A Report
on

Meta-data Processing for Extracting Building Information from Satellite Images of Disasters

Submitted in fulfillment for the credits of course
of
Lab Based Project (CSN-300)
in
B.Tech. 3rd Year

Submitted by
Amit Vishwakarma (17114010)
Kishan Kumar (17114046)
Surya Prakash (17114062)

Under the guidance of
Dr. Sudip Roy
(Assistant Professor)
Department of Computer Science and Engineering

Indian Institute of Technology Roorkee
Roorkee- 247667, India
June, 2020

Abstract

When a disaster strikes, quick and accurate situational information is critical to an effective response. Before responders can act in the affected area, they need to know the location, cause, and severity of the damage. However, disasters can strike anywhere, disrupting local communication and transportation infrastructure, making the process of assessing specific local damage difficult, dangerous, and slow. This project is related to programming for automation in damage assessment during a disaster. Given a JSON file, mapping of polygons given in WKT format onto the corresponding PNG file of the remotely sensed RGB image. Implementation is to be done using Python programming language.

Contents

List of Figures

Chapter 1

Introduction

1.1 Objective

To map polygons given in JSON files onto corresponding remotely sensed RGB images. On mapping we need to obtain binary masks to identify buildings in the image then we need to color-code these masks showing the damage level of buildings

1.2 Motivation

With recent, abrupt changes in weather patterns around the world, natural disasters have become more unpredictable and have wider impacts than ever before. Logistics, resource planning, and damage estimation are difficult tasks after a disaster, and putting first responders into post-disaster situations is dangerous and costly. This project is an ode to passive method which can be used to perform damage assessment saving manpower and lowering risk. This project is about doing assessment on xBD, which is a dataset of pre-disaster and post-disaster images of disaster affected areas. The dataset contains image png files and JSON files which contain different polygon coordinates of buildings to be assessed in Well-Known Text format.

1.3 Problem Statement

This project is about masking of polygons on images which definitely can't be done manually by selecting each polygon then putting it on the image. So for automating this process we learnt about JSON parsing. Then after getting polygons we need to find a way to put these polygon objects on the image. For this we searched the web and found various techniques to do this but a combination of Shapely and OpenCV proves out to be most effective.

.

1.4 Dataset Details

The dataset consists of a total 5598 images with half of them being pre disaster version of the target zone and the other half of them being the same zone after the disaster with label files for each image consisting of wkt polygons annotating the buildings which are to be assessed for damage. The format of the image file is .png and that of the label file is JSON. The label file also contains other metadata data about the location like longitude and latitude, type of disaster which affected the location, etc.

Chapter 2

Literature Survey

1. **Earthquake damage detection using high-resolution satellite images(2013).** quickBird observed the city of Zemmouri, Algeria, before and after the May 21, 2003 Algeria earthquake. Using the pre-event and post-event pan-sharpened images, visual inspection of building damage was carried out by the five authors of this paper individually. A total 1,399 buildings were classified into five damage levels of European Micro-seismic Scale. The results from the different interpreters were reasonably close for collapsed buildings but the difference becomes larger for smaller damage levels. The locations of refugee tents in the two post-event images were also identified. These observations indicate that high-resolution satellite images can provide quite useful information to emergency management after natural disasters.

2. **Cyclone damage detection on building structures from pre- and post-satellite images using wavelet based pattern recognition(2015).** The majority of building structural losses during the past few decades has been due to wind induced damage, especially tropical cyclone damages. Rapid identification of damage locations as well as damaged buildings, and appropriate maintenance, can diminish the impact of such natural disasters. This paper describes detection of damage to cyclone-prone building roof structures from pre- and poststormsatellite imagesof the shoresof Punta Gordabefore and afterthe Hurricane 'Charley' 2004 disaster using a wavelet-based change detection method. Damaged buildings are automatically identified using wavelet-extracted statistical features and by edge detection, and classification using Artificial Neural Network (ANN) and Support Vector Machine (SVM). A comparison analysis is then carried out by comparing these results with results obtained using conventional change detection methods. It is observed that the wavelet-based change detection method yields identification information of damaged buildings superior to that obtained using conventional methods. In this work, the percentage of the damaged area of each damaged building is also calculated by a newly introduced texture-wavelet analysis on roof-tops, and the results are validated by counting the damage pixels manually. A positive increase in the extracted statistical features is observed as the percentage area of damage increases, which adds to the accuracy of the identification method

