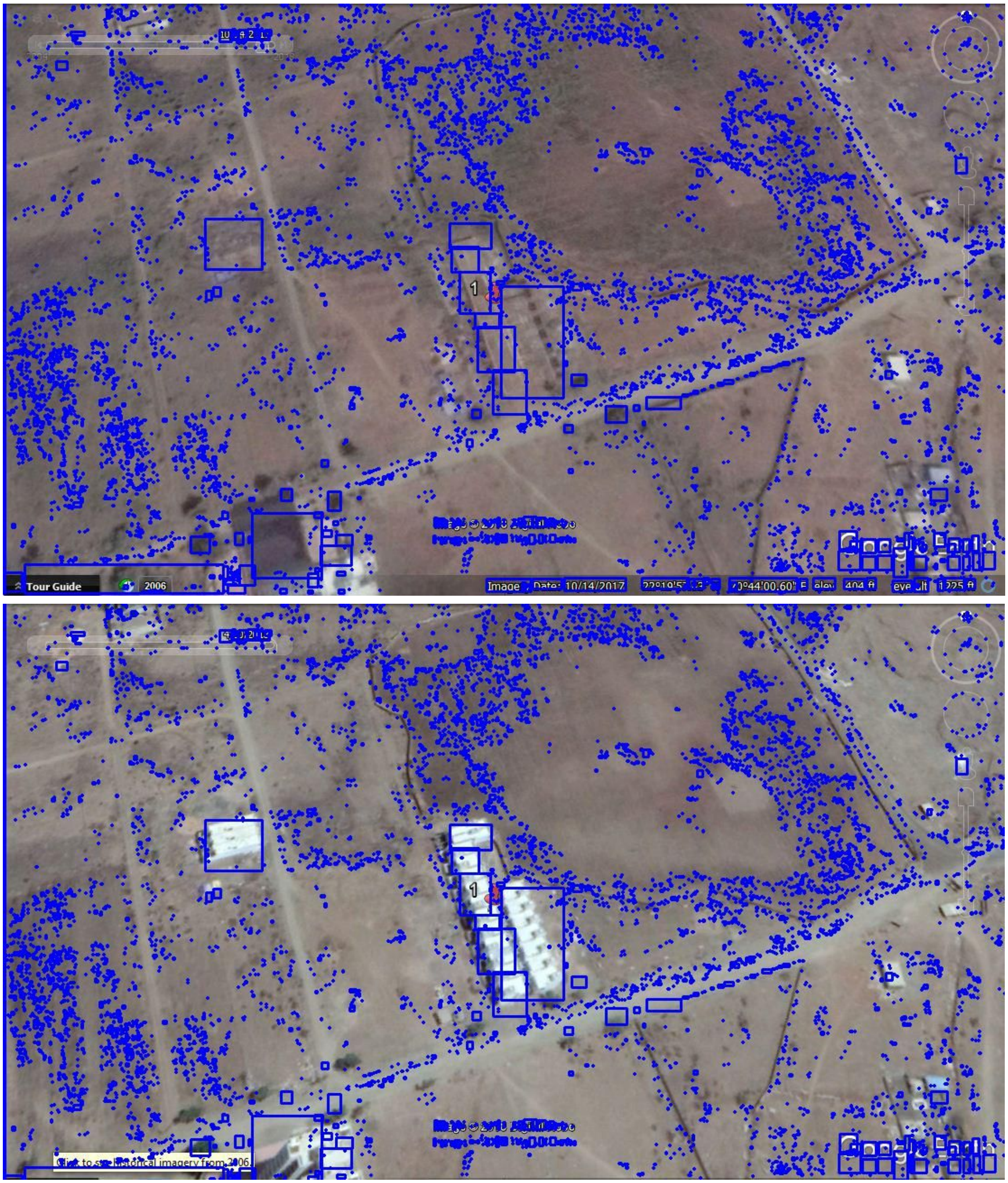


Fig: Marked Old and New Image of BISAG Plot1 after change detection

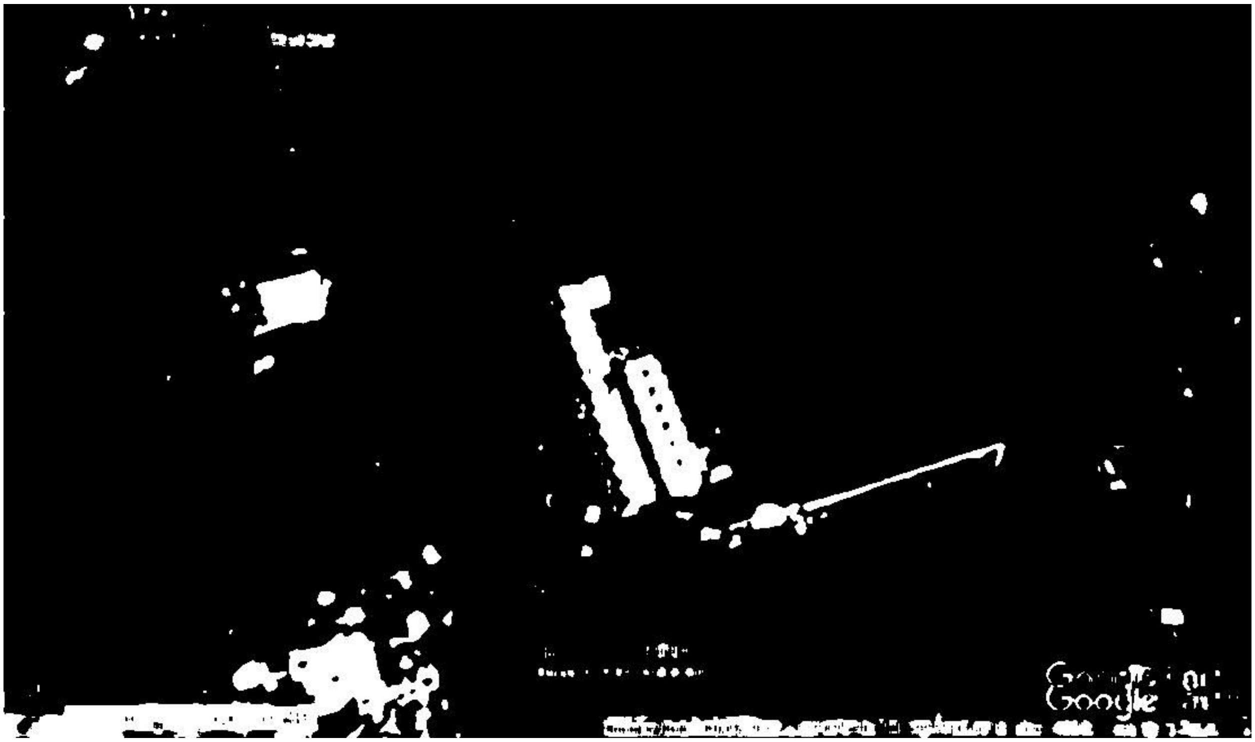


Fig: Change Map

BISAG Plot 2:

Fig: Old and New Image of Bisag plot 2

Fig: Marked Old and New Image of BISAG Plot 2 after change detection

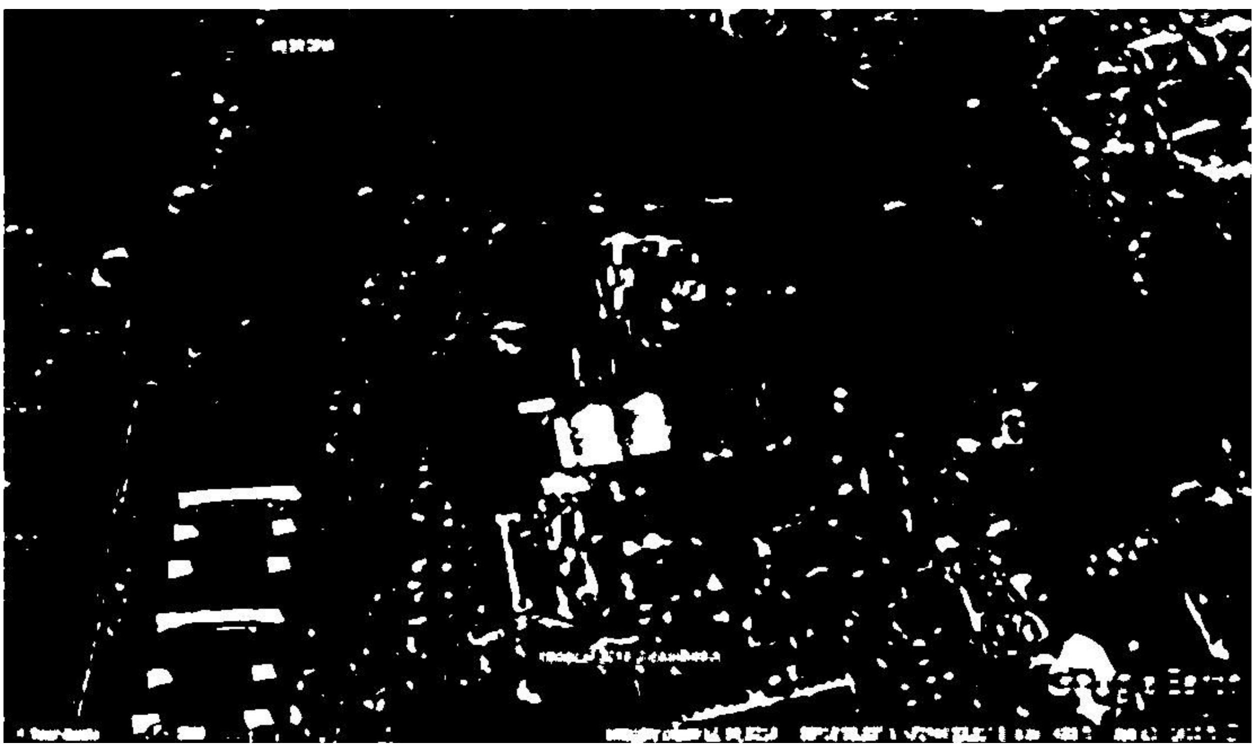


Fig: Change Map

2.5.3 Limitations and Conclusions

Though the changes detected through this method were accurate, yet, a lot of noise was obtained in the change maps. This noise refers to categorisation of unnecessary pixel differences as changes and hence its inclusion in the change map.

In order to improve upon this limitation, the SSIM technique has been employed.

2.6 Structural Similarity Index

2.6.1 Overview of Method

SSIM is used for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE).

In order to remedy some of the issues associated with MSE for image comparison, we have the Structural Similarity Index, developed by Wang et al.:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

The SSIM method is clearly more involved than the MSE method, but the gist is that SSIM attempts to model the perceived change in the structural information of the image. This leads to a more robust approach that is able to account for changes in the structure of the image, rather than just the perceived change.

The parameters to Equation include the *(x, y)* location of the *N x N* window in each image, the mean of the pixel intensities in the *x* and *y* direction, the variance of intensities in the *x* and *y* direction, along with the covariance.

This method is already implemented in the scikit-image library for image processing.

2.6.2 Process of Obtaining Change Map and Change Contours

Firstly, the program computes the Structural Similarity Index (SSIM). Using the compare_ssim function from scikit-image, a score and difference image, diff is calculated. The score represents the structural similarity index between the two input images. This value can fall into the range *[-1, 1]* with a value of 1 being a "perfect match".

The diff image contains the actual *image differences* between the two input images that we wish to visualize. The difference image is currently represented as a floating point data type in the range *[0, 1]* so we first convert the array to 8-bit unsigned integers in the range *[0, 255]* before we can further process it using OpenCV.

Next, OpenCV function is used to find the contours of the changes detected. The threshold change-map is calculated in a similar fashion as described in the PCA & K-Means Method. Finally, we show the comparison images with boxes around differences, the difference image, and the thresholded image

2.6.3 System Testing and Results

Before creating a web application, the algorithm was tested on multiple images obtained from the internet as well as few images obtained from BISAG.

