Rapid building damage assessment workflow: An implementation for the 2023 Rolling Fork, Mississippi tornado event

Caleb Robinson[1]*, Simone Fobi Nsutezo[1], Anthony Ortiz[1], Tina Sederholm[1], Rahul Dodhia[1],
Cameron Birge[2], Kasie Richards[3], Kris Pitcher[3], Paulo Duarte[3], Juan M. Lavista Ferres[1]

[1]Microsoft AI for Good Research Lab, [2]Microsoft Philanthropies,
[3]American Red Cross

Abstract

Rapid and accurate building damage assessments from high-resolution satellite imagery following a natural disaster is essential to inform and optimize first responder efforts. However, performing such building damage assessments in an automated manner is non-trivial due to the challenges posed by variations in disaster-specific damage, diversity in satellite imagery, and the dearth of extensive, labeled datasets. To circumvent these issues, this paper introduces a human-in-the-loop workflow for rapidly training building damage assessment models after a natural disaster. This article details a case study using this workflow, executed in partnership with the American Red Cross during a tornado event in Rolling Fork, Mississippi in March, 2023. The output from our human-in-the-loop modeling process achieved a precision of 0.86 and recall of 0.80 for damaged buildings when compared to ground truth data collected post-disaster. This workflow was implemented end-to-end in under 2 hours per satellite imagery scene, highlighting its potential for real-time deployment.

1. Introduction

Rapid building damage assessment in the wake of a natural disaster can inform first responders about the types and magnitudes of resources needed on the ground to support affected populations. Advancements in automated building damage assessment have been supported by the availability of high resolution imagery pre and post a disaster paired with corresponding damage labels in datasets such as xBD [5] and xFBD [8]. Variants of semantic segmentation methods have been applied to pre and post-disaster imagery to perform the dual tasks of outlining building footprints and classify damaged buildings ([6, 7, 1, 4]). Despite these advancements, building damage assessment models

*Corresponding author: caleb.robinson@microsoft.com

suffer from significant degradation when evaluated on out-of-distribution datasets [2, 4] due to the breadth of variations in the type of disaster, image modality, capture conditions, weather, and data size, among others. These issues limit the usability of existing disaster assessment models in real world scenarios where quick and accurate results are needed for a specific area using any available imagery.

We present a rapid response damage assessment workflow that utilizes a small number of labels and only post-disaster imagery. We demonstrate a fast mechanism by which a model can be tuned to perform well under a new disaster using only a small number of training examples thereby eliminating the need for generalization. In addition, this method reduces the dependence on pre-post image pairs, resulting in a more streamlined damage assessment pipeline. This workflow was deployed immediately after the occurrence of a natural disaster and validated in collaboration with the American Red Cross (ARC), showing that this approach can support rapid disaster response.

2. Application Context

On **Friday, March 24, 2023,** an EF4 tornado formed in Mississippi. The tornado lasted for 71 minutes and caused 16 fatalities and 165 injuries. It first touched down in Issaquena County, and tracked northeastward through Sharkey and Humphreys Counties, causing extensive damage in Rolling Fork, Midnight, and Silver City. The most severe damage was observed in Rolling Fork, where numerous structures were destroyed or severely damaged, including homes, businesses, a hospital, schools, and a water tower. Throughout its path, the tornado produced varying degrees of damage, from minor tree and power pole damage to catastrophic destruction of buildings and infrastructure. The tornado eventually dissipated after having traveled 59.4 miles (95.6 km) in total.

On **March 25th**, our partners at the ARC suggested that we use the event to test our damage assessment workflow, to

Figure 1. (**Left**) Post disaster Planet SkySat imagery over Rolling Fork, Mississippi from March 25th; (**Right**) Building level model damage estimates (color gradient indicates damage from no damage colored in white to destroyed in red).

provide them with satellite imagery-based damage analysis to inform ground operations that would occur over the following week. The same day, a Planet Labs SkySat satellite captured a clear post-disaster image of the town of Rolling Fork, which provided us with the necessary data to implement our workflow (described in Section 3).

On **March 26th**, we executed our damage assessment workflow, and provided the ARC with a set of 1347 building footprints (from the Global Microsoft Building ML Footprints dataset [9]), and an estimate of how damaged each building was on a scale of 0 to 100.

Over the following weeks, the ARC deployed more than 800 trained disaster workers to respond to the event. These disaster workers provided sheltering to those displaced by the disaster as well as supported health, mental health services, and recovery support. ARC disaster workers provided shelf-stable meals in the community, where other resources were not available and provided relief items including comfort kits to people in need. The disaster response team trained in damage assessment conducted more than 3500 assessments of residential damage across the state of Mississippi. The ground truth damage assessment included over 909 addresses in Rolling Fork where they categorized the building at each address according to the FEMA building damage scale: no visible damage, affected, minor damage, major damage, or destroyed [3].

