Final Technical Report

USGS Award # G19AP00056 and G19AP00057

Building Collapse Detection using Satellite Imagery after Earthquake Events: Collaborative Research with Tufts University and Boston University

Laurie G. Baise, Professor and Chair

Dept. Civil and Env. Engineering, Tufts University, 200 College Ave, Medford, MA, 02155

617-627-2211, 617-627-3994, laurie.baise@tufts.edu

Babak Moaveni, Professor

Dept. Civil and Env. Engineering, Tufts University, 200 College Ave, Medford, MA, 02155

617-627-5642, 617-627-3994, babak.moaveni@tufts.edu

Magaly Koch, Research Associate Professor

Center from Remote Sensing, Boston University, 725 Commonwealth Ave, Boston, MA, 02215

617-353-7302, mkoch@bu.edu

June 1, 2019 – May 31, 2020

This material is based upon work supported by the U.S. Geological Survey under Grant No. G19AP00056 and G19AP00057.

Abstract

Introduction

With the increase in availability of high resolution satellite data after natural disasters as a result of an increased number of available satellites that have been launched in recent years, we have an opportunity to use this stream of high-resolution optical data with increasing temporal resolution to map damage immediately after natural disasters. The raw images provided by the satellites do not contain labeled pixels or objects and therefor it is crucial for us to develop training classifiers using Machine Learning algorithms to fill this gap [9]. To overcome this issue, the "xView" dataset has been recently provided by the Defense Innovation Unit Experimental (DIUx) and NGA for a broad range of research in computer vision and object detection [10]. xView is one of the largest publicly available dataset of labeled satellite imagery. It has thousands of satellite images containing different labeled objects on the ground including intact and demolished buildings. All images in the xView are captured by World-View 3 sensor and have a spatial resolution of 30 cm and are provided in RGB (Red, Green, and Blue spectral bands) mode by MAXAR company. The World-View 3 (as well as World-View 2) collects imagery across eight spectral bands but xView only includes RGB. To compare the value of the additional spectral bands, we compare spectral content from collapsed and intact buildings using xView data (RGB) and World-View 2 imagery from after the September 19, 2017 Mexico earthquake which resulted in 20 collapsed buildings. Using the larger xView labeled dataset and deep learning methods, we test an automated framework to detect demolished buildings from RGB optical imagery after a natural disaster.

In this project, we used the xView dataset as a way of evaluating the identifiability of collapsed buildings with RGB optical imagery. We compare spectral content of collapsed buildings and intact buildings across the RGB bands and compare that to the spectral content of both collapsed

and intact buildings using the full spectral range for a smaller dataset derived from World-View 2 imagery from the September 19, 2017 Mexico earthquake. To evaluate automated classification methods applied to the xView dataset and RGB imagery, we train a model based on a fully Convolutional Neural Network (CNN) to detect demolished and intact buildings from the satellite imagery and generate prediction masks. We test the accuracy of the framework by generating confusion matrix and calculating conventional indices such as sensitivity and specificity.

xView satellite imagery (RGB)

One of the motivations for this research was the free availability of thousands of satellite imagery provided recently by the Defense Innovation Unit Experimental (DIUx) and National Geospatial Intelligence Agency (NGA), also known as xView [10]. xView is a collection of thousands of satellite imagery that has been labeled to provide motivation for creating sophisticated, novel and robust models and algorithms that can detect different objects on the ground level. xView dataset is huge in quantity (covers more than 1400 square kilometers) and has a very high spatial quality (30 cm resolution). All the images have been corrected for atmospheric effect, orthorectified and pansharpened. xView contains more than 2 million labeled instances across 60 object categories. This diversity in object classes is to make the xView suitable for applied researches in different majors including disaster response. “Demolished building” and “building” are two of the 60 classes with more than a thousand and 300,000 of instances, respectively, across the world to account for geographic diversity. Each geographic location has its own specific natural and artificial features such as physical differences (forest, coastal, desert) and constructional differences (layout of houses, cities, roads); this geographical diversity will increase the perspective of objects within a class [10]. All the images within xView are from MAXAR (formerly Digital Globe) and captured by WorldView-3 sensor; this is an important advantage of

xView dataset that all the images are from a same sensor as this will eliminate the spectral bias across the images and minimize the variation of spectral information between the same objects on the ground in different images. xView provides the satellite imagery only in three visible bands (Red, Green, Blue). The xView labeling process has been carried out by humans and with extensive care and multiple quality standards [10]. Labeling quality has been controlled in three stages: 1) labeler; 2) supervisor; and 3) an expert. For more details about the labeling process, please look at the xView dataset manual provided by [10]. Figure 1 shows an example of how the xView provides demolished building labels. Note that in this Figure, demolished building labels are boxed in red.

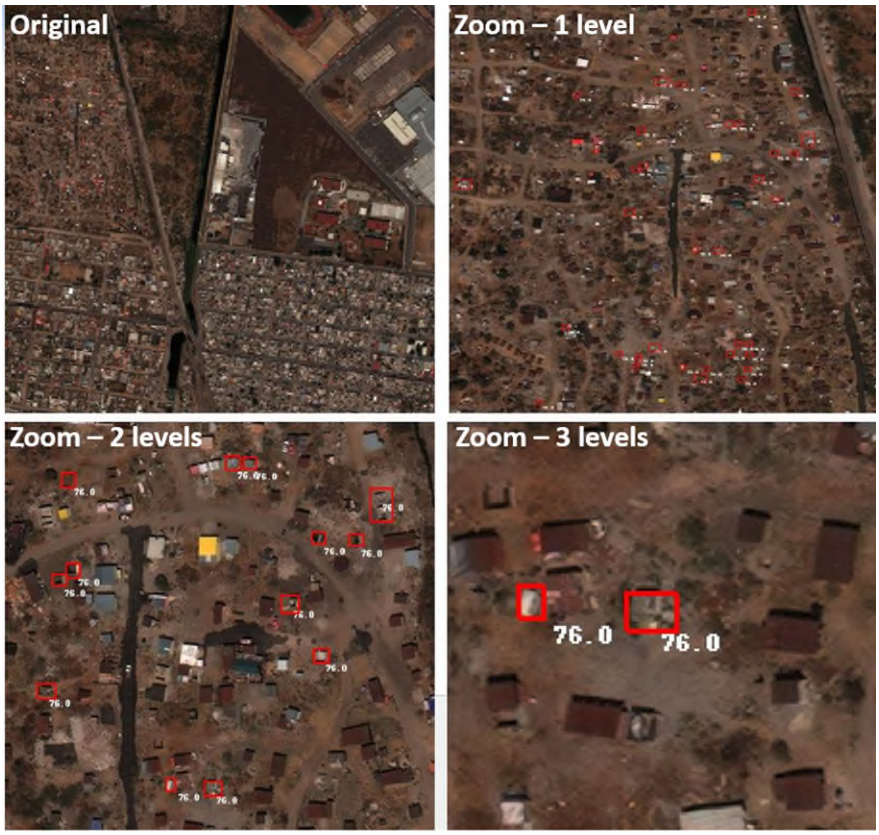


Figure 1. Satellite imagery from the xView dataset with demolished buildings shown in red boxes. The four tiles show four different zoom levels. The 76.0 label identifies the collapsed building label.