3. **Detection of Urban Damage Using Remote Sensing and Machine Learning Algorithms: Revisiting the 2010 Haiti Earthquake(2016).** Remote sensing continues to be an invaluable tool in earthquake damage assessments and emergency response. This study evaluates the effectiveness of multilayer feedforward neural networks, radial basis neural networks, and Random Forests in detecting earthquake damage caused by the 2010 Port-au-Prince, Haiti 7.0 moment magnitude (Mw) event. Additionally, textural and structural features including entropy, dissimilarity, Laplacian of Gaussian, and rectangular fit are

investigated as key variables for high spatial resolution imagery classification. Our findings show that each of the algorithms achieved nearly a 90

4. **Disaster Detection from Aerial Imagery with Convolutional Neural Network (2017).** In recent years, analysis of remote sensing imagery is imperatives in the domain of environmental and climate monitoring primarily for the application of detecting and managing a natural disaster. Satellite imagery or aerial imagery is beneficial because it can widely capture the condition of the surface ground and provides a massive amount of information in a piece of satellite imagery. Since obtaining satellite imagery or aerial imagery is getting more ease in recent years, landslide detection and flood detection is highly in demand. In this paper, They propose automatic natural disaster detection particularly for landslide and flood detection by implementing convolutional neural network (CNN) in extracting the feature of disaster more effectively. CNN is robust to shadow, able to obtain the characteristic of disaster adequately and most importantly able to overcome misdetection or misjudgment by operators, which will affect the effectiveness of disaster relief. The neural network consists of 2 phases: training phase and testing phase. They created training data patches of pre-disaster and post-disaster by clipping and resizing aerial imagery obtained from Google Earth Aerial Imagery. They were focusing on two countries which are Japan and Thailand. Training dataset for both landslide and flood consist of 50000 patches. All patches are trained in CNN to extract region where changes occurred or known as disaster region occurred without delay. They obtained accuracy of our system in around 80

5. **Detecting Damaged Building Regions Based on Semantic Scene Change from Multi-Temporal High-Resolution Remote Sensing Images(2017).** The detection of damaged building regions is crucial to emergency response actions and rescue work after a disaster. Change detection methods using multi-temporal remote sensing images are widely used for this purpose. Differing from traditional methods based on change detection for damaged building regions, semantic scene change can provide a new point of view since it can indicate the land-use variation at the semantic level. In this paper, a novel method is proposed for detecting damaged building regions based on semantic scene change in a visual Bag-of-Words model. Pre-and post-disaster scene change in building regions are represented by a uniformvisual codebook frequency. The scene change of damaged and non-damaged building regions is discriminated using the Support Vector Machine (SVM) classifier. An evaluation of experimental results, for a selected study site of the Longtou hill town of Yunnan, China, which was heavily damaged in the Ludian earthquake of 14 March 2013, shows that this method is feasible and effective for detecting damaged building regions.

Chapter 3

Methodology

3.1 Phase I

3.1.1 Approach

So our first step of pre-processing involved collecting the pre-disaster image, label pairs and post-disaster image, label pairs separately, For which we have used simple regular expression to divide them based on their file name and collected them using dictionary data structure such that given a name of image as a key to the dictionary we can access its corresponding JSON label file. Next, having collected the image-label pairs our next step of preprocessing involved analysing and implementing a function to parse the JSON file to collect the x,y coordinates of polygons which we achieved the same through the inbuilt JSON parsing library in python.