In the following sections we outline our damage assessment workflow, describe our modeled output over Rolling Fork (as shown in Figure 1), and perform a post-hoc analysis of the model output using the ground truth data.

3. Damage Assessment Workflow

Given a new disaster area of interest (*AOI*), we outline our building damage assessment workflow:

Satellite image and building footprint acquisition. We acquire *post-disaster* high-resolution satellite imagery *scenes* overlapping the AOI. A *scene* is defined as an image captured from the same pass by a satellite[1]. We obtain corresponding building footprints (polygons) for the same scene from OpenStreetMap [11] or the Microsoft Building Footprint dataset [9].

Satellite image-based label acquisition. We set up an instance of the "satellite imagery labeling tool" [10] with post-disaster imagery. We manually label a random selection of building footprints in the AOI as "damaged" or "not damaged" based on a visual inspection of the post-disaster imagery and use them for validation of the modeling output.

Model training. First, we manually label a total of ≈ 100 example polygons of "background", "building", and "damaged building" classes using the labeling tool instance. Next, we fine-tune a pre-trained semantic segmentation model[2] with the cumulative labeled examples. We run the fine-tuned model on the entire scene to classify each pixel as: "background", "building", or "damaged building". We

---

[1]Large AOIs are often made up of multiple scenes from potentially different satellite passes at different times of day, with different imaging angles, cloud coverage, illumination, and on-the-ground conditions.

[2]We used a U-Net with an ImageNet pre-trained ResNet-50 backbone as our semantic segmentation architecture, but practitioners can choose alternative architectures with pre-trained models better suited to the task.

compute, for each building footprint in the AOI, the percentage area of the footprint that is classified as "damaged building", $\hat{y}_i$. These damage proportions are used to represent the likelihood that a building is damaged. This process is repeated, per scene, until the performance of the modeled output is acceptable (with respect to the validation labels).

Formally, the output of this building damage assessment process is a set of N building footprints (polygons) and the associated damage estimates, $\{\hat{y}_i\}_{i=1}^N$ where $\hat{y}_i$ is the estimated percent damage of the ith building, i.e. $\hat{y}_i \in [0, 100]$.

Validation. Given building level percent damage estimates obtained from a model, we determine the performance of the model by computing a set of validation metrics with respect to labeled building level data. Here, labeled building level data can be obtained either from a ground truth assessment by experts (e.g. a categorization of each building according to the FEMA building damage criteria) or, less desirably, by an independent visual interpretation of the post-disaster imagery. Formally, we assume that we have N labels over C damage classes, $\{y_i\}_{i=1}^N$, where $y_i \in \{1, \cdots, C\}$ and the damage classes represent categories such as "no visible damage", "affected", "minor damage", "major damage", and "destroyed".

First, we aggregate the C damage classes into two classes, i.e. converting the problem into a binary classification problem. For example, if our damage labels follow the FEMA categorization, we may consider "major damage" and "destroyed" as a single "damaged" class with the remaining categories grouped into a "not damaged" class.

Now, given a percent damage estimate, $\hat{y}_i$, and binary damage label, y_i, for each building, we compute precision and recall for the damaged class under an *any damage* threshold (i.e. where $\hat{y}_i > 0$), as well as a precision-recall curve and average precision metric. More specifically, given a threshold value, θ, we compute the following:

tp$_\theta$ *true positives*, the number of buildings in which $\hat{y}_i > \theta$ and $y_i = 1$, i.e. the number of buildings where our model estimated a damage percentage greater than θ and the building was "damaged".

fp$_\theta$ *false positives*, the number of buildings in which $\hat{y}_i > \theta$ and $y_i = 0$, i.e. the number of buildings where our model estimated a damage percentage greater than θ and the building was "not damaged".

fn$_\theta$ *false negatives*, the number of buildings in which $\hat{y}_i \leq \theta$ and $y_i = 1$, i.e. the number of buildings where our model estimated a damage percentage less than or equal to θ and the building was "damaged".

Precision and recall under the any damage threshold ($\theta = 0$)

are then:

$$\textbf{Precision}_0 = \frac{\text{tp}_0}{\text{tp}_0 + \text{fp}_0} \qquad (1)$$

$$\textbf{Recall}_0 = \frac{\text{tp}_0}{\text{tp}_0 + \text{fn}_0} \qquad (2)$$

Intuitively, precision is the fraction of buildings that the model estimates to be damaged, that are, in fact, damaged. E.g. if precision is 0.8, then 1 out of every 5 buildings the model predicts to be damaged, will not be damaged. Similarly, recall is the fraction of damaged buildings that the model correctly estimates to be damaged. E.g. if recall is 0.8 then the model identified 80% of the damaged buildings in the scene.