September 19, 2017 Mexico Earthquake Imagery (8 multi-spectral bands)

The September 19, 2017 Mexico Earthquake resulted in the collapse of 20 buildings. To evaluate the benefit of the additional spectral content in identifying collapsed buildings, we acquired pre- and post-event imagery from Digital Globe (now MAXAR) that included 19 of the collapsed buildings. The pre-event imagery was captured on January 16, 2016 and the post-event imagery was captured on October 20, 2017 (roughly one month after the event). These dates provided the best cloud-free imagery. The imagery includes 8 multi-spectral bands at 2 m resolution and the panchromatic band at 50 cm resolution. Figure 2 shows the imagery and an example of a collapsed building in pre and post-event imagery surrounded by intact buildings. The imagery was corrected for atmospheric effects, orthorectified and pansharpened.

Figure 2. a. World-View 2 imagery from October 20, 2017 capturing collapsed and intact buildings after the September 19, 2017 Mexico Earthquake. b. zoom of collapsed and intact buildings shown pre-event (left) and post-event (right).

Spectral Analysis

Using the multi-spectral imagery from the 2017 Mexico earthquake, we were able to assess the contribution of each spectral band on identifying collapsed buildings. Figure 3 shows different bands of the WorldView-2 sensor and their corresponding wavelength on the spectrum. Generally, objects on the ground have different reflection when exposed to a same wavelength of energy; therefore, having more spectral bands could help up better discriminate objects on the ground. In Figure 4, we show the spectral reflectance in

pre-event and post-event imagery for 19 collapsed buildings. Spectral band 2, 3, and 5 correlate with the Blue, Green, Red bands, respectively. The mean spectral reflectance increases for most of the spectral bands including RGB in all 19 buildings in the post-event imagery as compared to the pre-event imagery. As demonstrated by Building 17, the mean spectral reflectance decreases at Spectral Band 7 and 8 for some cases when comparing post-event to pre-event imagery. The variance of the spectral reflectance also increases across all spectral bands when comparing post-event and pre-event imagery, with the exception of Building 7, 8, 14, and 19. The spectral trends from pre- and post-event imagery appear consistent between the RGB bands and the remaining spectral bands (1, 4, 6-8). If we compare the spectral reflectance across all 19 buildings within each of the spectral bands as shown in Figure 5, we can evaluate consistency of differences between pre and post imagery.

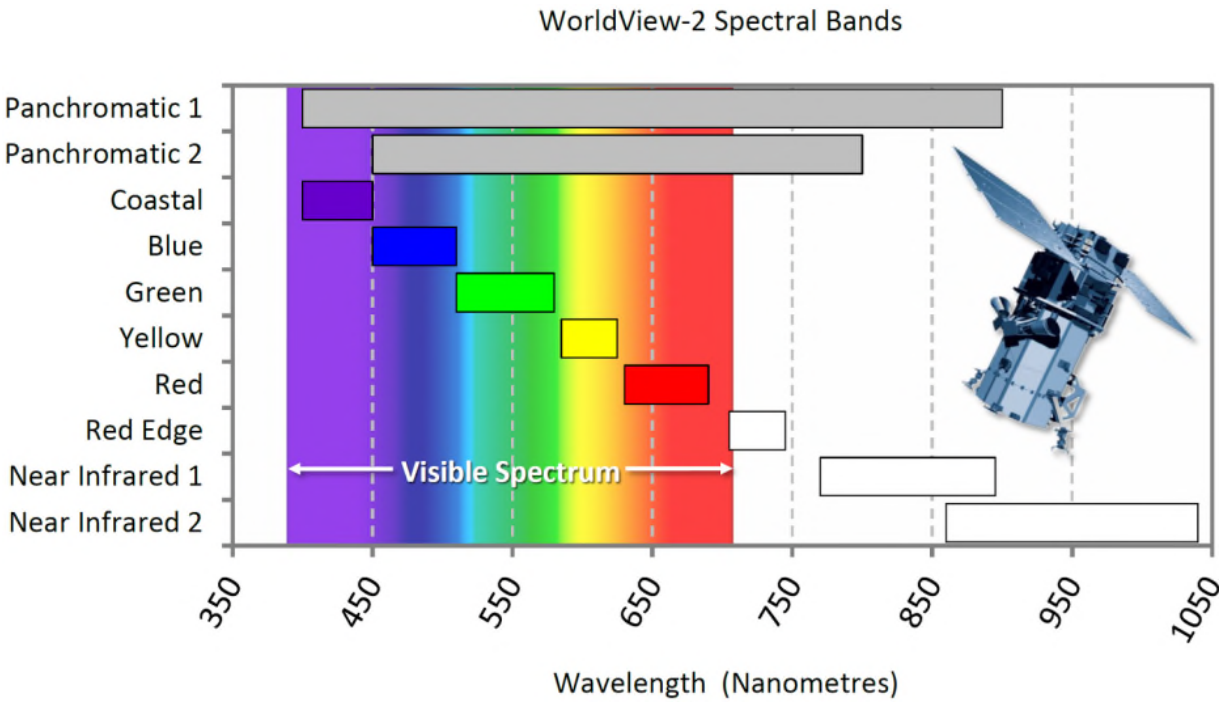
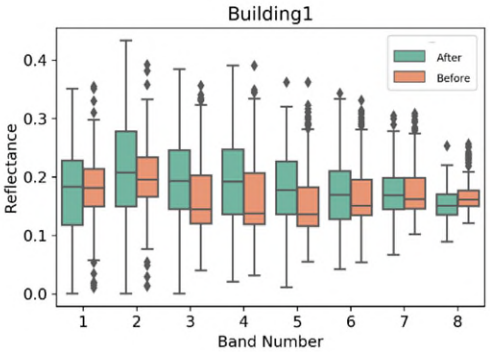
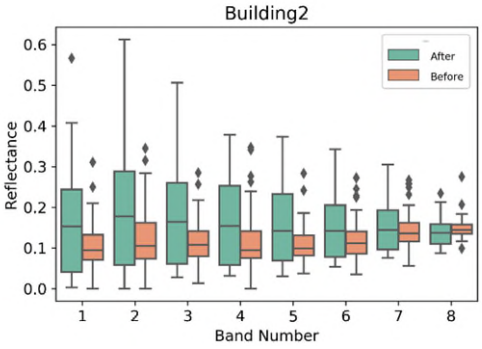
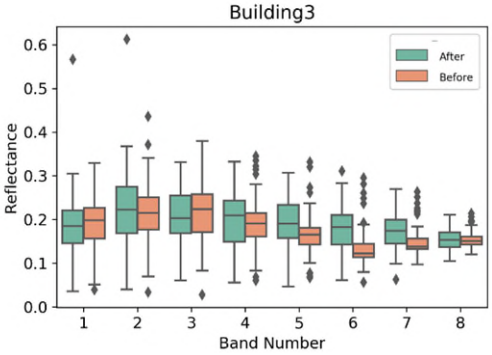
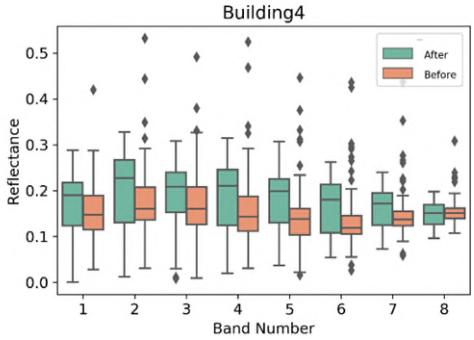


