On the Deployment of Post-Disaster Building Damage Assessment Tools using Satellite Imagery: A Deep Learning Approach

Shahrzad Gholami[1], Caleb Robinson[1], Anthony Ortiz[1], Siyu Yang[1], Jacopo Margutti[2],
Cameron Birge[1], Rahul Dodhia[1], Juan Lavista Ferres[1]
[1]*AI for Good Research Lab, Microsoft, Redmond, USA,*
[2]*510 an initiative of the Netherlands Red Cross, The Hague, The Netherlands*

***Abstract*—Natural disasters frequency is growing globally. Every year 350 million people are affected and billions of dollars of damage is incurred. Providing timely and appropriate humanitarian interventions like shelters, medical aid, and food to affected communities are challenging problems. AI frameworks can help support existing efforts in solving these problems in various ways. In this study, we propose using high-resolution satellite imagery from before and after disasters to develop a convolutional neural network model for localizing buildings and scoring their damage level. We categorize damage to buildings into four levels, spanning from not damaged to destroyed, based on the xView2 dataset's scale. Due to the emergency nature of disaster response efforts, the value of automating damage assessment lies primarily in the inference speed, rather than accuracy. We show that our proposed solution works three times faster than the fastest xView2 challenge winning solution and over 50 times faster than the slowest first place solution, which indicates a significant improvement from an operational viewpoint. Our proposed model achieves a pixel-wise F1 score of 0.74 for the building localization and a pixel-wise harmonic F1 score of 0.6 for damage classification and uses a simpler architecture compared to other studies. Additionally, we develop a web-based visualizer that can display the before and after imagery along with the model's building damage predictions on a custom map. This study has been collaboratively conducted to empower a humanitarian organization as the stakeholder, that plans to deploy and assess the model along with the visualizer for their disaster response efforts in the field.**

***Index Terms*—satellite imagery datasets, neural networks, image segmentation, building damage classification, natural disasters, humanitarian action**

I. Introduction

Natural disasters affect 350 million people each year causing billions of dollars in damage and were the main driver of hunger for 29 million people in 2021 [1]. Providing timely humanitarian aid to affected communities is increasingly challenging due to the growing frequency and severity of such events [2]. Impact assessment of natural disasters in a short time frame is a crucial step in emergency response efforts as it helps first responders allocate resources effectively. For example, dispatching aid, sending shelters, and allocating building material for reconstruction can be more efficient with estimates of where damaged buildings are, and how badly damaged they are.

Microsoft AI for Good/Humanitarian Action has collaborated with Netherlands Red Cross to use high-resolution satellite imagery from before and after natural disasters, delineated in the publicly available xBD dataset, to develop an end-to-end Siamese convolutional neural network that can localize buildings and score their damage level. Such a model is trained on historical disaster data and then applied on demand to identify damaged buildings during future disasters. Such AI and data-driven decision-aid tools can empower humanitarian organizations to take more informed actions at the time of disaster and allocate their resources more strategically during their field deployments. Throughout the course of our collaboration, extensive deployment experience shared by field experts and their valuable perspective as a stakeholder were instrumental in informing our empirical analysis of the model pipeline and will be vital in future assessments of the model performance in the fields when actual disasters happen.

In 2019, the xView2 challenge and the xBD dataset were announced at the Computer Vision for Global Challenges Workshop at the Conference on Computer Vision and Pattern Recognition to benchmark automated computer vision capabilities for localizing and scoring the degree of damage to buildings after natural disasters [3]. In this challenge, participants had to train their model offline and upload their predictions for evaluation and display on the public leaderboard based on a single unlabeled test dataset, which they could download. While this challenge provided a great opportunity for AI researchers to weigh in on damage assessment tasks, it assumed no constraints on the level of computational resources available to participants for model training and did not strictly prevent the potential hand-labeling and use of the test datasets in the training phase. The winning solutions used large ensembles of models, and although they perform well on the test set, they were not optimized for inference runtime and require a prohibitively large amount of compute resources to be run on large amounts of satellite imagery on demand during disaster events. For example, the first-place winner proposed an ensemble of four different models, requiring 24 inference passes for each input.

In this study, we propose a single model which predicts both building edges and damage levels and that can be run efficiently on large amounts of input imagery. The proposed multitask model includes a building segmentation module and a damage classification module. We use a similar model architecture proposed by previous studies on building damage assessment [4], [5]; however, we use a simpler encoder and do not include attention layers. We evaluate the performance of our model extensively for several different splits of the dataset to assess its robustness to unseen disaster scenarios. From an operational perspective, the model's runtime is of paramount importance. Thus, we benchmark the inference speed of our model against the winning solutions in the xView2 competition and the existing models deployed by our stakeholder. We show that our model works three times faster than the fastest xView2 challenge winning solution and over 50 times faster than the slowest first place solution. The baseline solution available to our stakeholder consists of two separate models for building segmentation and damage classification [6]. We were able to show that our proposed approach works 20% faster than the baseline model available to the stakeholder and also conducts the task in an end-to-end and more automated way, which can improve their field operations and deployment.

Finally, we develop a web-based visualizer that can display the before and after imagery along with the model's building damage predictions on a custom map. This is an important step in deploying a model for real-world use cases. Even a perfect building damage assessment model will not be practically useful if there is not a mechanism for running that model on new imagery and communicating the results to decision-makers that are responding to live events. A web-based visualizer allows *anyone* to see both the imagery and predictions without GIS software for any type of disaster.