In order to test the code, firstly, all the required packages need to be installed. This includes installing NumPy, Scikit-learn, PIL, OpenCV, SentineHub etc. The system testing was performed on linux environment. However, one can easily do the same using other OS such as Windows or Macintosh.

In order to test the code, open the terminal on the linux machine. Now, go to the directory where the code has been saved and run it using the command: *python3 image_diff.py*
The new and old images corresponding to the coordinated will be obtained, masked as per requirement and then SSIM will be used to generate a thresh-hold change map. Further, the change contours will be added to the images and if there is a change, that is an SSIM score of less than 1, then the corresponding images with the change contours and threshold change map will be displayed on the screen. The program can be easily modified to save these result images instead of simply showing them by using imsave function instead of imshow.

Following are few samples from testing:

Palms Dubai:

Fig: Old and New Image of Palms Dubai

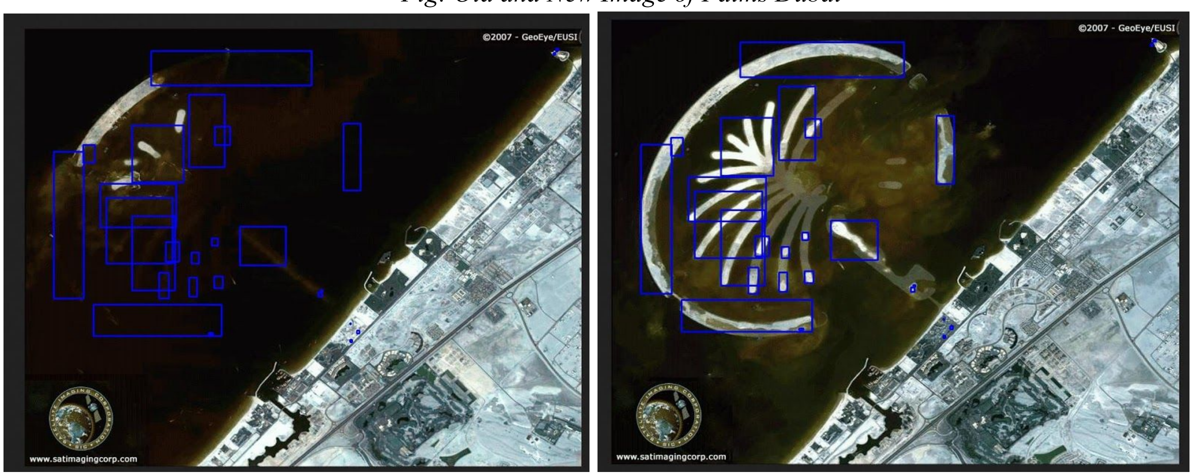


Fig: Marked Old and New Image of Palms Dubai after change detection

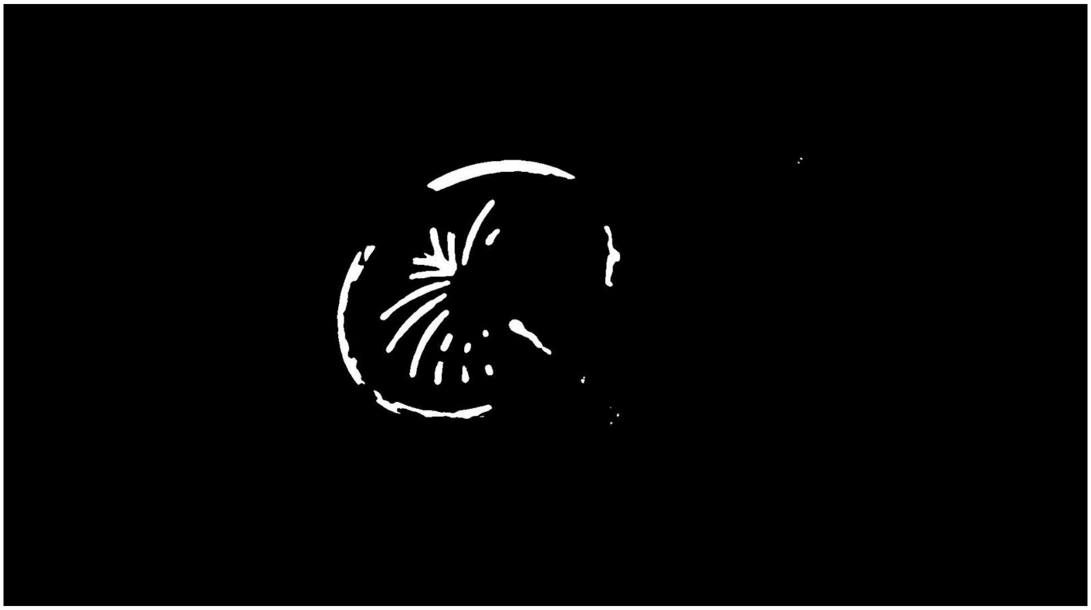


Fig: Change Map

Elephant Butte:

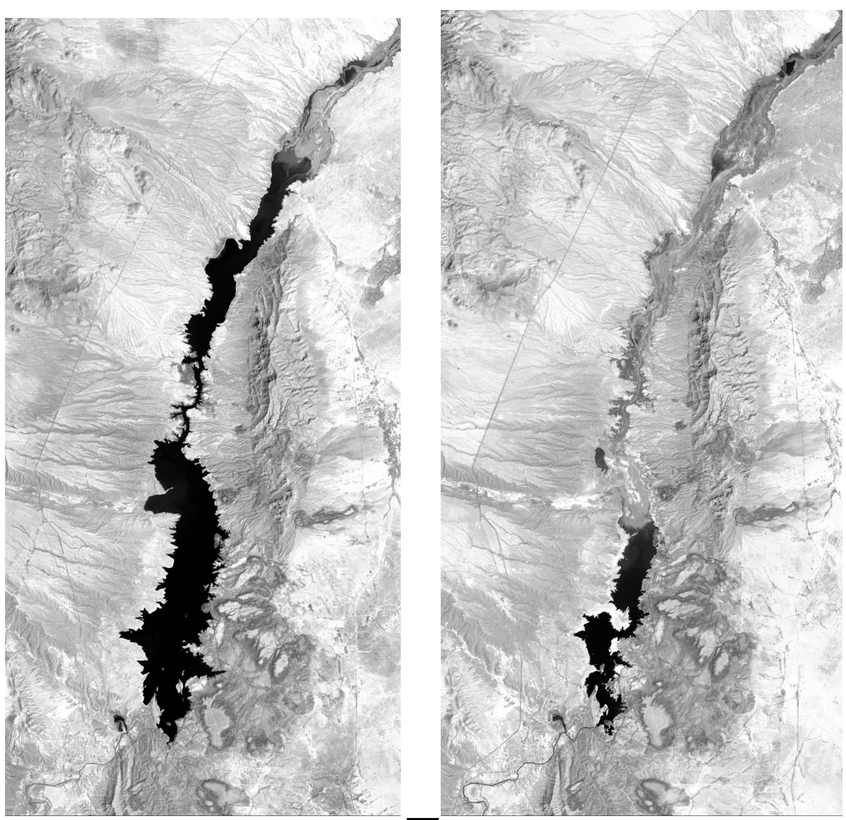


Fig: Old and New Image of Elephant Butte

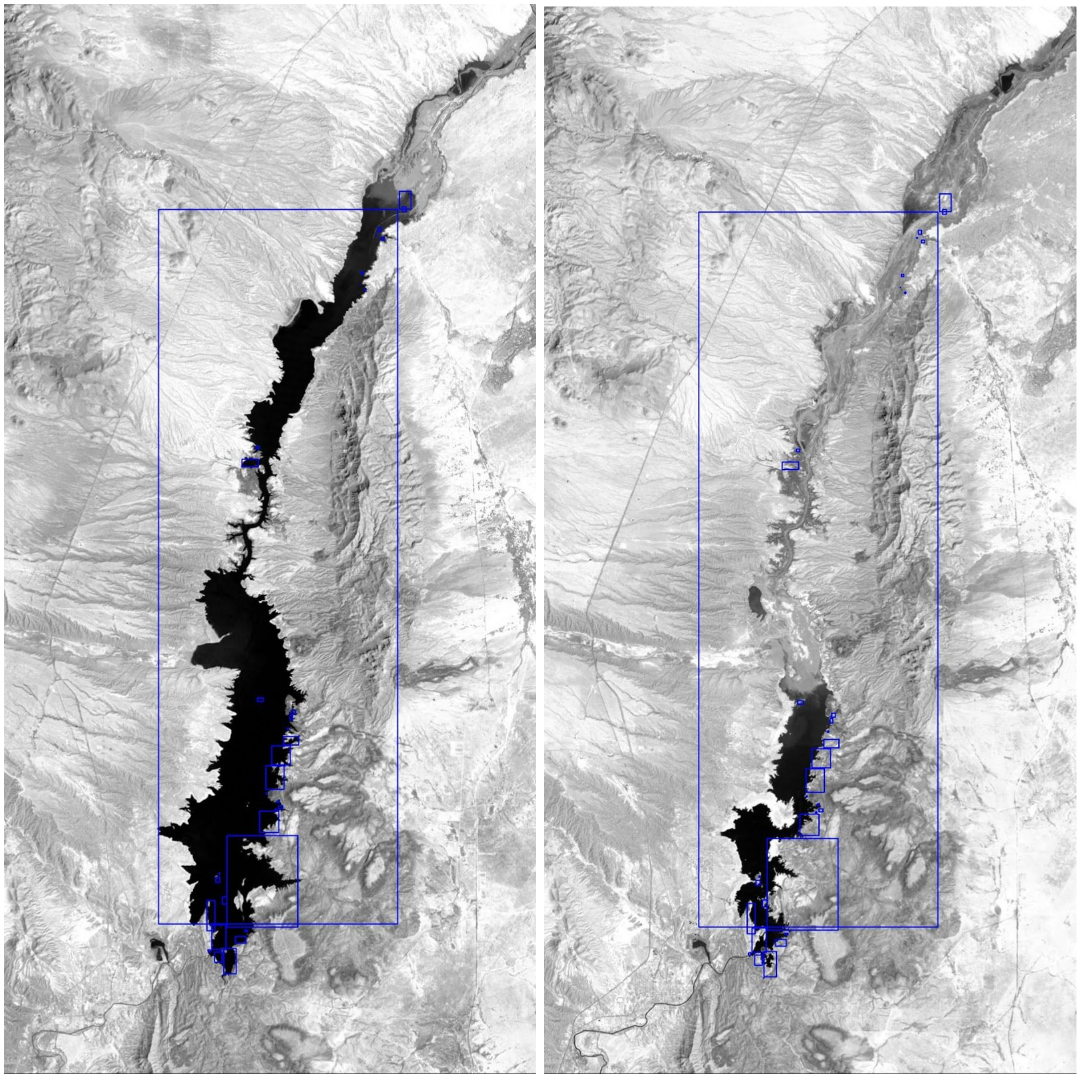


Fig: Marked Old and New Image of Elephant