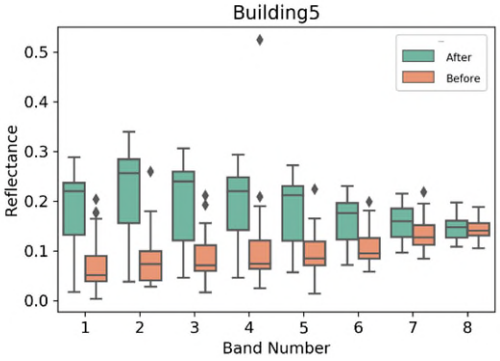
Figure 3. Spectral bands in Worldview-2 sensor and their corresponding wavelength value. Spectral Band 1=Coastal, 2=Blue, 3=Green, 4=Yellow, 5=Red, 6=Red Edge, 7=Near Infrared 1, and 8=Near Infrared 2

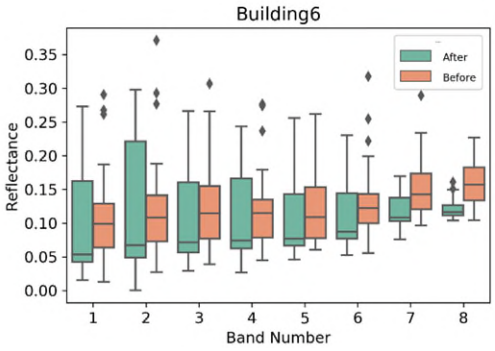

Building1
After
Before
Reflectance
Band Number

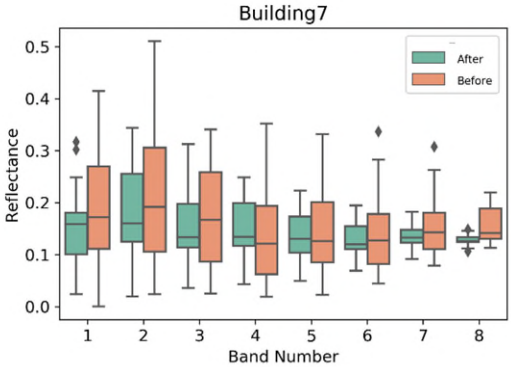

Building2
After
Before
Reflectance
Band Number

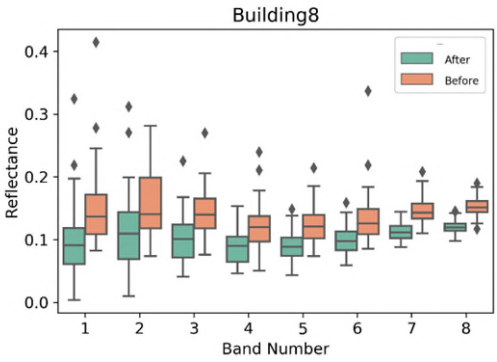

Building3
After
Before
Reflectance
Band Number

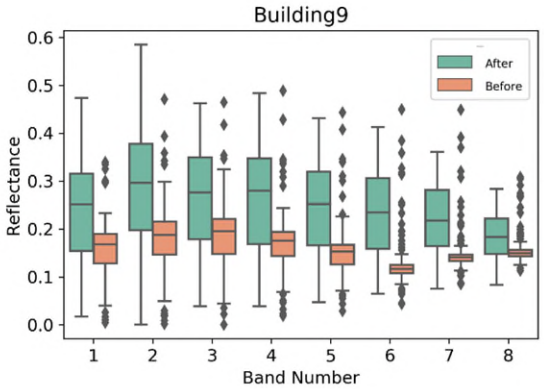

Building4
After
Before
Reflectance
Band Number

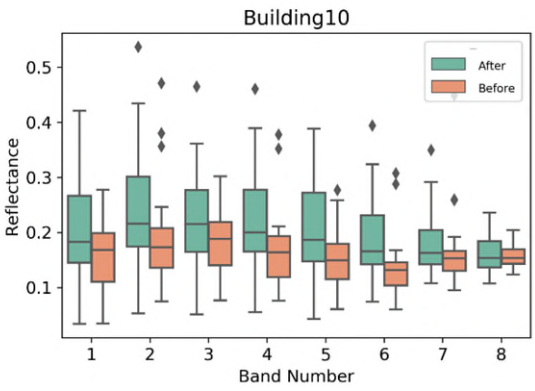

Building5
After
Before
Reflectance
Band Number

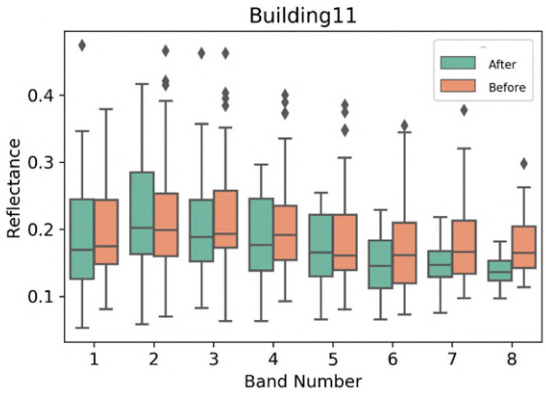

Building6
After
Before
Reflectance
Band Number

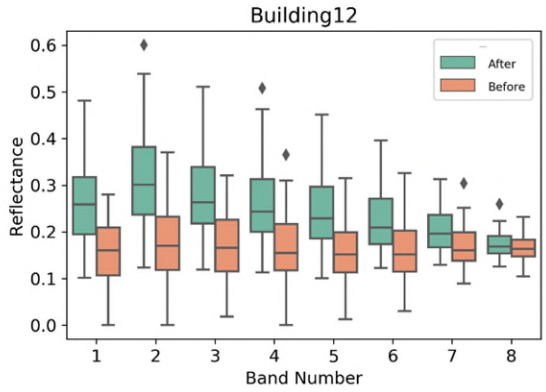

Building7
After