II. Related Work

Convolutional neural networks (CNN) have been used for change detection tasks in satellite imagery for disaster response and other domains including but not limited to changes in infrastructures. [7] proposed using pre-trained CNN features extracted through different convolutional layers and concatenation of feature maps for pre- and post-event images. The authors used pixel-wise Euclidean distance to compute change maps and thresholding methods to conduct classification. [8] leverages hurricane Harvey data, in particular, to train CNNs to classify images as damaged and undamaged. While they report very high accuracy numbers, they did not focus on detecting building edges and used a binary damage scale at the image-frame level. A Siamese CNN approach was proposed in [9] to extract features directly from the images, pixel by pixel. To reduce the influence of imbalance between changed and unchanged pixels, the authors used weighted contrastive loss. The unique property of the extracted features was that the feature vectors associated with changed pixel pairs were far away from each other in the feature space, whereas the ones of unchanged pixel pairs were close. Fully convolutional Siamese networks for change detection were introduced in [4] and were proposed by other studies as well [10], [11].

In [4], convolutional Siamese networks are trained end-to-end from scratch using only the available change detection datasets. The authors proposed fully convolutional encoder-decoder networks that use the skip connection concept. [12] presented an improved UNet++ model with dense skip connections to learn multiscale and different semantic levels of visual feature representations. Attention layers have been proposed for general change detection networks [13] as well as building damage assessment tasks as presented in [5]. Also, [14] proposes an attention-based two-stream high-resolution network to unify the building localization and classification tasks into an end-to-end model via replacing the residual blocks in HRNet [15] with attention-based residual blocks to improve the model's performance. RescueNet, an end-to-end model that handles both segmentation and damage classification tasks was proposed in [16]. It was trained using a localization aware loss function, that consists of a binary cross-entropy loss and dice loss for building segmentation and a foreground-only selective categorical cross-entropy loss for damage classification. [6] explored the applicability of CNN-based models under scenarios similar to operational emergency conditions with unseen data and the existence of time constraints. [17] proposed a dual-task Siamese transformer model to capture non-local features. Their model adopts transformers as the backbone rather than a convolutional neural network and relies on a lightweight decoder for the downstream tasks.

Graph-based models have been explored in [18] for building damage detection solutions to capture similarities between neighboring buildings for predicting the damage. They used the xBD dataset for cross-disaster generalization. While their proposed approach showed some advantages in terms of accuracy, it did not consistently outperform the Siamese CNN model in terms of F1 score, which would be a more appropriate metric for imbalanced datasets. Furthermore, [19] proposed BLDNet based on a Siamese CNN combined with a graph node classification approach to be trained in a semi-supervised manner to reduce the number of labeled samples needed to obtain new predictions. They benchmarked their approach with a semi-supervised multiresolution autoencoder and showed performance improvements. The extremely imbalanced distributions of the building damages are addressed in [20] by supplementing the architecture with a new learning strategy comprising normality-imposed data-subset generation and incremental training. However, they propose a two-step solution approach for building localization and damage classification. Self-supervised comparative learning approach has been studied in [21] to address the task without the requirement of labeled data. Their proposed approach is an asymmetric twin network architecture evaluated on the xBD dataset.

In this study, we propose a Siamese approach inspired by [4], [5] where UNet architecture is used for the building segmentation task and UNet's encoders with shared parameters for pre-disaster and post-disaster imagery, are used to score building damage levels via an end-to-end approach. Furthermore, we also evaluate the performance of our model in various scenarios that resemble operational emergency conditions. Web visualizer tools have been developed for other *specific* domains like data-driven wildfire modeling [22] and fire inspection prioritization [23] in the past. Our developed web visualizer allows imagery and prediction layers visualization for any disasters where before and after disaster satellite images are available.

III. Data

In this study, we use the xBD dataset introduced in [24] as a new large-scale dataset for the advancement of change detection and building damage assessment for humanitarian assistance and disaster recovery research. This dataset has been sourced from the Maxar/DigitalGlobe Open Data Program. It covers 19 different disasters from around the world for which there exists high-resolution (<0.8m/px resolution) imagery. The disaster types include flood, wind, fire, earthquake, tsunami, and volcano. The entire dataset contains 22,068 image *tiles* of 1024×1024 pixels that cover a total of 45,361.79 sq. km. There are 850,736 building polygons available along with a damage level label that indicates: no-damage, minor-damage, major-damage, and destroyed. The breakdown of the number of polygons for pre-disaster images across different disasters is shown in Table I. Figure 1 and Figure 2 show some examples of pre- and post-disaster image frames from the xBD dataset. See figure 5 for legend.

Name/Location	Type	# of polygons
Palu, Indonesia	Earthquake/Tsunami	55,789
Mexico City, Mexico	Earthquake/Tsunami	51,473
Nepal	Flood	43,265
Hurricane Harvey, USA	Flood	37,955
Hurricane Michael, USA	Wind	35,501
Hurricane Matthew, USA	Wind	23,964
Portugal	Wildfire	23,413
Moore, OK	Wind	22,958
Santa Rosa, CA	Wildfire	21,955
SoCal, CA	Wildfire	18,969
Sunda Strait, Indonesia	Earthquake/Tsunami	16,847
Joplin, MO	Wind	15,352
Tuscaloosa, AL	Wind	15,006
Midwest USA	Flood	13,896
Hurricane Florence, USA	Flood	11,548
Woolsey, CA	Wildfire	7,015
Pinery, Australia	Wildfire	5,961
Lower Puna, HI	Volcanic eruption	3,410
Guatemala	Volcanic eruption	991

TABLE I
Disaster events in the xBD dataset.