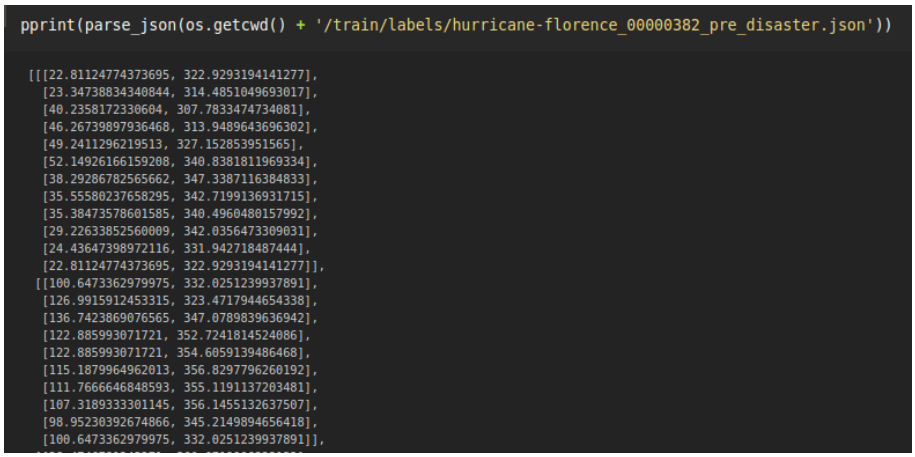


Figure 3.1: Polygon coordinates with input as label file

So, given an image having collected the wkt polygon coordinates our next work involved plotting the polygons and creating a binary mask for buildings in the images. We attempted to do this for a single sample image first in the following manner. We first collected the wkt polygon coordinates in a list and converted them to shapely objects for convenience in plotting (Shapely is a Python package for set-theoretic analysis and manipulation of planar features), then we converted the list of polygons to a MultiPolygon object which can be passed as a parameter to OpenCV's fillpoly drawing function using which we were successfully able to obtain a binary mask of the polygons identifying buildings in the given image.

3.2 Phase II

Previously, only pre-disaster images were masked using two colors, black for background and white for building. Now post-disaster are also masked using multi-color with color coding representing damage level of buildings after disaster. Following colors represent damage levels:-

1. Red: Destroyed
2. Blue: Major-damage
3. Yellow: Minor-damage
4. Pink: No-damage
5. Green: Un-classified

3.2.1 Why is color masking required?

After capturing images from satellites before and after disaster, they need to be compared to get the exact extent of damage in the area after disaster. This is not possible by just looking at the images. Talking about the extent of damage of empty ground is absurd, thus we need to know about the damage of objects, buildings in the area to get an overall idea of damage. However this evaluation has been already done and information about extent of damage has been stored in a JSON file. If we could color the objects, buildings according to its damage-level then we will know better which part of the area needs first response. We in this project are working only with those JSON files.

3.2.2 Difference between pre-JSON and post-JSON files

Since the location of buildings in both pre- and post- images are the same, thus the polygons representing buildings in both images are identical. The only difference we can see is that there is an additional parameter **"subtype"** which represents the damage-level of buildings(polygons) after the disaster has happened. After exploring lots of JSON files we concluded that **"subtype"** has 5 different types of values namely:

1. Destroyed
2. Major-damage
3. Minor-damage
4. No-damage
5. Un-classified

These values are what we need which decides with which color a polygon needs to be filled. To use these values we extracted them in the same order of polygons. Now what we are left with is masking these polygons on images whose color is governed by subtype values.