Before
Reflectance
Band Number

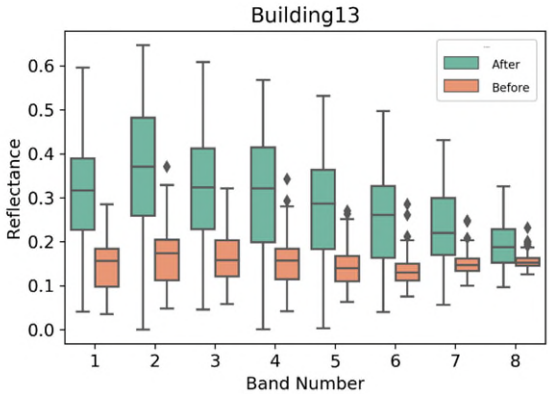

Building8
After
Before
Reflectance
Band Number

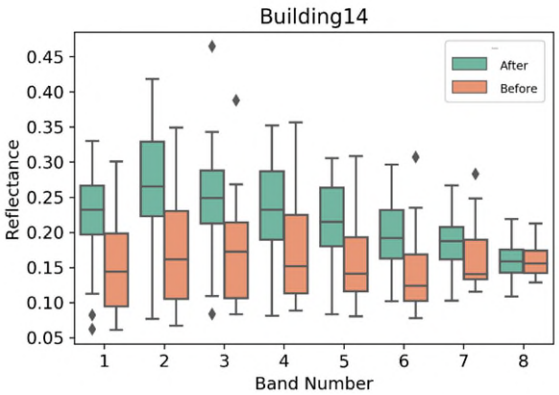

Building9
After
Before
Reflectance
Band Number

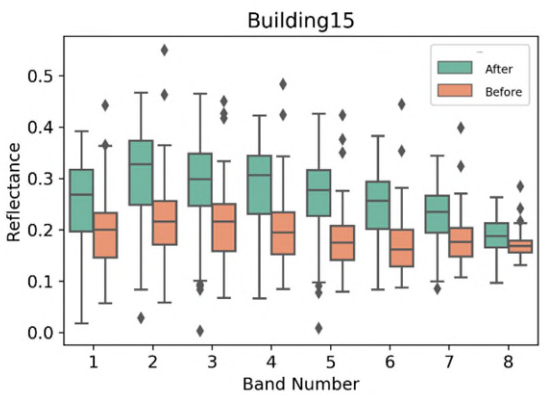

Building10
After
Before
Reflectance
Band Number

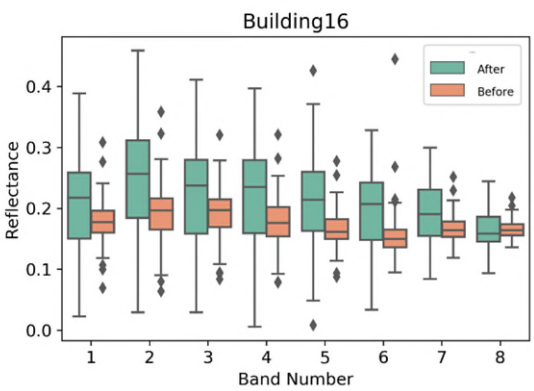

Building11
After
Before
Reflectance
Band Number


Building12
After
Before
Reflectance
Band Number


Building13
After
Before
Reflectance
Band Number


Building14
After
Before
Reflectance
Band Number


Building15
After
Before
Reflectance
Band Number


Building16
After
Before
Reflectance
Band Number

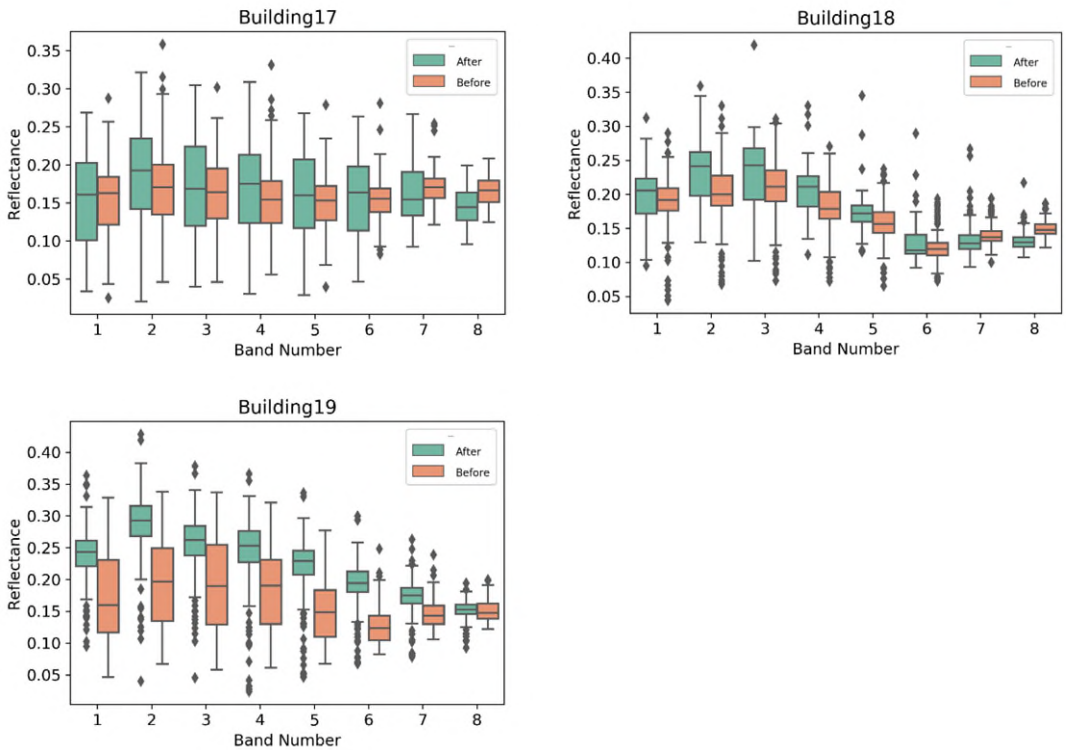


Figure 4. Box and Whisker plot showing changes in spectral reflectance across spectral bands for all 19 collapsed building using pre- and post-event imagery