IV. Model Architecture

We propose a deep learning model that conducts both building segmentation and damage classification tasks via a single pipeline. Our approach has some similarities to the proposed method in [5]. However, our architecture is less complex as we do not incorporate any attention layers in the model. One module of our model is based on the UNet architecture proposed in [25], which obtains the building segmentation mask. A single image frame is fed to the fully convolutional UNet model where local information is captured via encoder-decoder structures and global information is captured via several skip connections. In the damage assessment scenario, we have a pair of pre- and post-disaster image frames, which are given as inputs separately to the UNet module of our proposed method using shared weights. We use the embedding layers from the encoder part of the UNet architecture for pre- and post-disaster images to learn about the changes. In other words, the second module of our model is a separate decoder that conducts a damage classification task on the subtracted embedding layers using several convolutional layers. This idea is based on the approach proposed in [4]. Figure 3 demonstrates the overall schema of the architecture. Our UNet architecture has five convolution blocks for the encoder part and four convolution blocks for the decoder. Each downsampling block consists of convolution, batch normalization, ReLU, and max-pooling layers. Each upsampling block consists of upsampling with bilinear interpolation, convolution, batch normalization, and ReLU layers. For the damage decoder, the same upsampling blocks apply to subtracted and concatenated representations at each step. The details of the layers can be found in our code repository made publicly available[1]. The output of the damage classification mask has five channels for four damage levels and one background label. We use weighted binary cross-entropy loss for building segmentation and multi-label cross-entropy loss for damage classification. In the building

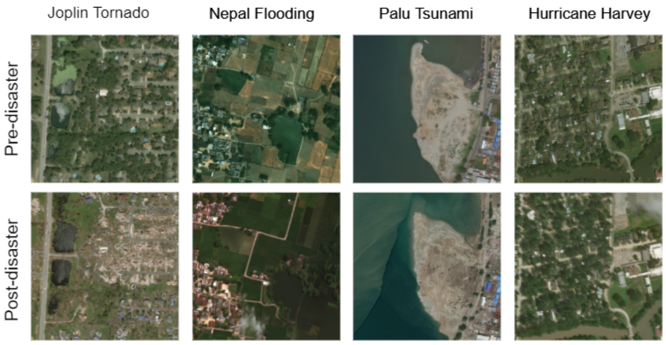
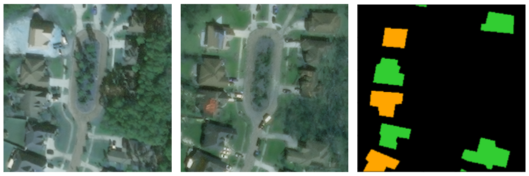


Fig. 1. Imagery samples from different disasters from DigitalGlobe.



(a) Pre-disaster (b) Post-disaster (c) Ground Truth

Fig. 2. Imagery samples with polygons showing building edges and colors showing damage level.

[1] https://github.com/microsoft/building-damage-assessment-cnn-siamese

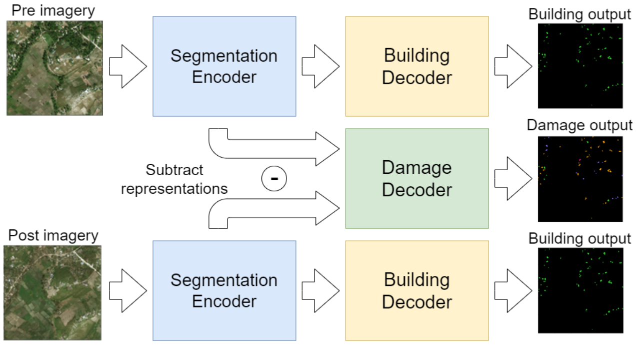


Fig. 3. We use a Siamese U-Net model architecture where the pre- and post disaster imagery are fed into an encoder-decoder style segmentation model (U-Net) with shared weights (blocks with the same color in the figure share weights). The features generated by the segmentation encoder from both inputs are subtracted and passed to an additional damage classification decoder that generates per-pixel damage level predictions. The weights of the damage classification decoder can be fine-tuned for specific disaster types, while relying on building segmentation output from the building decoder.

segmentation loss function shown in equation 1, $\omega_{s,1}$ and $\omega_{s,0}$ denotes weights on building pixels and background pixels, respectively. Subscript s denotes the segmentation task. y_s is the ground truth label for each pixel and p_s is the predicted probability. For both pre- and post-disaster image frames, loss functions $\mathcal{L}_{s_{pre}}$ and $\mathcal{L}_{s_{post}}$ are defined similarly and the UNet model has shared weights across these two components.

$$\mathcal{L}_{s_{pre}} = \mathcal{L}_{s_{post}} = -(\omega_{s,1} y_s \log p_s + \omega_{s,0} y_s \log (1 - p_s)) \tag{1}$$

In equation 2, $\omega_{d,c}$ denotes weight on each damage class c. We use subscript d to denote the damage classification task. y_d is the damage ground truth label for each pixel and p_d is the predicted probability. The damage loss, $\mathcal{L}_{dmg}$, is calculated only when a pixel is predicted as a building class or $\hat{y}_s == 1$.