Butte after change detection

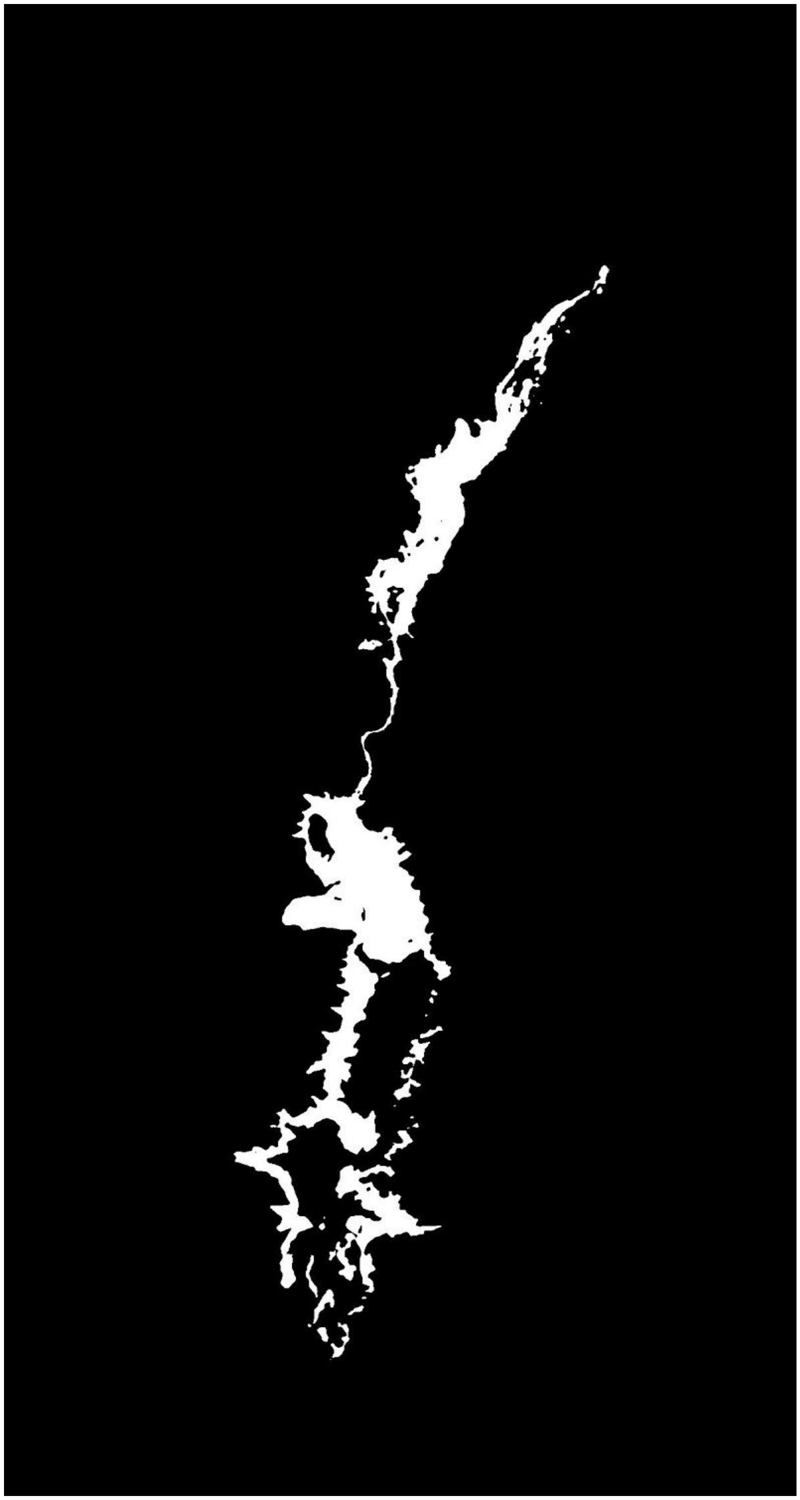


Fig: Change Map

<u>BISAG Plot 1:</u>

Fig: Old and New Image of Bisag plot 1