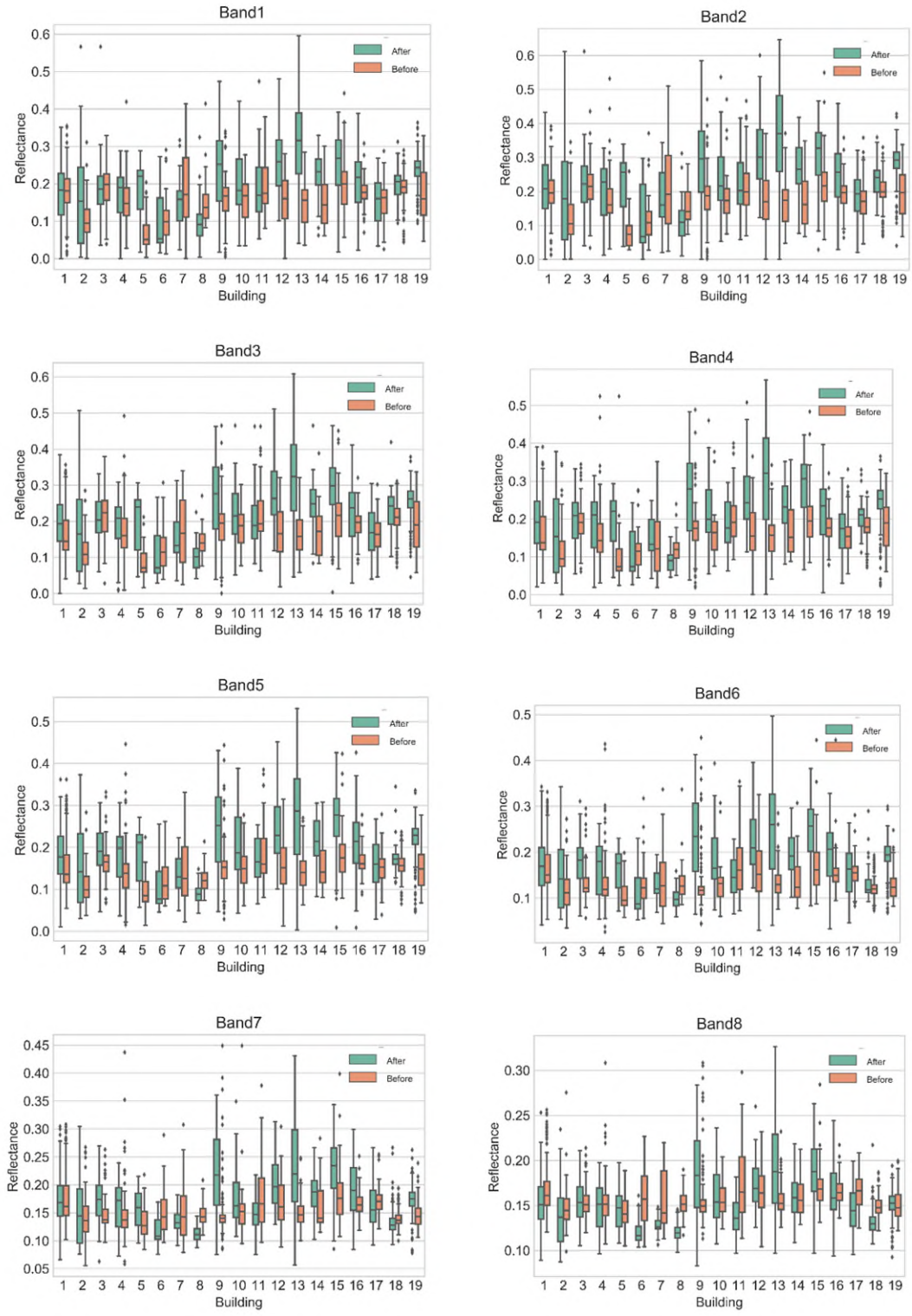


Figure 5 . Comparison of spectral reflectance across all 19 collapsed buildings for each spectral band.

When looking at spectral content in imagery, we often use combinations of spectral bands to differentiate objects in imagery. To compare intact and collapsed buildings, we calculate textural pattern using dissimilarity and homogeneity as shown in Figure 6. Dissimilarity index shows how much a pixel is different from its neihboring pixels; it is calculated by summing the differences in values between neighboring pixels and dividing by 2. Homogeneity index measures how much the variability of pixel values changes through out the building footprint.

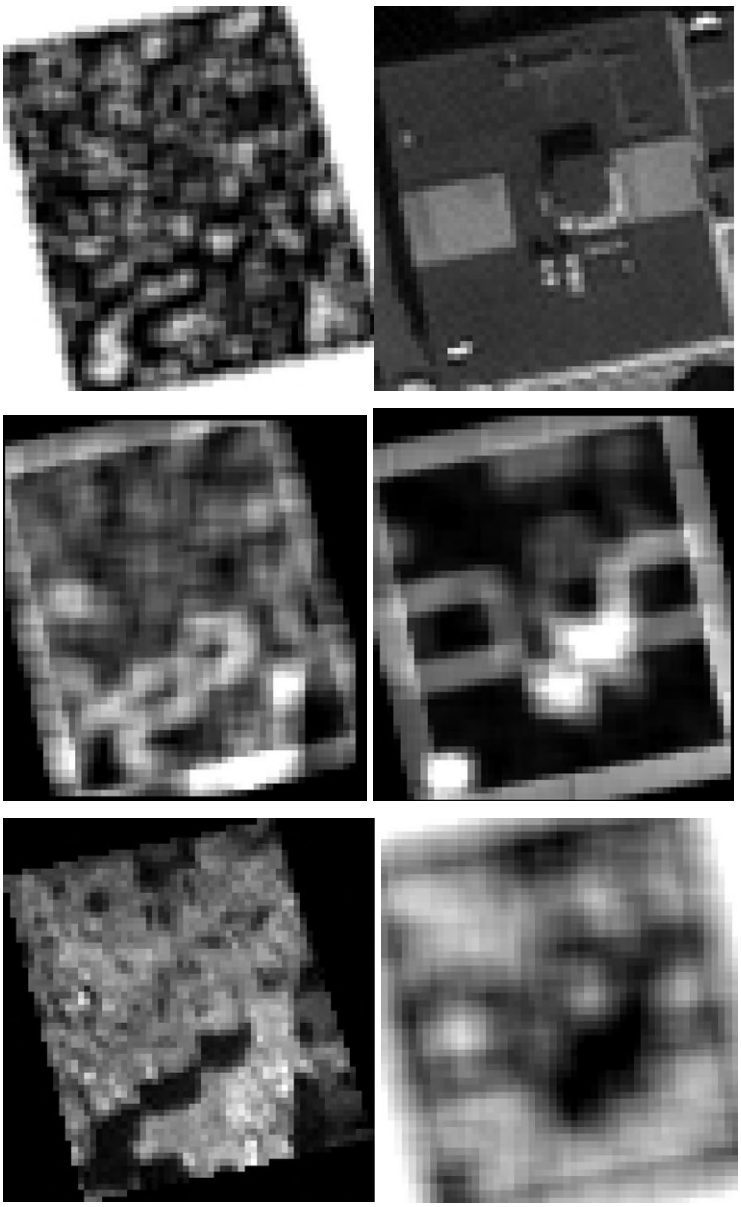


Figure 6. Panchromatic band, homogenity, and dissimilarity for an example collapsed building. Post-event is on the left, Pre-event is on the right.

Automated classification using xView imagery

U-net was originally proposed by [23] to solve biomedical imaging problems; however, soon after, it became popular in other fields of the computer vision research communities. The U-net style CNN architecture consists of contracting and expanding paths. High-resolution features in the contracting path are concatenated with up-sampled versions of global low-resolution features in the expanding path to help the network learn both local and global information. The contracting path contains padded 3 by 3 convolutions followed by ReLU non-linear activation layers. A 2 by 2 max pooling is applied after each two convolutional layers. After each down-sampling, the number of features is doubled. In the expanding path, a 2 by 2 up-sampling operation is used after each two convolutional layers, and the resulted feature map is concatenated to the corresponding feature map from the contracting path. At the final layer, a 1 by 1 convolution with linear output is applied to match the feature map with the number of classes (demolished building or intact building). Figure 2 shows the architecture of a U-net style convolutional network.