$$\mathcal{L}_{dmg} \quad = -\textstyle\sum_{c=1}^{5} \omega_{d,c}(y_d(c) \log p_d(c)), \text{ if } \hat{y}_s == 1 \tag{2}$$

Equation 3 indicates the combined weighted loss function for the tasks along with their corresponding weights.

$$\mathcal{L}_{total} \quad = \omega_{s_{pre}} \mathcal{L}_{s_{pre}} + \omega_{s_{post}} \mathcal{L}_{s_{post}} + \omega_d \mathcal{L}_{dmg} \tag{3}$$

V. Experiments and Results

For our first experiment, we divide each disaster's tiles available in the xBD dataset based on original tiles of 1024x1024 pixels into train/validation/test splits at the ratio of 80:10:10 randomly, to train and evaluate the performance of our model. Based on this way of splitting the dataset, it is possible to have different tiles from the *same* disaster incident across the training, validation, and test sets. To reduce the size of the input images, we further crop each tile into 20 patches of 256x256 pixels. The number of final patches in the train/validation sets is 176700:22220. We conduct tile-wise normalization on the pre-disaster and post-disaster imagery separately. We also apply random horizontal and vertical flipping during the training to reduce overfitting.

We observed that it is quite challenging to train the entire model from scratch for both tasks simultaneously as the performance of the building segmentation step impacts the performance of the damage classification task significantly. As such, we train the model sequentially based on two different sets of weights. First, we train the building segmentation module by setting the weight for damage classification as zero and setting the weights for the UNet in the loss function equal to 0.5 for both pre-disaster and post-disaster building segmentation tasks. We also set weights for building pixels equal to 15 and background pixels equal to 1 as there is a significant imbalance between the number of pixels across these two classes. In other words, $[\omega_{s_{pre}}, \omega_{s_{post}}, \omega_d] = [0.5, 0.5, 0]$ and $[\omega_{s,c=0}, \omega_{s,c=1}] = [1, 15]$. Label $c = 0$ denotes background pixels and label $c = 1$ denotes building pixels.

Once we get reasonable performance on the validation set for the first task, we freeze the parameters of the UNet and we start training the model for the second task, i.e., damage classification. Thus, we set the weights in the loss function for pre-disaster and post-disaster segmentation task as zero and we set the damage classification task equal to 1. Due to high imbalance across different damage classes, we assign higher weights to the major-damage class (label=3) and destroyed class (label=4). In other words, $[\omega_{d,c=0}, \omega_{d,c=1}, \omega_{d,c=2}, \omega_{d,c=3}, \omega_{d,c=4}] = [1, 35, 70, 150, 120]$ for the damage classification task and $[\omega_{s_{pre}}, \omega_{s_{post}}, \omega_d] = [0, 0, 1]$ for building segmentation. Label $d = 0$ denotes background pixels and labels $d = 1$ to $d = 4$ denotes damage levels scaled from not-damaged to destroyed.

Since our model handles two tasks, we demonstrate performance results separately on each task. The performance results for the tile-wise random split are demonstrated in the first row of Table III where the model is evaluated both on the validation set and test set. Columns in the table are named BLD-1, DMG-0, DMG-1, DMG-2, DMG-3, and DMG-mean. The BLD-1 column denotes the F1 score for class 0, which indicates building pixels. DMG-0, DMG-1, DMG-2, and DMG-3 indicate pixel-wise F1 scores for no-damage, minor-damage, major-damage, and destroyed classes, respectively. DMG-mean denotes the harmonic F1 score across all damage levels computed based on the following equation.

$$F1_{dmg} = \frac{4}{\sum_{c=1}^{4} \dfrac{1}{F1_c + \epsilon}} \tag{4}$$

As shown in the columns for extreme classes of not damaged and destroyed, i.e., DMG-0 and DMG-3 in Table III, we observe superior performance results compared to columns DMG-1 and DMG-2 for the minor-damage and the major-damage classes due to the strength in the signal for those extreme classes. We used one Nvidia Tesla V100 GPU with 32G Memory to train the model and it took 6 days for the training to complete. Adam optimization algorithm was used and learning rate and batch size were set at 0.001 and 32.

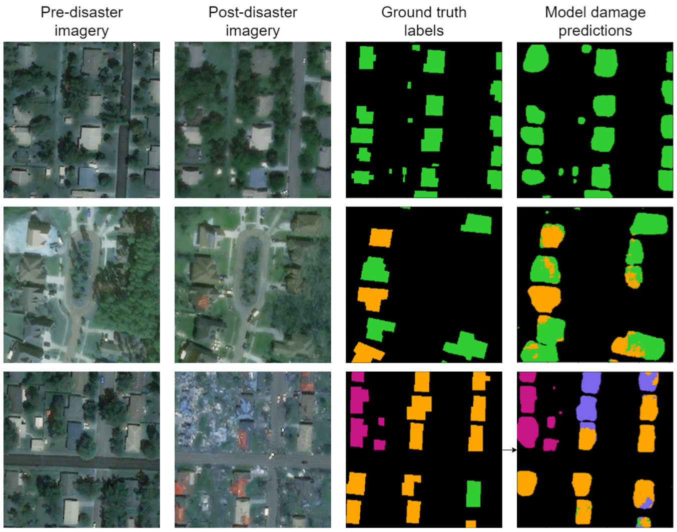


Fig. 4. Example model predictions at three locations with varying amounts of observed damage.