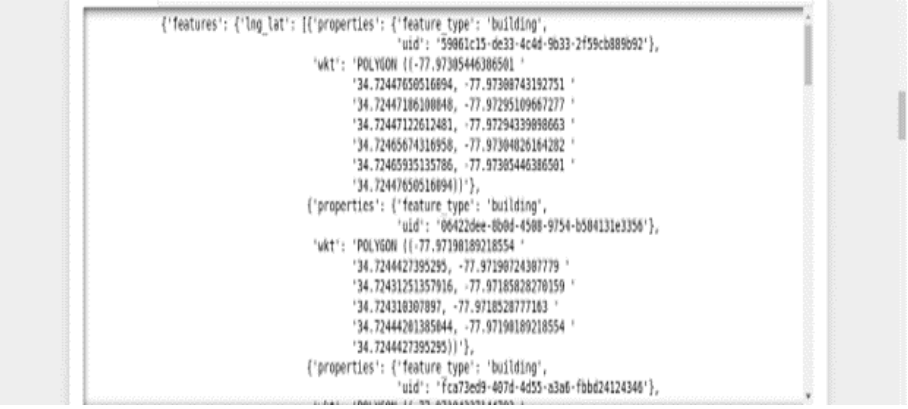


Figure 3.2: Hurricane-florence-pre-disaster.json

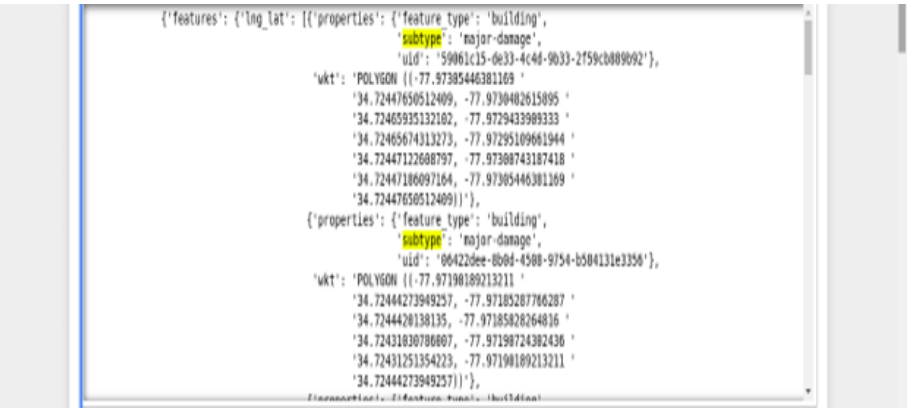


Figure 3.3: Hurricane-florence-post-disaster.json

3.2.3 Approach

Data-preprocessing has been already done and some important functions have been already defined. We just manipulated and rewritten some of them to account for both binary and colored masking. To achieve colored masking, we first need to import images. We used the 'imread' module from scipy.misc which is helpful in obtaining image data without altering it. Then extracted subtype values of polygons by parsing JSON files in a list. We also obtained polygons in the same order of subtype values in other lists.

Now this polygon list needs to be converted to a multipolygon object which is required in fillPoly() function of openCV. Multipolygon objects were obtained using the Shapely library. For binary-masking we directly passed this multipolygon object in fillPoly() because there only one color was needed to be filled in polygons. But now we require different colors for different polygons, so we need to iterate over this multipolygon object to work with it.

Also we can see that the coordinates of polygons are rounded upto 14 decimals in the JSON files thus we need to convert these into integers, because fillPoly() doesn't recognise double values. This also results in loss of polygons data. We can see that in some masked images buildings are totally covered but partially covered in a very few of them. This is because of loss of polygons data while rounding off. We then iterated over this multipolygon object and passed this object+image data(which is stored in an array)+color values as parameters to cv2.fillPoly() function for masking. This color value is different for different polygons which is determined by a dictionary 'labels', it has keys as 'subtype' values which directs color.

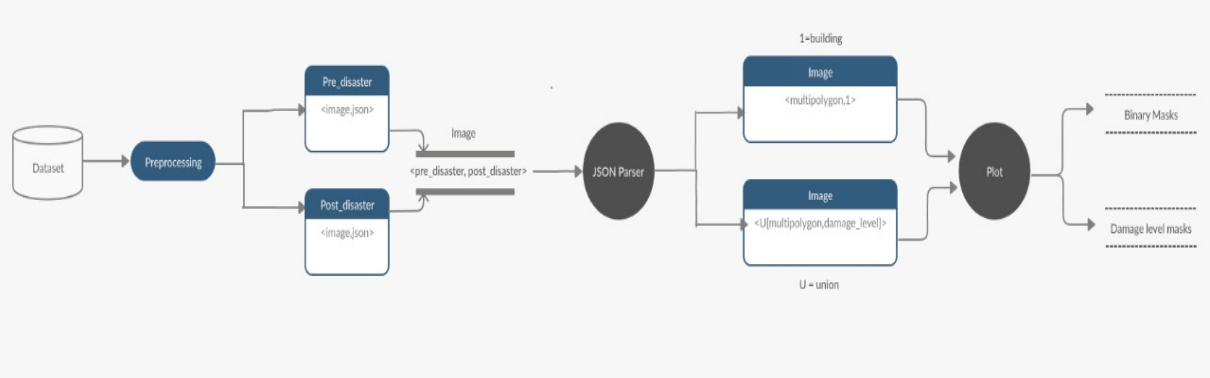


Figure 3.4: Workflow

This is the plot summarising our workflow during the project.

1. Binary masks is for **building** vs **non-building** classification.
2. Damage level masks is for **damage level** classification among buildings.

Chapter 4

Results

In some of the binary-masked images we can see that the image is totally black OR in color-masked there is no masking of any color at all, this shows that corresponding label file doesn't contain any wkt-polygon. This can be verified by parsing the corresponding JSON file. But only those buildings in images were masked which had its polygon coordinates in a JSON file.

If a building is there in image which appears to be not masked then this will be due to absence of polygon coordinates corresponding to it in JSON file. This glitch may have occurred during generating polygon coordinates of buildings from image which is beyond the scope of this project.