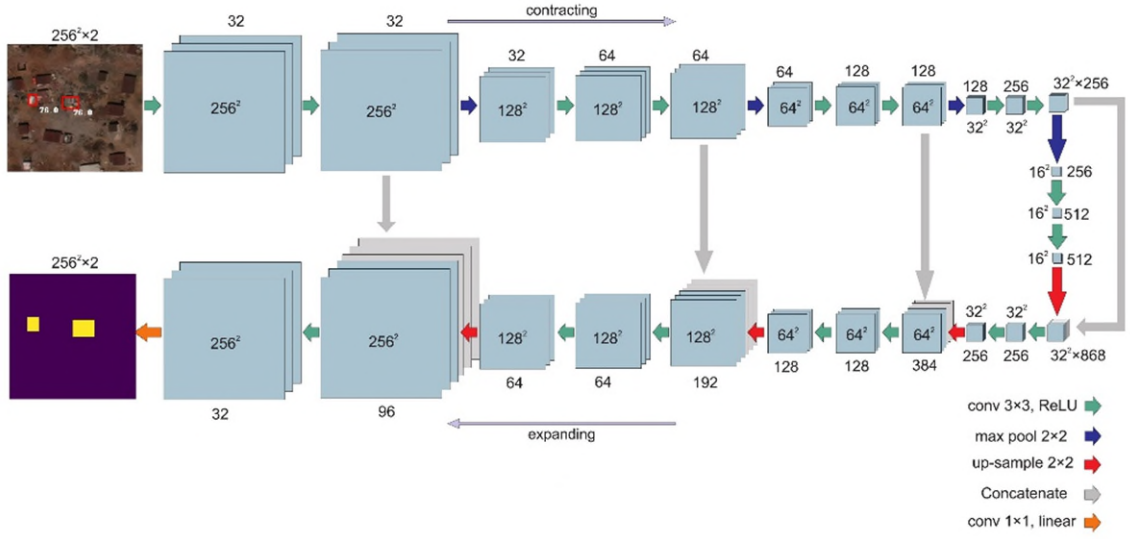


Figure 2. The U-net style convolutional neural networks used in this study. The input size of the dataset is 256 by 256.

In compare to the original U-net architecture, we have also used two regularizing operations to improve the training process: 1- batch normalization [24] and 2- dropout [25]. Batch normalization (BN) operation is used to reduce the amount by what the hidden unit values shift around. BN operation normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. An example for the BN utilization is when you train a network by only feeding black color cats but want to test it on white color cats. In this research, we have used BN at every convolutional layer. Dropout, like BN, is a regularization method that ignores some neurons during training phase randomly and therefore, the ignored neurons will not be considered in forward or backward pass. In a fully connected layer and during the training phase, neurons develop interdependency between each other that could result in over-fitting of the training data. By using dropout operation and ignore some of the trained neurons, we try to prevent the network from over-fitting.

Methodology and accuracy measurement

As discussed earlier, there are 1067 ‘demolished building’ and around 300,000 ‘building’ labels in the xView dataset. As this is a heavily unbalanced dataset, we have randomly selected 1067 ‘building’ labels to balance the dataset. This will make sure that the network statistics are not biased toward the ‘building’ class due to larger sample size. A biased network can assign higher probabilities to the majority class to avoid greater penalty (higher loss). However, in reality, data often exhibit class imbalance (e.g. intact building: demolished building), where some classes are represented by large number of pixels while other classes by a few [26]. Another step taken before feeding the network with training data was ‘data augmentation’. CNNs usually require large number of data to converge the training process, prevent over-fitting, and minimize the loss. To this end, it is a common practice in computer vision community to increase the number of samples

synthetically or augment the original data. Augmentation is a process in which a network is being fed a same image patch multiple times with different orientations. A single image is represented to the network as an array of RGB pixel values and if the orientation of this array change, the network assumes it as a new image. Flipping horizontally or vertically, rotating at a degree, crop randomly, and scale inward or outward are among common augmentation techniques to generate new images. In this study, as the original number of 'demolished building' labels is low (1067), we have used combination of abovementioned techniques to augment the dataset. Note that we have also augmented the 'building' labels after randomly selecting 1067 out of original xView dataset to maintain an unbiased approach toward both 'demolished building' and 'building' labels. In the xView dataset, there are 153 satellite images each of which at least includes one label of 'demolished building' (total number of 'demolished building' in these 153 images is 1067). Out of these 153 images, 130 were randomly selected and used for training and the remaining 23 for testing phases. Note that the size of each of these satellite images is about 3000 by 3000 pixels (each pixel has a resolution of 30 cm). For this study, we have divided each image into 256 by 256 pixels size patches as more GPU memory is required to store the feature maps with increase in image size. Then, we have augmented patches that have 'demolished building' and / or 'building' label(s) within them. Finally, we have 10,000 patches for training that overall have 13,099 and 12,168 'demolished building' and 'building' labels, respectively. Each patch could include multiple labels from both classes, only one label from one class, and nothing (to train the network that there could be a patch with no labels of interest). Figure 3 shows three examples of patches used to train the network. Same procedure was done for testing satellite images and 2,000 patches were generated with 1,243 and 1,103 labels for 'demolished building' and 'building' classes.