Setting θ to all possible values between 0 and 1 and plotting the resulting precision and recall values gives us a "precision-recall curve" that shows the trade off between precision and recall. As the percent damage threshold with which we use to classify a building as damaged increases, the precision of the model will increase while the recall decreases. Intuitively, the precision-recall curve serves as proxy for damage localization, where "major damage" and "destroyed" buildings should have large percent damage estimates.

Finally, **average precision** (AP) is a summary of precision and recall across multiple thresholds. Specifically, AP is a weighted mean of precisions over the set of possible θ values, where the weighting is obtained from the change in recall between thresholds [12]:

$$\textbf{AP} = \sum_n \left(\text{Recall}_{\theta_n} - \text{Recall}_{\theta_{n-1}} \right) * \text{Precision}_{\theta_n} \qquad (3)$$
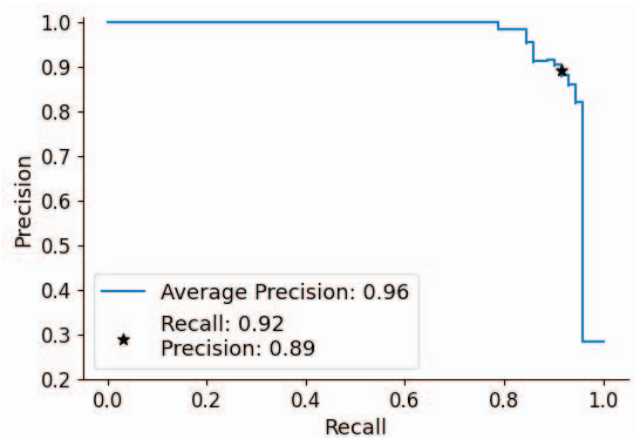

Figure 2. Precision-recall curve for our building damage estimates compared to image interpretation based labels.
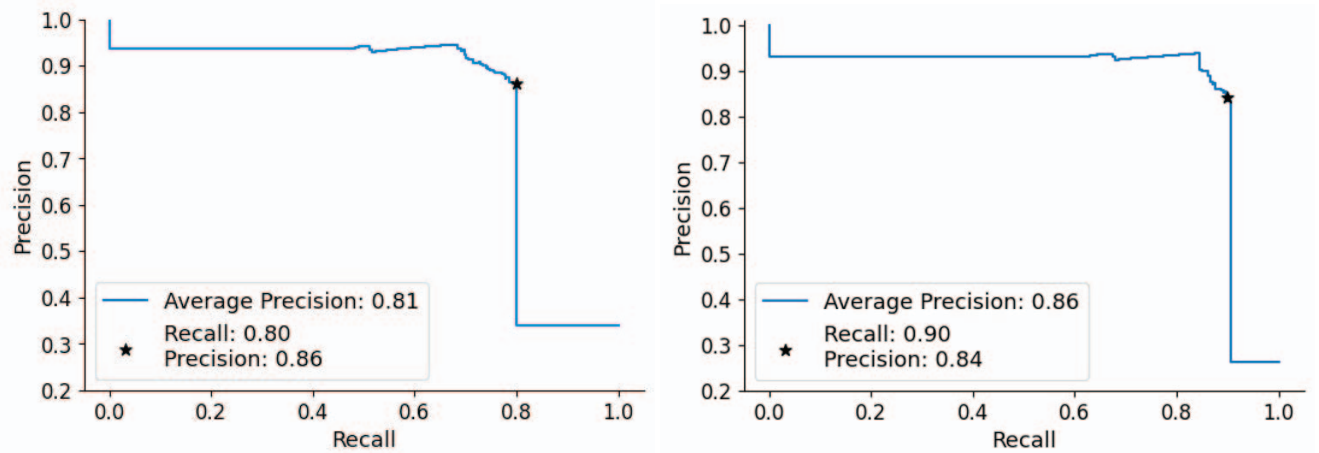
Figure 3. Precision-recall curves for our building damage estimates compared to ground truth labels under two definitions of "damaged" buildings. The (**left**) panel shows the result when "major damage" and "destroyed" points are considered "damaged" while the (**right**) panel shows the result when only the "destroyed" points are considered "damaged".

4. Rolling Fork, Mississippi Results

For our Rolling Fork building damage assessment, we produced percent damage estimates for 1,347 buildings over the city of Rolling Fork, Mississippi using Planet SkySat imagery from March 25th, 2023[3]. We annotated 153 examples of "background" (34 examples), "undamaged building" (71 examples), and "damaged building" (48 examples), then used these to train a semantic segmentation model that can make predictions over the entire image.

We perform two validation efforts – using labeled data collected through a visual interpretation of the satellite imagery and using labeled data collected through a ground truth analysis. We detail the results of these efforts below.