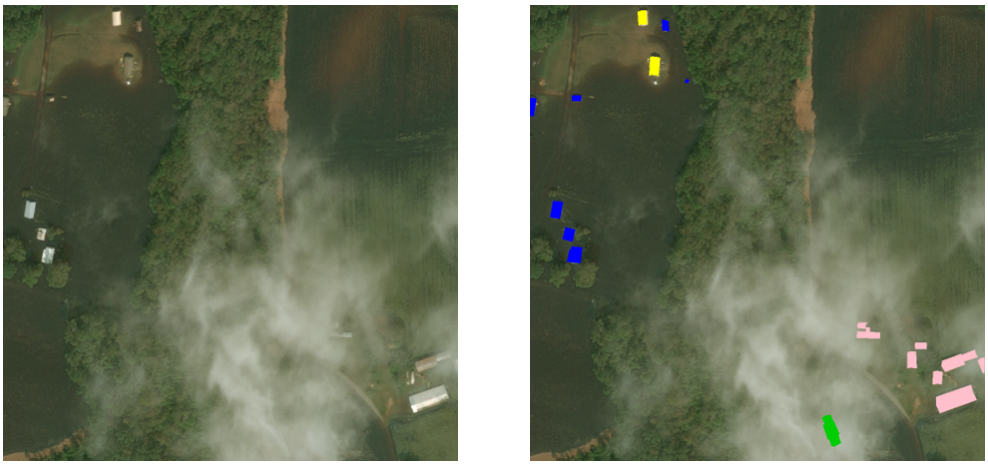
Except for figure 3.1, in all the images below these parameters represent damage levels:

1. RED - Destroyed
2. BLUE - Major-damage
3. YELLOW - Minor-damage
4. PINK - No-damage
5. GREEN - Un-classified

.