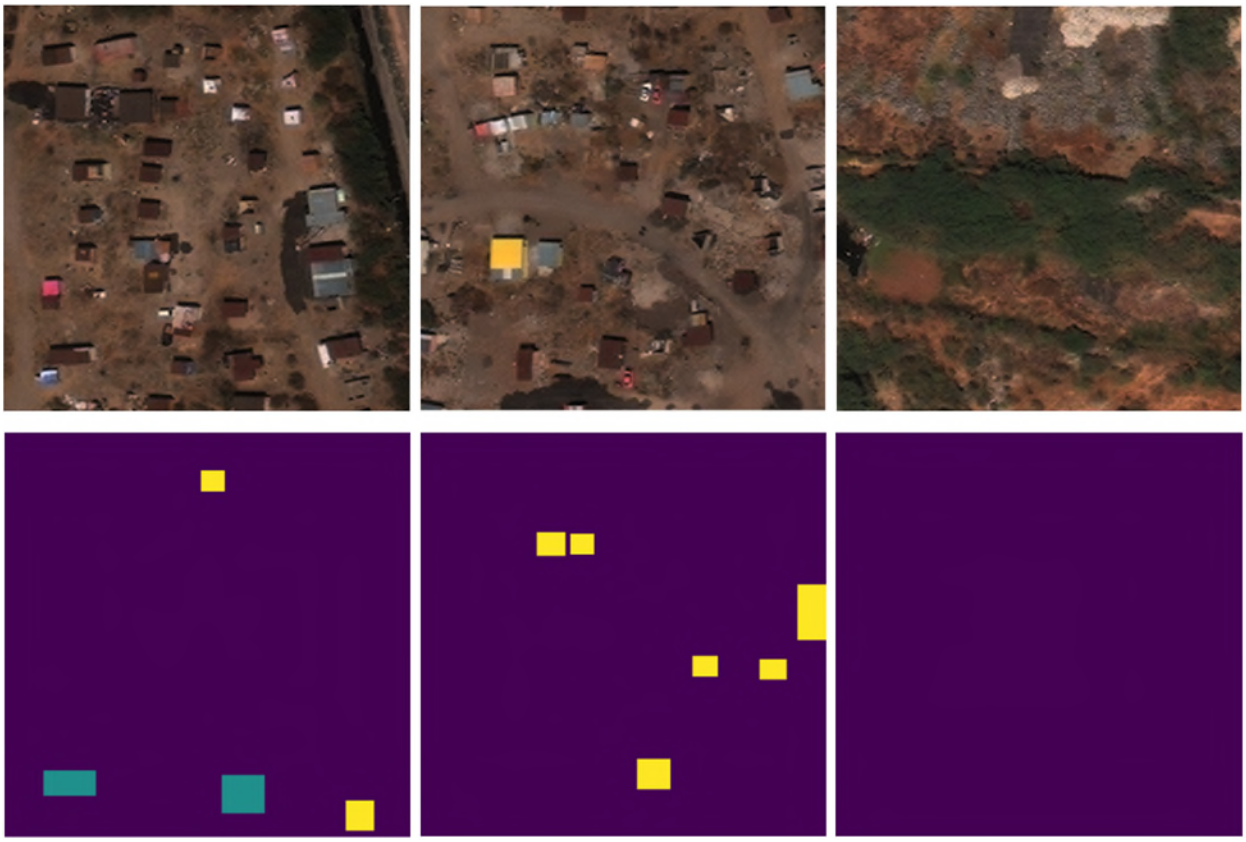


Figure 3. Three examples of input patches to the network at top row along with their corresponding label masks at the bottom row. The left patch included two labels of each 'demolished building'(yellow) and 'building' (blue) classes. The middle patch includes 6 labels of only 'demolished building' class. The right patch included none of the two classes.

Each single patch is normalized by its maximum RGB value before feeding into the network to increase the learning speed. The network is trained with 16 patches at each input batch and for 1000 epochs. The learning rate was set to 0.0001 to ensure the network optimizer would not be trapped in local minima. Stochastic gradient descent (SGD) optimization algorithm was used to minimize the loss function. SGD is one of the dominant techniques in CNNs optimization [27] and has proved to be effective in optimization of large-scale deep learning models [28]. Categorical

cross-entropy loss (Softmax loss) was used as the loss function to output the probability map over the two classes.

For measuring the accuracy of the network, after the training data is used to train the network, a confusion matrix is generated against the validation data. A confusion matrix is a table that summarizes the performance of a classifier by presenting the number of ground truth pixels for each class and how they have been classified by the classifier. There are several important accuracy measurements that can be calculated from a confusion matrix amongst them are: 1-Overall accuracy, 2-Producer's accuracy (sensitivity), and 3-User's accuracy. Overall accuracy is calculated as the ratio of the total number of correctly classified pixels to the total number of the test pixels. The overall accuracy is an average value for the whole classification method and does not reveal the performance of the method for each class. Producer's and User's accuracies are defined for each of the classes. The producer's accuracy also common as classifier sensitivity corresponds to error of omission (exclusion or false negative rate) and shows how many of the pixels on the classified map are labeled correctly for a given class in the reference data. Producer's accuracy is calculated as:

$$\text{Producer's Accuracy} = \frac{Number\ correctly\ identified\ in\ ref.\ plot\ of\ a\ given\ class}{Number\ actually\ in\ that\ ref.\ class} \qquad (1)$$

User's accuracy corresponds to the error of commission (inclusion or false positive rate) and shows how many pixels on classified map are correctly classified. User's accuracy is calculated as:

$$\text{User's Accuracy} = \frac{Number\ correctly\ identified\ in\ a\ given\ map\ class}{Number\ claimed\ to\ be\ in\ that\ map\ class} \qquad (2)$$

We also calculate and report specificity defined as $TN / TN + FP$, where TP, FP, and FN are the true positive, false positive, and false negative rates, respectively.

All pre-processing, training, and testing tasks are performed on Tufts University High Performance Cluster configured with 128 GB of RAM, 8 cores of 2.6 GHz CPU and an NVIDIA Tesla P100 GPU. All tasks are coded in Python language using various libraries including Numpy, Shapely, Keras, Tensorflow etc.

Result and discussion

The primary function of the trained U-net network herein is to segment the image. In other words, it will assign a probability value for each class to each pixel; the class with higher value of probability will be considered as the network prediction. All pixels within each individual label should be ideally classified as either of ‘demolished building’ or ‘building’ using a proper probability thresholding. As a common practice, if the probabilities of classes for a given pixel are less than 50%, then that pixel will be classified as none. Having that said, the network performance on testing (unseen) data was evaluated after reaching a loss value of 0.14 on training dataset. As the primary goal of this research was set to detect demolished buildings after a major natural hazard, Figure 4 exemplifies the network prediction for ‘demolished building’ class on one unseen patch that includes two labels of ‘demolished building’ and three labels of ‘building’. In the figure, closer the color to solid yellow, higher the probability for that class. As can be seen, the network can predict the pixels within the real ‘demolished building’ labels with higher probability in compare to the pixels within the ‘building’ labels. Note that the network performance is only evaluated on pixels within the labels, which are the ground truth for the network. There are also some pixels within the ‘building’ label mistakenly classified as ‘demolished building’ label. Considering the segmentation performance of the network on all labeled pixels, the sensitivity of the model to detect ‘demolished building’ is 60.3%. This low value partially could be due to the

fact that many of the xView labels represent an area larger than the demolished building footprint size. This can affect the spectral and textural information extraction process, which is supposed to only learn the 'demolished building' class pixels and not around it.

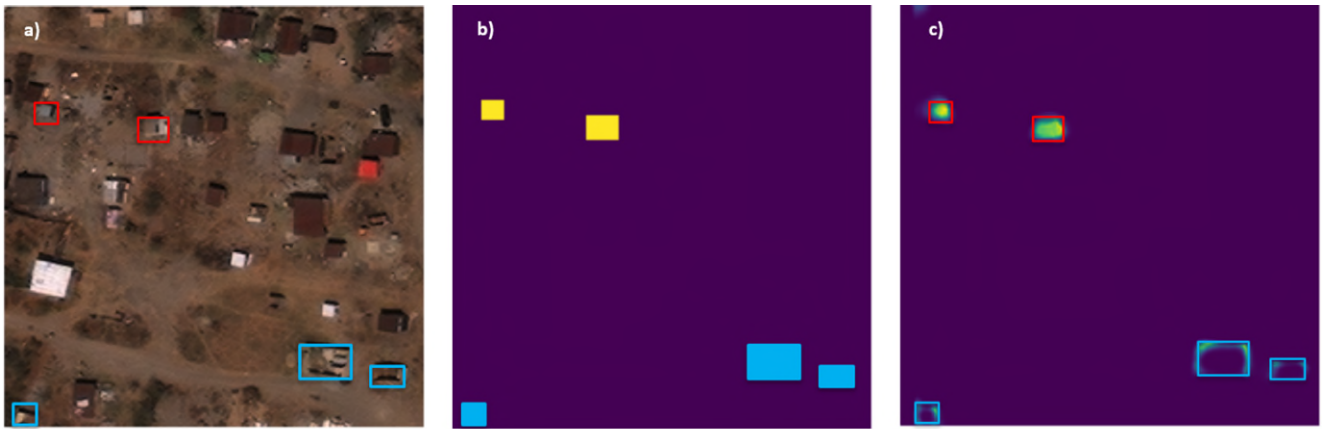


Figure 4. An example of the network prediction for 'demolished building' labels. Boxes in red are 'demolished building' labels and boxes in blue are 'building' labels; a) original patch b) input label mask; yellow is 'demolished building' and blue is 'building' classes ground truth c) network prediction for 'demolished building' labels; closer to yellow, higher the probability of the label.