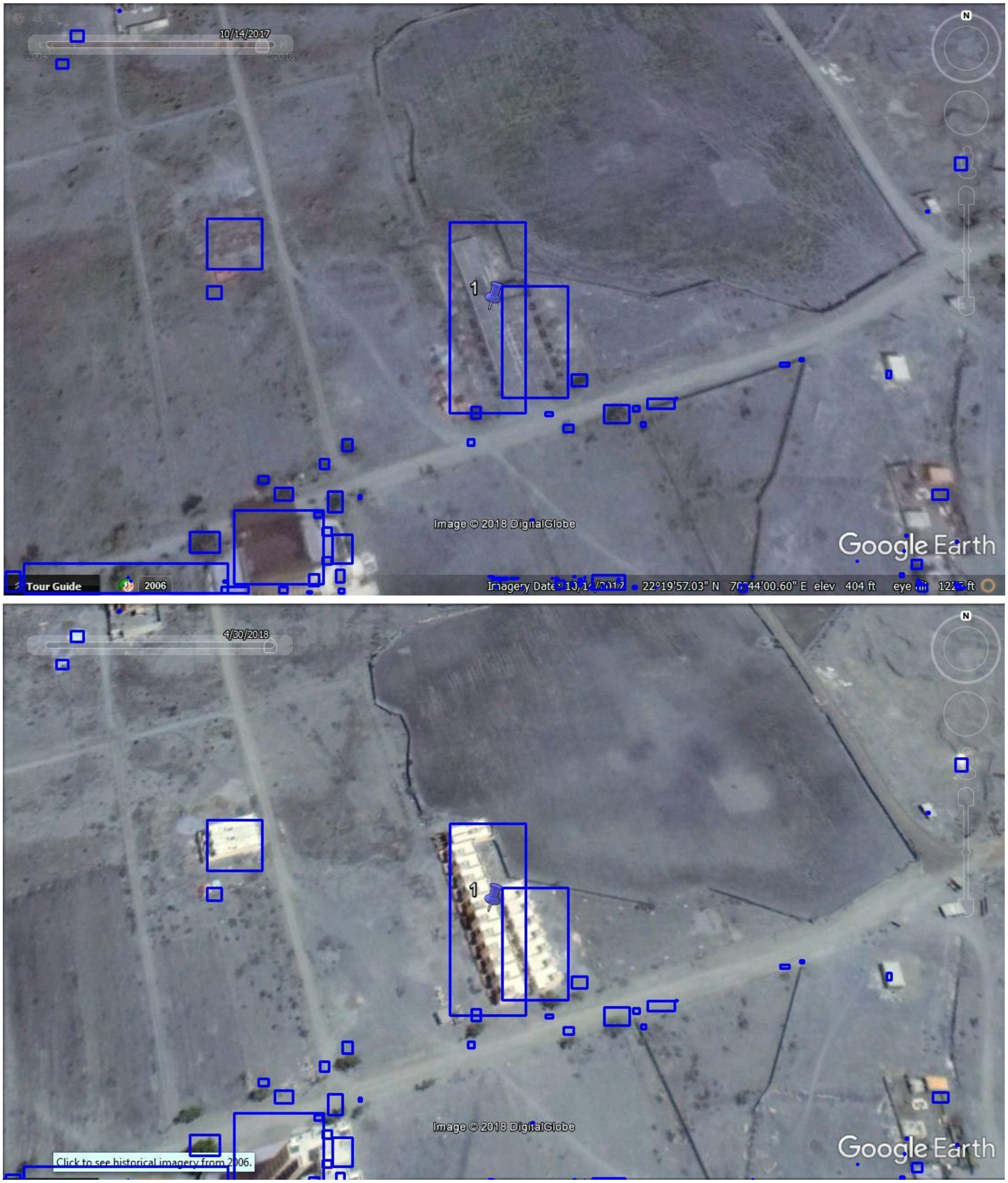


Fig: Marked Old and New Image of BISAG Plot1 after change detection

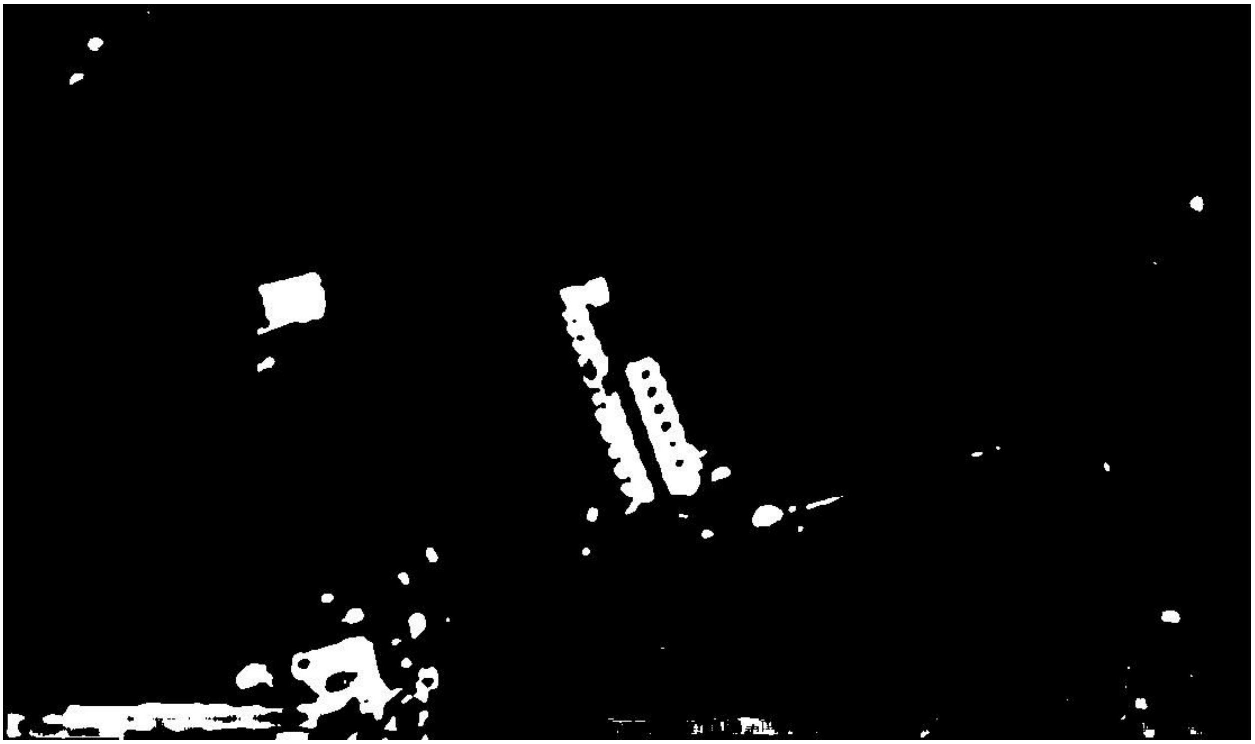


Fig: Change Map

BISAG Plot 2:

Fig: Old and New Image of Bisag plot 2

Fig: Marked Old and New Image of BISAG Plot 2 after change detection

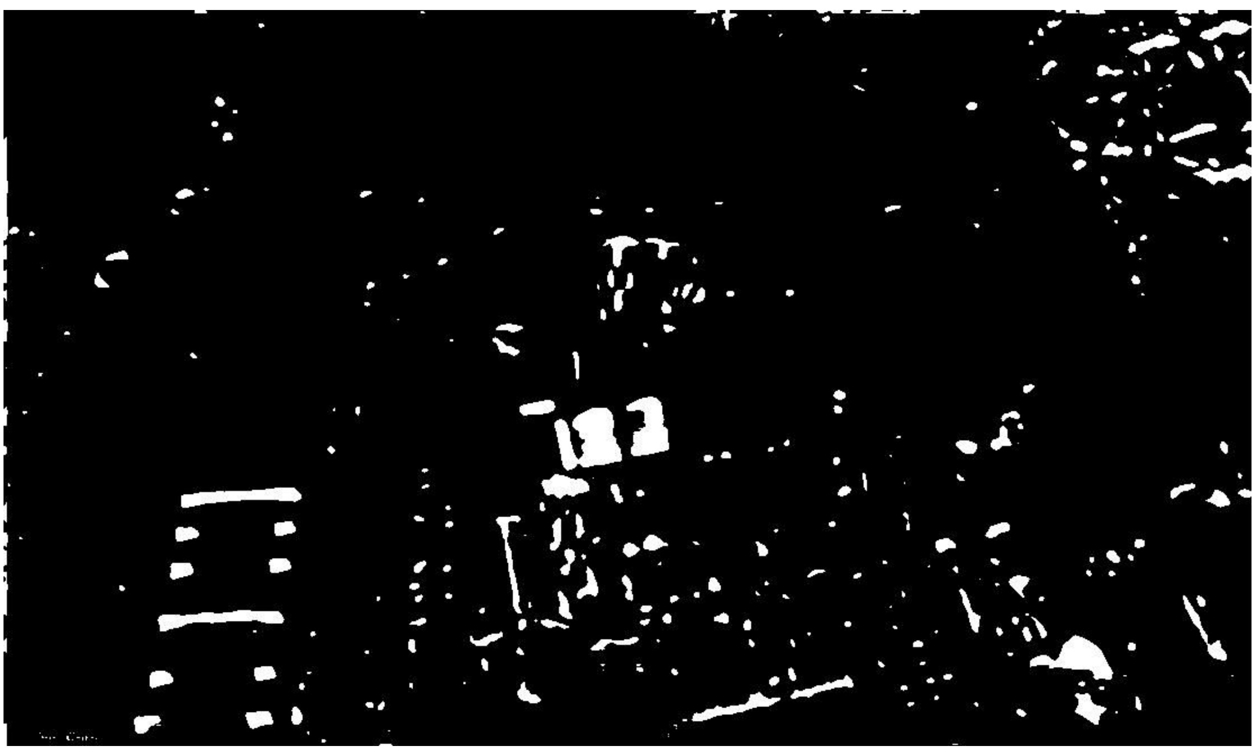


Fig: Change Map

2.6.4 Limitations and Conclusion

The data obtained through sentinelhub did not have a high resolution when the radius of the bounding-box is reduced too much. This is a limitation on the part of data availability and can be easily addressed by getting data from other sources such as Google Earth Engine, which is paid.

Only limitation of the application at present is that it simply detects changes, and not the type of changes. Thus it forms a basic technique of change detection that can be enhanced further

2.7 Web Application for end-user

Web application to allow a user friendly and easy experience for the end-user has been created using flask. The web app is hosted on the localhost presently.