4.1. Validation with image-based labels

Our initial validation effort involves annotating a set of 250 randomly selected building footprints as "damaged" or "not damaged" based on an interpretation of the post-disaster imagery that was independent of the annotation effort used to train the damage assessment model. Our building level damage estimates have an average precision of 0.96 with a precision-recall curve shown in Figure 2. The precision and recall under the "any damage" threshold are 0.96 and 0.74.

4.2. Validation with ground truth based labels

We received 909 labels from the ARC based on a ground truth analysis that they performed during their disaster response. These labels were assigned to *addresses* and categorized according to the FEMA building damage assessment scale. We first geocoded the addresses to obtain latitude, longitude point locations, and matched them to the set

of building footprints that we performed the damage assessment over. Due to noise in the geocoding process and the fact that not all buildings were analyzed by the ARC, sometimes the ground truth points are not contained in a building footprint[4]. Thus, we associate each ground truth point with the nearest building footprint in a 20 meter radius (if any).

We compute the validation metrics over the 830 ground truth points that are associated with a building footprint.

If we consider ground truth points that are categorized as "major damage" or "destroyed" to be the "damaged" class, and the remaining categories to be an "undamaged" class, precision and recall under the "any damage" threshold are 0.862 and 0.801. When only the "destroyed" class is considered to be "damaged", then precision and recall are 0.84 and 0.90. Figure 3 shows the localization performance of the model where average precision increases from 0.81 ("major" and "destroyed") to 0.86 (only "destroyed").

5. Conclusion

Our work demonstrates that a human-in-the-loop workflow can provide a rapid and effective method of building damage assessment post-natural disasters. Our method has demonstrated its practical usefulness by quickly training with minimal data, effectively bypassing the requirement for extensive labeled datasets and the challenges of training models that account for diverse disaster scenarios and satellite imagery variations. A key advantage of the workflow is its capability to be implemented in under 2 hours per satellite imagery scene by any user familiar with annotating satellite imagery, which enables its potential use in

---

[3]Scene ID: 20230325_195842_ssc11_u0001.

[4]Examples are i) the point occurs along a road, ii) several points occur on a single building or iii) a building has no matching points.

production scenarios where immediate action and response are of paramount importance. This was evident in the case of the tornado event in Rolling Fork, Mississippi, where the model, having been trained and validated collaboratively with the ARC, resulted in a precision of 0.86 and a recall of 0.80 when evaluated against ground truth data. Future work will focus on refining this workflow, exploring scalability across disaster types, and strengthening collaboration between AI and human expertise.

Acknowledgments

References

[1] Yanbing Bai, Junjie Hu, Jinhua Su, Xing Liu, Haoyu Liu, Xianwen He, Shengwang Meng, Erick Mas, and Shunichi Koshimura. Pyramid pooling module-based semi-siamese network: A benchmark model for assessing building damage from xBD satellite imagery datasets. *Remote Sensing*, 12(24):4055, 2020.

[2] Vitus Benson and Alexander Ecker. Assessing out-of-domain generalization for robust building damage detection. *arXiv preprint arXiv:2011.10328*, 2020.

[3] FEMA. FEMA Preliminary Damage Assessment Guide, August 2021. `https://www.fema.gov/sites/default/files/documents/fema_2021-pda-guide.pdf`, 2021.

[4] Shahrzad Gholami, Caleb Robinson, Anthony Ortiz, Siyu Yang, Jacopo Margutti, Cameron Birge, Rahul Dodhia, and Juan Lavista Ferres. On the deployment of post-disaster building damage assessment tools using satellite imagery: A deep learning approach. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1029–1036. IEEE, 2022.

[5] Ritwik Gupta, Richard Hosfelt, Sandra Sajeev, Nirav Patel, Bryce Goodman, Jigar Doshi, Eric Heim, Howie Choset, and Matthew Gaston. xbd: A dataset for assessing building damage from satellite imagery. *arXiv preprint arXiv:1911.09296*, 2019.

[6] Rohit Gupta and Mubarak Shah. Rescuenet: Joint building segmentation and damage assessment from satellite imagery. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4405–4411. IEEE, 2021.

[7] Eugene Khvedchenya and Tatiana Gabruseva. Fully convolutional siamese neural networks for buildings damage assessment from satellite images. *arXiv preprint arXiv:2111.00508*, 2021.

[8] Dennis Melamed, Cameron Johnson, Chen Zhao, Russell Blue, Philip Morrone, Anthony Hoogs, and Brian Clipp. xFBD: Focused building damage dataset and analysis. *arXiv preprint arXiv:2212.13876*, 2022.

[9] Microsoft. Global ML Building Footprints. `https://github.com/microsoft/GlobalMLBuildingFootprints`, 2023.

[10] Microsoft. Satellite Imagery Labeling Tool. `https://github.com/microsoft/satellite-imagery-labeling-tool`, 2023.

[11] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.