To alleviate this issue and improve the final network results, we have considered the overall performance of the network on each label instead of each pixel. In other words, instead of evaluating the performance of the network in segmenting labeled pixels, we evaluate the network's performance in localizing the labels. To this end, if more than 30% of a given label's pixels are classified correctly, the whole label will be considered as one true positive case. (See discussion of the impact of different thresholds included below). If more than 30% of a 'building' label's pixels are classified as 'demolished building' class, then it is one false positive case for 'demolished building' class and vice versa. If less than 30% of a 'demolished building' label's

pixels are classified as such, then it is one false negative case for ‘demolished building’ class and vice versa. If less than 30% of a ‘building’ label’s pixels are classified as ‘demolished building’ class, then it is one true negative case for ‘demolished building’ class and vice versa. Figure 5 exemplifies a scenario for true positive case.

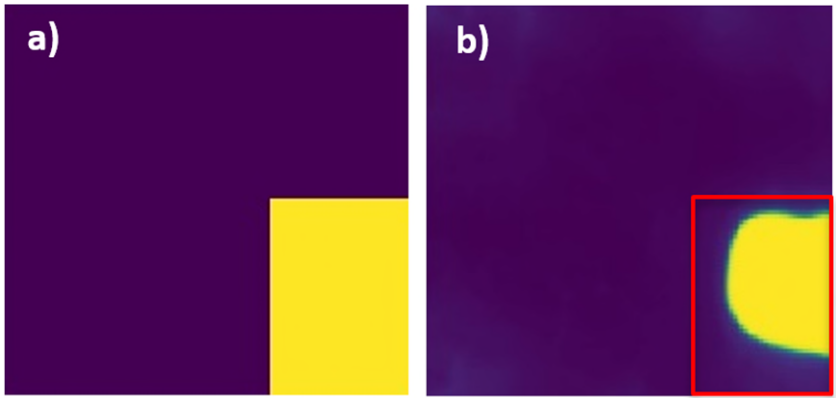


Figure 5. Redefining true positive scenario: a) ground truth b) model prediction considered as a true positive case as more than 30% of the label’s pixels are classified correctly.

After redefining the accuracy measurement definitions, the sensitivity of the model to detect ‘demolished building’ labels is 76%. Table 1 presents the sensitivity and specificity for both ‘demolished building’ and ‘building’ classes. As can be seen, the model sensitivity to detect ‘building’ class is much higher than detecting ‘demolished building’ class (96%); however, its specificity for ‘demolished building’ class is higher than the ‘building’ class.

Table 1. Network’s sensitivity and specificity to input classes.

	Sensitivity	Specificity
Intact	0.959	0.774
Demolished	0.760	0.970
Overall	0.854	0.866

The user's accuracy for the 'demolished building' class is 97%. In other words, if the network predicts a label as 'demolished building', there is 97% probability that outcome is correct. This value for 'building' class is 77. The detailed confusion matrix against all the labels in testing dataset is presented in Table 2. Out of 1243 'demolished building' labels, 281 are misclassified as 'building'; while out of 1103 labels for 'building' labels only 33 are misclassified as 'demolished building'. Overall, the network can predict the correct label with 85% accuracy.

Table 2. Confusion matrix against the labels in testing dataset.

Confusion Matrix		Ground Truth		
		Number of demolished label	Number of intact label	Total
Model Prediction	Number of demolished label	945	33	978
	Number of intact label	281	1058	1339
	Missed (classified as neither)	17	12	29
	Total	1243	1103	2346

Note that the threshold of 30% discussed earlier in redefining the accuracy measurement, was selected among 20%, 30% and 40%. Higher thresholding values (e.g. 40%) decreases number of false positive in expense of lower true positive as well. Lower thresholding values (e.g. 20%) increases number of true positive (slightly) in expense of higher false positive. Table 3 presents the details for these three thresholds. Choosing a proper threshold greatly depends on the application of the network and what accuracy measurement plays more important role in the decision-making process. If sensitivity is more important, lower threshold should be selected and if specificity is more important, higher threshold should be selected.

Table 3. Effect of thresholding on the model accuracy.

Threshold:	20%		30%		40%	
	True Positive	False Positve	True Positive	False Positve	True Positive	False Positve
Intact	0.965	0.235	0.959	0.226	0.905	0.214
Demolished	0.763	0.035	0.760	0.030	0.732	0.021
Overall	0.858	0.141	0.854	0.134	0.813	0.123

This trained network can be implemented very fast in time of a need. Usually after a natural hazard happens, it takes a couple of days to access the satellite imagery of the affected area. After some image pre-processing and geospatial analysis that could be done in a very short time, this trained network could be implemented to generate a map of demolished buildings. Depending on the computational power and size of the affected area, this process would only take minutes to accomplish. If ideally in rapid response community, we would have access to satellite imagery minutes after a disaster, the whole process to generate the demolished buildings map could happen fast and information could be used by the first responder at the affected area. Note that this framework has been constructed and trained with very high resolution (30 cm) optical satellite imagery from MAXAR company captured after disastrous events. We can improve and expand the framework by adding pre-events imagery or other types of the remotely sensed data such as Synthetic Apparatus Radar (SAR) or Light Detection and Range (LiDAR). We can train a network based on all these range of remotely sensed data and in case of need, we could test it on whatever data is available.

Project Data

The xView datasets is publically available. The World-View 2 imagery used for the 2017 Mexico earthquake was acquired from Digital Globe.

Bibliography

1. Rashidian, V., LG Baise, M Koch Detecting Collapsed Buildings After a Natural Hazard on Vhr Optical Satellite Imagery Using U-Net Convolutional Neural Networks IGARSS 2019-2019 IEEE International Geoscience, 2019.
2. Rashidian, V., Koch, M. and L.G. Baise (2018). Rapid earthquake-induced damage using satellite imagery and machine learning algorithms for the M7.1 Central Mexico Earthquake. 2018 SSA Annual Meeting.
3. Rashidian, V., Baise, L.G. and M. Koch (2018). Compiling a training data set for rapid detection of earthquake-induced building collapse using satellite imagery. AGU Fall Meeting.