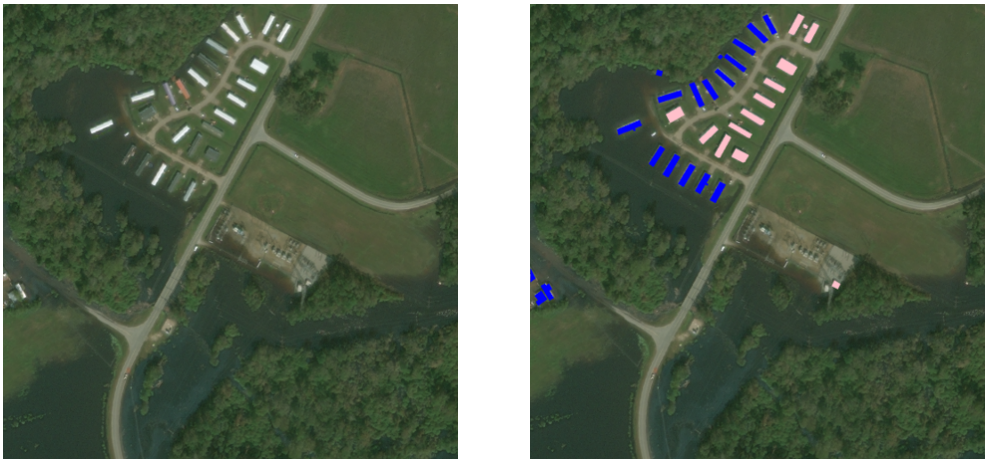
(a) Image (b) Mask

Figure 4.1: Santa-rosa-wildfire-1



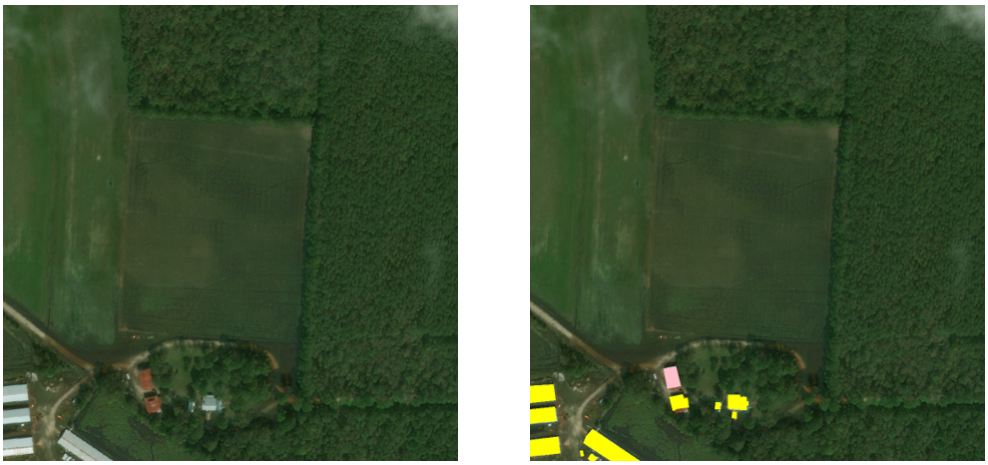
(a) Image (b) Mask

Figure 4.2: Hurricane-florence-1



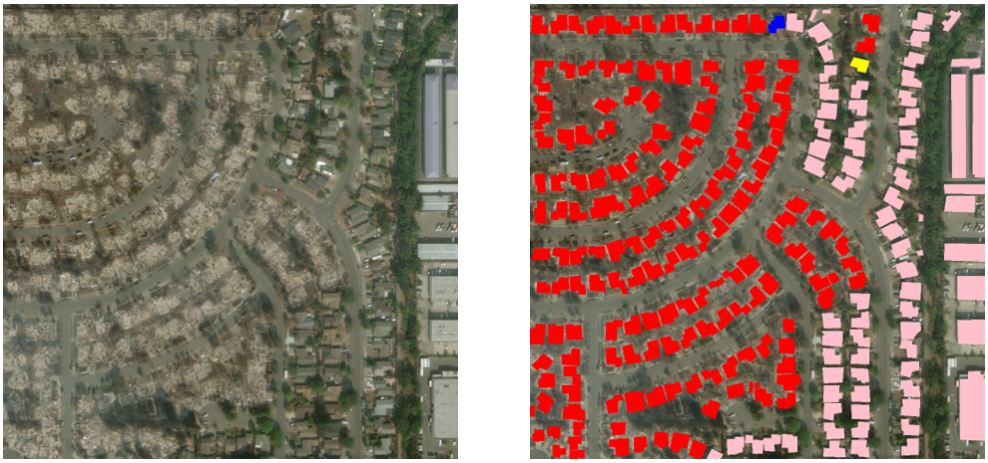
(a) Image (b) Mask

Figure 4.3: Hurricane-florence-2



(a) Image (b) Mask

Figure 4.4: Hurricane-florence-3



(a) Image (b) Mask

Figure 4.5: Santa-rosa-wildfire-2

.

Chapter 5

Conclusion and Future Work

With the help of python JSON techniques and important open source image processing libraries like OpenCV and Shapely we are able to successfully mask the images. But some precautions need to be taken care before working with training data.

1. The training data must be properly defined and filtered. There should not be any missing image corresponding to a label file or vice-versa.

2. There should not be any duplicate image or label file in images or labels folder otherwise these will result in inaccurate results. Because, creation of image-label pairs is based on sorting, not on hash value of images and labels which takes far greater time to process the data than the sorting method.

By masking the images we can know a lot about the disaster affected areas and help in informing the front-line responders to which area they need to provide the relief first.

Bibliography

[1] S. Gillies, "How to use the shapely python package for computational geometry." https://shapely.readthedocs.io/en/latest/manual.html. Accessed: March,2020.

[2] OpenCV, "Drawing functions, fills the area bounded by one or more polygons." https://docs.opencv.org/2.4/modules/core/doc/drawing_functions.html#fillpoly. Accessed: March,2020.

[3] SciPy, "Read an image from a file as an array.." https://docs.scipy.org/doc/scipy-1.2.1/reference/generated/scipy.misc.imread.html. Accessed: April,2020.

[4] R. G. et al, "A dataset for assessing building damage from satellite imagery." arXiv preprint arXiv.

[5] F. Yamazaki, K. Kouchi, M. Kohiyama, N. Muraoka, and M. Matsuoka, "Earthquake damage detection using high-resolution satellite images," in *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, vol. 4, pp. 2280–2283 vol.4, 2004.

[6] S. Radhika, "Cyclone damage detection on building structures from pre- and post-satellite images using wavelet based pattern recognition." sciencedirect.com/science/article/pii/S0167610514002207. Accessed: March,2020.

[7] D. F. W. H. Q. S. H. Tu, J.; Li, "Detecting damaged building regions based on semantic scene change from multi-temporal high-resolution remote sensing images.." https://www.mdpi.com/2220-9964/6/5/131#cite. Accessed: March,2020.

[8] Y. Cooner, A.J.; Shao, "Detection of urban damage using remote sensing and machine learning algorithms: Revisiting the 2010 haiti earthquake.." https://www.mdpi.com/2072-4292/8/10/868#cite. Accessed: March,2020.

[9] S. N. K. B. Amit and Y. Aoki, "Disaster detection from aerial imagery with convolutional neural network," *2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, pp. 239–245, 2017.