Inputs: The input include

1. The ID of the plot whose changes are to be monitored.
2. The initial and final date of the period of monitoring

Outputs: The output include

1. SSIM Score of the images
2. Display the change contours marked images and change map.

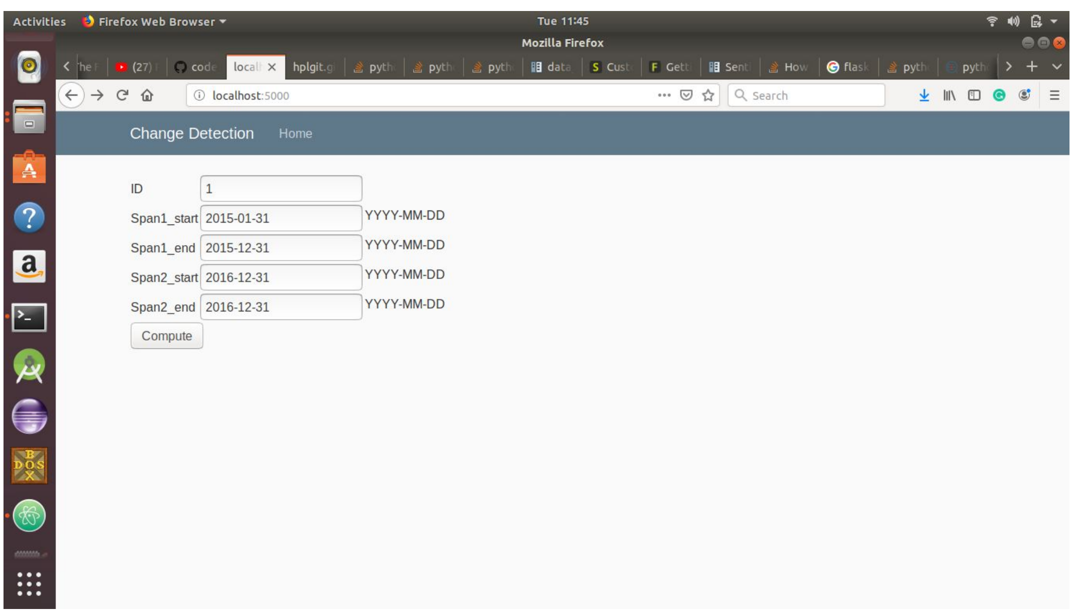


Fig: Screenshot of WebApp

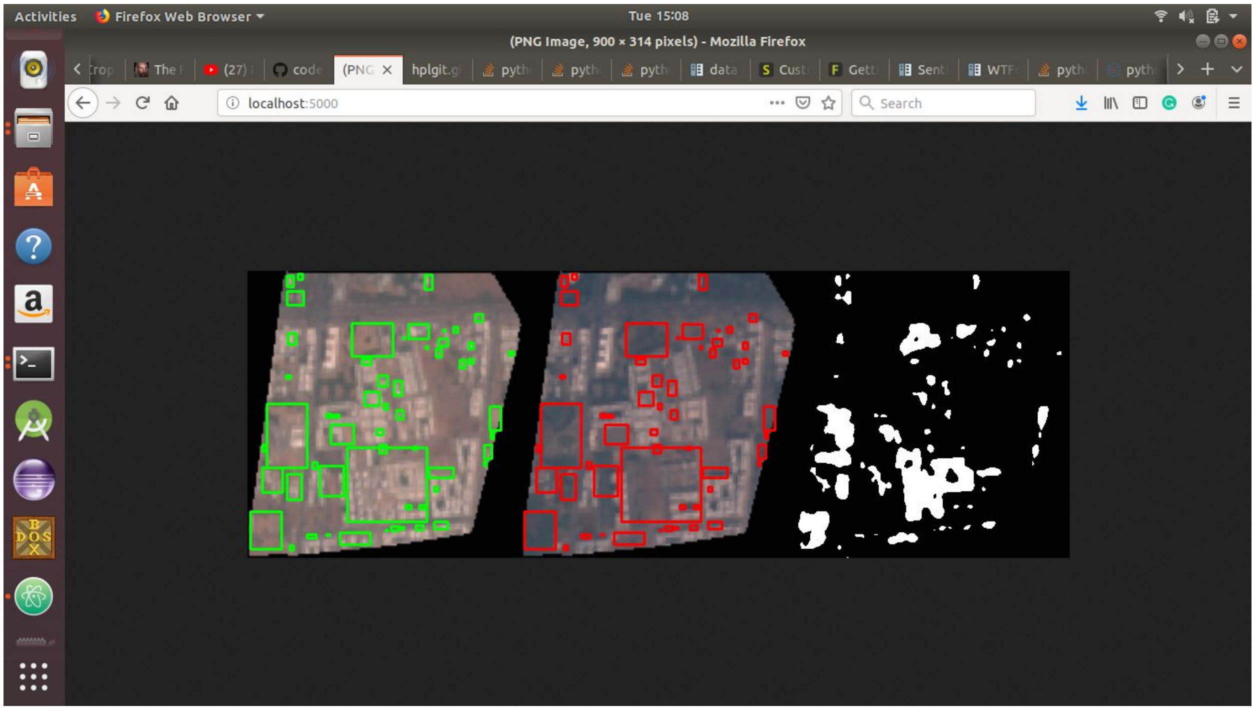


Fig: Screenshot of Output WebApp

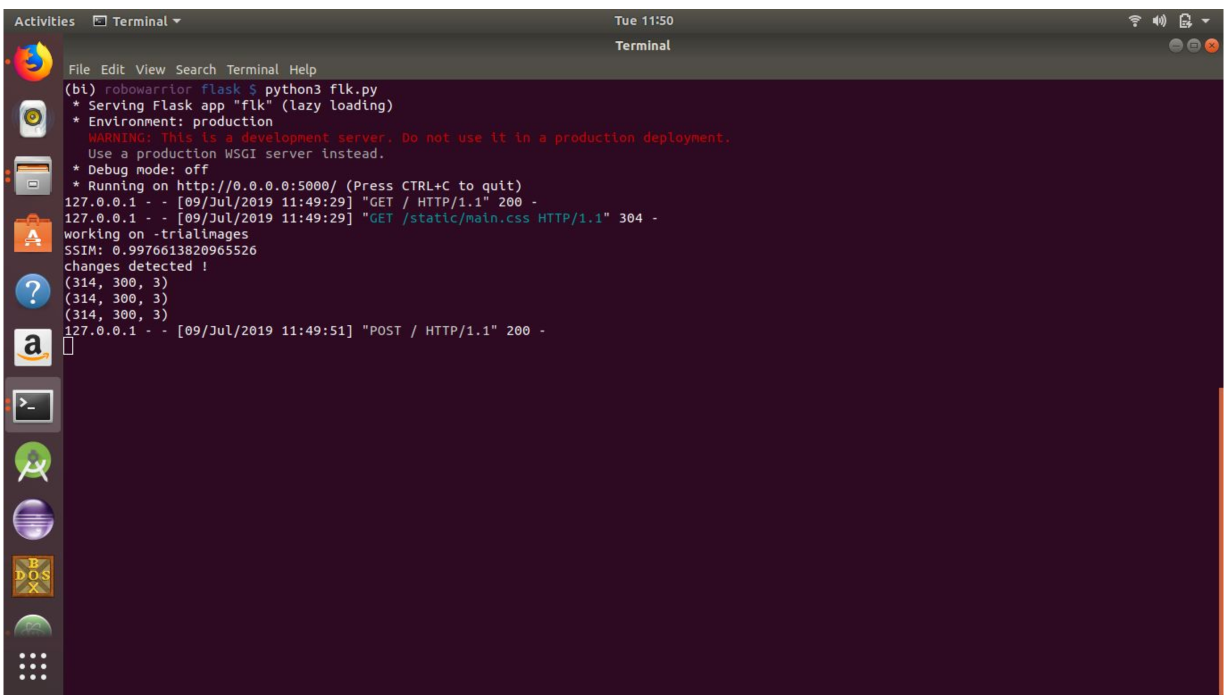


Fig: Screenshot of SSIM Score being displayed

2.8 Further Scope and Applications

By obtaining better quality data from other resources, the application can be further enhanced. Presently, the application is being used only to monitor change in usage of land of particular marked plots. It can be extended to monitor urban growth by comparing changes in the cities. The application can also find a major application in monitoring usage of natural resources. Changes in forest cover due to deforestation can be easily detected using the same technique. It can also be extended to monitor the water resources and coastlines. The same technique can be applied to images obtained by aerial photography in order to monitor the progress and extent of large scale dam or mining projects.

2.9 Conclusion

The project has been successfully implemented as a python program that performs change detection using the SSIM technique. This technique was found to give better performance as compared to PCA & K-Means. The data has been obtained from sentinelhub which is a cloud based GIS for viewing the data provided by sentinel-2. The data obtained is through a WMS Request which is the standard protocol for obtaining Geo-Spatial maps through http. Finally a web application using flask has been created to allow an easy and user friendly experience of working with the application. It allows the user to select the plot coordinates that has to be analysed as well as the radius of bounding-box and the dates in which the monitoring is to be performed.

BIBLIOGRAPHY

1. Adrian Rosebroke, Comparison of images using SSIM, September 15, 2014 retrieved from https://www.pyimagesearch.com/2014/09/15/python-compare-two-images
2. Documentation on SentinelHub python package retrieved from https://sentinelhub-py.readthedocs.io/en/latest/
3. Flask Documentation retrieved from http://flask.pocoo.org/docs/1.0/tutorial/
4. Iftekher Mamun, Image classification using SSIM, Jan 2017 retrieved from https://towardsdatascience.com/image-classification-using-ssim-34e549ec6e12
5. OpenCV Documentation and tutorials retrieved from https://docs.opencv.org/2.4/doc/tutorials/tutorials.html
6. Turgay Celik, “Unsupervised change detection in satellite images using Principal Component Analysis and K-means clustering”, IEEE Geoscience and Remote Sensing Letters, Vol. 6, No.4, October 2009.
7. Unsupervised Change Detection in Multi-Temporal Satellite Images using PCA & K-Means : Python code, retrieved from https://appliedmachinelearning.blog/2017/11/25/unsupervised-changed-detection-in-multi-temporal-satellite-images-using-pca-k-means-python-code
8. V. Andrearczyk & Paul F. Whelan, Using Filter Banks in Convolutional Neural Networks for TextureClassification, September 23, 2016