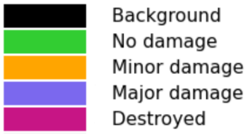


Fig. 5. Legend for damage level colors shown in Figure 2 and 4

Figure 4 demonstrates the predicted polygons for buildings along with their predicted labels. In the second row, the model was able to capture two missing buildings in the ground truth mask. Green, orange, purple and dark pink colors are used to indicate no-damage, minor-damage, major-damage and destroyed classes respectively. See figure 5 for legend.

The baseline model available to our stakeholder shows F1 score of 0.64 on the test set for building segmentation, which is inferior to our result of 0.74, shown in Table III. For the stakeholder's baseline damage classification performance, we do not have access to the results for a comparable data split to report here. We also show our model's performance against the baseline model presented in [26] in Table II. Our proposed solution demonstrates a significant improvement in damage classification task.

Model	BLD	DMG
Baseline	0.79	0.03
Ours	0.74	0.58

TABLE II
Comparison with the baseline model presented in [26], both results are based on training models on the xBD tier 1 dataset

VI. Model Robustness to Unseen Disasters

To assess the robustness of the model performance to unseen disasters, we conduct four additional experiments outlined in Table III. To that end, each time, we leave either the Joplin tornado or the Nepal flooding out for testing purposes and we train and validate the model based on the random split of the remaining data. Additionally, to see the impact of training damage classification only based on a *specific* type of disaster, we conduct two additional experiments: (I) when damage classification is trained only on wind-caused data, and (II) when damage classification is trained only on flood disasters. In both cases, the building segmentation module is trained on 90% of the entire training data; not on a specific disaster type as in the damage classifier module. In Table III, the second row shows the results for the case when we leave out the Joplin tornado for testing purposes and use a random split of the remaining data for training and validation for both building segmentation and damage classification tasks. For this unseen disaster, the harmonic mean of F1 scores on the test set drops by 4% compared to the completely random split of the dataset. The drop in the performance is more significant, 0.54%, when we leave Nepal flooding out as outlined in the fourth row of Table III. Regression in performance is also notable for the building segmentation task for Nepal flooding. This observation can be associated to the geographical distribution of the data. Unlike Nepal flooding, the majority of the disasters in the dataset, used in the training phase, are concentrated around North and Central America, which could explain the dramatic decrease in the F1 score when testing the model on a completely new geographical region. Test set results outlined in rows three and five of Table III demonstrate that training the damage classifier on the specific type of disasters boosts the performance when testing on a completely unseen disaster event.

VII. Inference Speed Benchmarking

Inference speed (specifically, the number of pixels/second that a model can process) is an important property of models that will be deployed to run on imagery collected from future disasters. Slow models will result in larger compute costs and potentially delayed results in time-sensitive disaster response applications.

We benchmark the inference speed of the top-ranked solutions on the xView2 challenge with our proposed model and find that our proposed model is three times faster than the fastest winning solution and over 50 times faster than the slowest first place solution. Table IV shows the performance results of each solution (except for the 4th place solution which was not reproducible). To benchmark each solution we use the following setup:

- A NC6 virtual machine instance on Microsoft Azure which contains a Tesla K80 GPU.
- The single input inference script provided in each solution's code release from the official "DIUx-xView" GitHub account. If the inference script did not contain a flag for enabling GPU acceleration we modified it to use the GPU for model inference.
- Three pre- and post-disaster inputs from the xBD dataset.

Experiment	Train	Test	BLD-1	DMG-0	DMG-1	DMG-2	DMG-3	DMG-mean
Random splits	80% at random	10% at random	0.74	0.89	0.43	0.54	0.73	0.60
Joplin held out	90% of non-Joplin	Joplin only	0.76	0.89	0.50	0.36	0.81	0.56
Joplin held out (wind only damage classifier)	90% of non-Joplin	Joplin only	0.74	0.89	0.42	0.54	0.77	0.60
Nepal held out	90% of non-Nepal	Nepal only	0.63	0.42	0.17	0.23	0.02	0.06
Nepal held out (flood only damage classifier)	90% of non-Nepal	Nepal only	0.64	0.54	0.12	0.27	0.07	0.14

TABLE III

PIXEL-WISE F1 SCORE ACROSS VARIOUS SPLITS OF THE XBD DATASET. WE TEST GENERALIZATION PERFORMANCE OF MODELS ON THE JOPLIN WIND AND NEPAL FLOODING EVENTS IN TWO SETTINGS: ONE IN WHICH WE TRAIN ON *all* AVAILABLE DATA THAT IS NOT FROM THE SPECIFIC EVENT, AND ANOTHER SETTING IN WHICH WE TRAIN THE DAMAGE CLASSIFICATION DECODER ON OTHER WIND-ONLY EVENTS (FOR TESTING ON THE JOPLIN EVENT) AND OTHER FLOOD ONLY EVENTS (FOR TESTING ON THE NEPAL FLOODING EVENT).

- The same Python virtual environment for all experiments to remove the effect of different packages on the performance.

Additionally, the inference times reported in Table IV include the file I/O, model loading, pre-processing, and post-processing costs associated with each approach and therefore represent an upper bound on the time taken to process any given 1024×1024 input (i.e. when running such approaches over large amounts of input, the models would only need to be loaded from the disk a single time).

As previously discussed, the xView2 challenge[2] encouraged participants to optimize for leaderboard performance instead of throughput. As such, many of the top-placed solutions used techniques such as ensembling and test time augmentation, as well as larger, more complex models in order to improve their performance at the cost of inference speed. The top-performing solution, for instance, consists of an ensemble of 12 models that are run 4 times for each input (test time augmentation with 4 rotations). These solutions are prohibitively costly to run on large inputs. For example, the Maxar Open Data program released $\sim 20,000\text{km}^2$ of pre- and post-disaster imagery covering areas impacted by Hurricane Ida in 2021. Assuming the inference times from Table IV, 0.3m/px spatial resolution of the input imagery and $0.9/hr cost of running a Tesla K80 (based on current Azure pricing), the first place solution would cost $6,500 to run, while our solution would only cost $100 to run. In this case, our solution would generate results for the area affected by Hurricane Ida in 4.7 days while the first place solution would take up to 301.4 days using a single NVIDIA Tesla K80 GPU.

Finally, we benchmark our proposed solution in an optimized setting compared to the above setting: we load data with a parallel data-loader (vs. loading a single tile on the main thread), we run pre- and post-processing steps on the GPU, we maximize the amount of imagery that is run through the model at once (vs. running on a single 1024×1024 tile of imagery), and we use the most recent version of all relevant packages (vs. the earliest version pinned in the environments from the xView2 solution repositories). Here, we find that our model is able to process 612.29 square kilometers per hour compared to 89.35 square kilometers per hour under the same assumptions in the previous setup despite using the same hardware. In this case, our model could process the Hurricane Ida imagery in 2 days at a cost of $14.7. The stakeholder's baseline solution's speed is 1000 square kilometers per hour on Azure NC12 GPU. We project our runtime to be 20% faster than their baseline solution on a similar GPU.

Method	Inference time (s)	sq. km/hr
xView2 1st place	245.75 (0.73)	1.38
xView2 2nd place	121.03 (0.36)	2.81
xView2 3rd place	108.21 (0.6)	3.14
xView2 4th place	not reproducible	not reproducible
xView2 5th place	10.94 (0.06)	31.07
Our method	3.8 (0.02)	89.35

TABLE IV

COMPARISON OF BUILDING DAMAGE MODEL INFERENCE TIMES ON A SINGLE 1024X1024 PIXEL TILE FOR DIFFERENT METHODS USING A SINGLE TESLA K80 GPU (ON AN AZURE NC6 MACHINE). TIMES ARE IN SECONDS AND ARE AVERAGED OVER THREE RUNS WITH A STANDARD DEVIATION IN PARENTHESES. THE RESULTS FOR THE WINNING XVIEW2 SOLUTIONS ARE REPRODUCED THROUGH THE OFFICIAL GITHUB REPOSITORIES PUBLISHED FOR EACH, WHERE THE ONLY MODIFICATIONS TO THE ORIGINAL CODE WAS TO ENABLE GPU PROCESSING FOR EACH INFERENCE SCRIPT. THE RIGHTMOST COLUMN SHOWS THE INFERENCE SPEED IN TERMS OF (SQ. KM)/HR ASSUMING A 0.3M/PIXEL INPUT SPATIAL RESOLUTION.

VIII. WEB VISUALIZER TOOL

In contrast to standard vision applications, semantic segmentation models that operate over satellite imagery need to be applied over arbitrarily large *scenes* at inference-time. As such, distributing the imagery and predictions made by such models is non-trivial. First, high-resolution satellite imagery *scenes* can be many gigabytes in size, difficult to visualize (e.g. requiring GIS software and normalization steps), and may require pre-processing to correctly align temporal samples. Second, the predictions from a building damage model are strongly coupled to the imagery itself. In other words, only distributing georeferenced polygons of where damaged buildings are predicted to be is not useful in a disaster response setting. The corresponding imagery is necessary to interpret and perform quality assessment on the predictions.

Considering these difficulties, we implement a web-based visualizer to distribute the predictions made by our model over satellite image scenes. This approach bypasses the need for any specialized GIS software, allowing any modern web-browser

[2] Most machine learning competitions follow a similar format, whereby participant solutions are *only* ranked in terms of the held-out test set performance.

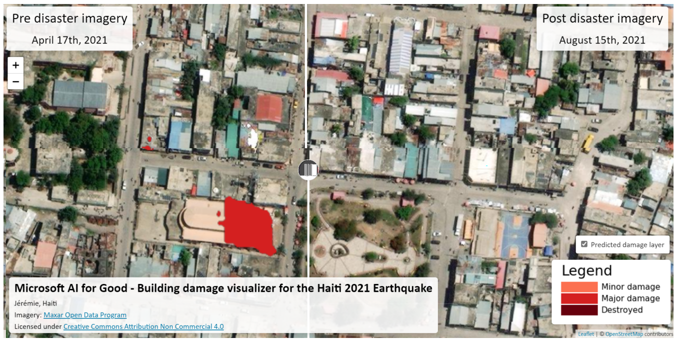

Pre disaster imagery
April 17th, 2021
+
−
Post disaster imagery
August 15th, 2021
☑ Predicted damage layer
Legend
Minor damage
Major damage
Destroyed
Microsoft AI for Good - Building damage visualizer for the Haiti 2021 Earthquake
Jérémie, Haiti
Imagery: Maxar Open Data Program
Licensed under Creative Commons Attribution Non Commercial 4.0
Leaflet | © OpenStreetMap contributors

Fig. 6. Screenshot of the building damage visualizer instance for the August, 2021 Haiti Earthquakes. The left side of the map interface shows the pre-disaster imagery while the right side shows the post-disaster imagery. The slider in the middle of the interface allows a user to switch between the pre- and post-disaster layers to quickly see the difference in the imagery. Finally, the building damage predictions are shown as polygons with varying shades of red corresponding to increasing damage. The visibility of these predictions can be toggled in the interface so that a user can see the underlying imagery.

to view the imagery and predictions, and doesn't require users to have any formal GIS experience as all imagery is pre-rendered. Specifically, users can:

1) Toggle back and forth between the pre- and post-disaster imagery to easily see the differences;
2) Change the visibility of the damage predictions to see the extent of the damage;
3) Show standard layers (e.g. OpenStreetMap or Esri World Imagery) for additional spatial context.

This is implemented with open-source tools including: GDAL[3], leaflet[4], and Docker[5].

An instance of our visualizer is shown in Figure 6 for a scene from Jeremie, Haiti after the Haiti Earthquake in August, 2021. The tower of the Cathedral of Saint Louis Roi of France (middle of the scene) is classified as damaged by the model and can be seen to be destroyed. The code for running inference with our final building damage model, as well as setting up an instance of the building damage visualizer tool is publicly available [6].

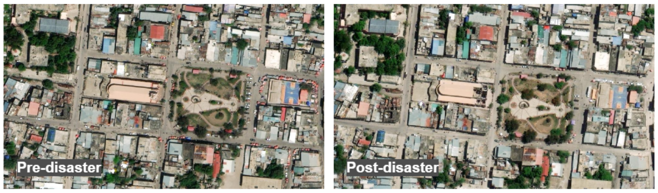


Fig. 7. Full screenshots of pre- and post-disaster images shown partially in the building damage visualizer instance in Figure 6 for better visibility. 2021 Haiti Earthquakes.

IX. Deployment and Applications

Automating building detection and damage assessment has the potential to tremendously speed up disaster response processes [6], [27], which are of critical importance for humanitarian organizations to estimate the geographical extent and the severity of a disaster and plan accordingly [28]. To ensure that such an assessment can be delivered in time, this study's stakeholder is implementing the proposed model within a scalable, distributed computing system. Within this system, satellite images are divided among many identical instances of the model, which process them in parallel. This guarantees a fixed computation time with any number and size of input satellite images. The model's output is then shared with the wider humanitarian network in three ways: the aforementioned web visualizer, the open data-sharing platform "Humanitarian Data Exchange", and via man-made maps (in digital or printed format), which can be directly sent to and used by first responders in the field. This ensures the rapid diffusion of information among all stakeholders involved in disaster response management. It is worth noting that our stakeholder's experience with applying such tools in humanitarian settings and discussions with practitioners have highlighted the importance of two aspects. First, the value of automating damage assessment lies in speed, rather than accuracy. Regardless of visible damages, detailed ground-level inspections by trained personnel are still needed to assess the structural integrity of a building [29] and it is unlikely that remote sensing technology will replace that in the near future. For this reason, the focus of satellite-based damage assessments should be to provide broad numerical estimates as fast as possible, rather than building-level prescriptions. Secondly, while the immediate response is primarily informed by disaster impact (which can be quantified by the number of damaged buildings, among other metrics), long-term shelter recovery programs must take into account several other contextual factors, such as the socio-economic conditions of affected people and land ownership [30]. Because of this, information on building damage often needs to be combined with other data to be useful. Providing raw data including geo-referenced building footprint masks and corresponding damage levels to the humanitarian community is necessary to enable these analyses. Furthermore, as we discussed in section VI, trained models based on the proposed approach might not be robust to significant distribution shift across different geographies; as such, domain adaptation techniques need to be explored to address the data biases issues [31]. In this context, active learning and human-machine collaboration approaches have been discussed in [32] and [33].

X. Conclusion

Natural disasters' frequency is growing; thus, the impact of such events on communities continues to increase. The strategic response of humanitarian organizations to allocate resources and save lives after disasters can be improved by using AI tools. We propose a convolutional neural network model that uses satellite images from before and after natural

[3] https://gdal.org/programs/index.html
[4] https://leafletjs.com/
[5] https://www.docker.com/
[6] https://github.com/microsoft/Nonprofits/

disasters to localize buildings using the UNet model and score their damage level on a scale of 1 (not-damaged) to 4 (destroyed) using a multi-class classifier. We showed that while our proposed model demonstrates decent performance, it also works three times faster than the fastest xView2 challenge winning solution and over 50 times faster than the slowest first place solution, which indicates a significant improvement from an operational perspective. We also developed a web-based visualizer that can display the before and after imagery along with the model's building damage predictions on a custom map to allow better inspection of the impacted areas by decision-makers. This paper outlines results of a collaboration between Microsoft AI for Good/Humanitarian Action and 510 an initiative of the Netherlands Red Cross, to help inform field deployments using satellite imagery and AI technologies. Our solution outperforms stakeholder's current baseline model significantly in terms of inference speed and segmentation accuracy. This study's stakeholder is planning to deploy and assess our proposed solution at the time of actual disasters.

References

[1] OCHA-GHO, "Global humanitarian overview," https://reliefweb.int/sites/reliefweb.int/files/resources/GHO2019.pdf, 2019.

[2] W. M. O. WMO, "Climate and weather related disasters surge five-fold over 50 years, but early warnings save lives - wmo report," https://news.un.org/en/story/2021/09/1098662, 2021.

[3] Defence-Innovation-Unit, "xview2: Assess building damage," https://xview2.org/challenge, 2019.

[4] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 4063–4067.

[5] H. Hao, S. Baireddy, E. R. Bartusiak, L. Konz, K. LaTourette, M. Gribbons, M. Chan, E. J. Delp, and M. L. Comer, "An attention-based system for damage assessment using satellite imagery," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 4396–4399.

[6] T. Valentijn, J. Margutti, M. van den Homberg, and J. Laaksonen, "Multi-hazard and spatial transferability of a cnn for automated building damage assessment," *Remote Sensing*, vol. 12, no. 17, p. 2839, 2020.

[7] A. M. El Amin, Q. Liu, and Y. Wang, "Convolutional neural network features based change detection in satellite images," in *First International Workshop on Pattern Recognition*, vol. 10011. International Society for Optics and Photonics, 2016, p. 100110W.

[8] S. Kaur, S. Gupta, S. Singh, D. Koundal, and A. Zaguia, "Convolutional neural network based hurricane damage detection using satellite images," *Soft Computing*, pp. 1–15, 2022.

[9] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep siamese convolutional network for optical aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1845–1849, 2017.

[10] F. Rahman, B. Vasu, J. Van Cor, J. Kerekes, and A. Savakis, "Siamese network with multi-level features for patch-based change detection in satellite imagery," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 958–962.

[11] H. Chen, C. Wu, B. Du, and L. Zhang, "Change detection in multi-temporal vhr images based on deep siamese multi-scale convolutional networks," *arXiv preprint arXiv:1906.11479*, 2019.

[12] D. Peng, Y. Zhang, and H. Guan, "End-to-end change detection for high resolution satellite images using improved unet++," *Remote Sensing*, vol. 11, no. 11, p. 1382, 2019.

[13] J. Chen, Z. Yuan, J. Peng, L. Chen, H. Huang, J. Zhu, Y. Liu, and H. Li, "Dasnet: Dual attentive fully convolutional siamese networks for change detection in high-resolution satellite images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1194–1206, 2020.

[14] V. Oludare, L. Kezebou, O. Jinadu, K. Panetta, and S. Agaian, "Attention-based two-stream high-resolution networks for building damage assessment from satellite imagery," in *Multimodal Image Exploitation and Learning 2022*, vol. 12100. SPIE, 2022, pp. 224–239.

[15] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[16] R. Gupta and M. Shah, "Rescuenet: Joint building segmentation and damage assessment from satellite imagery," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 4405–4411.

[17] H. Chen, E. Nemni, S. Vallecorsa, X. Li, C. Wu, and L. Bromley, "Dual-tasks siamese transformer framework for building damage assessment," *arXiv preprint arXiv:2201.10953*, 2022.

[18] A. Ismail and M. Awad, "Towards cross-disaster building damage assessment with graph convolutional networks," *arXiv preprint arXiv:2201.10395*, 2022.

[19] ——, "Bldnet: A semi-supervised change detection building damage framework using graph convolutional networks and urban domain knowledge," *arXiv preprint arXiv:2201.10389*, 2022.

[20] Y. Wang, A. W. Z. Chew, and L. Zhang, "Building damage detection from satellite images after natural disasters on extremely imbalanced datasets," *Automation in Construction*, vol. 140, p. 104328, 2022.

[21] Z. Xia, Z. Li, Y. Bai, J. Yu, and B. Adriano, "Self-supervised learning for building damage assessment from large-scale xbd satellite imagery benchmark datasets," *arXiv preprint arXiv:2205.15688*, 2022.

[22] J. Block, D. Crawl, T. Artes, C. Cowart, R. de Callafon, T. DeFanti, J. Graham, L. Smarr, T. Srivas, and I. Altintas, "Firemap: A web tool for dynamic data-driven predictive wildfire modeling powered by the wifire cyberinfrastructure," in *AGU Fall Meeting Abstracts*, vol. 2016, 2016, pp. PA23B–2234.

[23] M. Madaio, S.-T. Chen, O. L. Haimson, W. Zhang, X. Cheng, M. Hinds-Aldrich, D. H. Chau, and B. Dilkina, "Firebird: Predicting fire risk and prioritizing fire inspections in atlanta," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 185–194.

[24] R. Gupta, B. Goodman, N. Patel, R. Hosfelt, S. Sajeev, E. Heim, J. Doshi, K. Lucas, H. Choset, and M. Gaston, "Creating xbd: A dataset for assessing building damage from satellite imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2015, pp. 234–241.

[26] R. Gupta, R. Hosfelt, S. Sajeev, N. Patel, B. Goodman, J. Doshi, E. Heim, H. Choset, and M. Gaston, "xbd: A dataset for assessing building damage from satellite imagery," *arXiv preprint arXiv:1911.09296*, 2019.

[27] A. Elia, S. Balbo, and P. Boccardo, "A quality comparison between professional and crowdsourced data in emergency mapping for potential cooperation of the services," *European Journal of Remote Sensing*, vol. 51, no. 1, pp. 572–586, 2018.

[28] D. P. Coppola, *Introduction to International Disaster Management*. Elsevier, 2006.

[29] GFDDR, "Post-disaster needs assessments guidelines volume b housing," 2017.

[30] H. Shelter, "Settlements guidelines," 2017.

[31] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.

[32] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari, "A survey of active learning algorithms for supervised remote sensing image classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 606–617, 2011.

[33] N. Jojic, N. Malkin, C. Robinson, and A. Ortiz, "From local algorithms to global results: Human-machine collaboration for robust analysis of geographically diverse imagery," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 270–273.