

---

# Damage and Accessibility Assessment for Post-Disaster Regions from Satellite Imagery

---

**Coco (Xiaoyan) Li**

Department of Computer Science  
Department of Cognitive Science  
Johns Hopkins University  
Baltimore, MD 21218  
xli152@jhu.edu

**Darius Irani**

Department of Computer Science  
Department of Applied Mathematics and Statistics  
Johns Hopkins University  
Baltimore, MD 21218  
dirani2@jhu.edu

**Jung Min Lee**

Department of Biomedical Engineering  
Department of Computer Science  
Johns Hopkins University  
Baltimore, MD 21218  
jlee581@jhu.edu

**Tara Tang**

Department of Biomedical Engineering  
Department of Computer Science  
Johns Hopkins University  
Baltimore, MD 21218  
ttang14@jhu.edu

## Abstract

In recent years, an increase in the frequency and severity of natural disasters led to deeper consideration of the role computer vision can play in automated interpretation of post-disaster satellite images. When providing assistance to victims of a natural disaster, first responders benefit from any and all information on damaged areas. While previous work on disaster monitoring has focused primarily on the amount of damage in a region, it cannot easily be interpreted by non-professionals. Thus, we attempt to not only calculate damage to man-made features, but also provide information on the accessibility of certain regions. We used labeled satellite imagery datasets and was able to train U-Net and FPN neural networks to segment roads with an IoU of 0.54 and 0.55 respectively. Using satellite imagery and crowd-sourced damage labels before and after Hurricane Irma, we used our trained networks to extract roads from the pre- and post-event images, then calculated damage levels as well as accessibility indexes for each image region.

**Keywords:** image segmentation, damage assessment, satellite imagery, deep learning, emergency response, emergency mapping, accessibility mapping

# 1 Introduction

Following natural disasters such as hurricanes and tsunamis, first responders have limited time to provide aid and evacuate residents. Central to their strategic deployment of forces is knowledge of (1) the extent of damage in regions and (2) the accessibility of different routes. The challenge is to quickly annotate provided satellite image maps as a standard resolution. In computer vision, semantic segmentation techniques have shown promising results in identifying artificial architectures such as roads and buildings from satellite imagery. This information can then be used to detect regions where these features have been heavily damaged. While a lot of research is ongoing in this field, most case studies have focused on speed and accuracy of damage mapping in order to mark destruction as soon as possible after a disaster occurs [1, 2]. In addition, the focus was mainly on marking areas of damage instead of providing more detailed information that could be immediately usable by first responders.

By contrast, the end goal of our project is to score and mark the accessibility of the affected regions using damage indices on man-made architectures (roads, buildings, etc.). Accessibility scores offer more information on how first responders should allocate their resources when stretched across a large damaged area post-disaster. Accessibility and damage index together can also prove useful in prioritizing certain regions in rescue and recovery efforts.

## 2 Proposed Approach

### 2.1 Overview

We propose to identify regions with reduced accessibility by determining areas with heavy road damage. Using a pre-trained semantic segmentation model, we extract roads from each satellite image, both before and after the disaster event. We then identify the level of damage in the roads by computing the difference between the two images. The accessibility index is computed by calculating the percentage change in the pixels identified as roads.

### 2.2 Model Architecture

For semantic segmentation, we are training two different models: U-Net and Feature Pyramid Network (FPN). U-Net is a convolutional neural network developed for biomedical image segmentation [9]. The network has multiple initial contraction blocks, where each block consists of two  $3 \times 3$  convolution layers followed by a  $2 \times 2$  max pooling layer. This is followed by several expansion blocks, where each block is again made up of two  $3 \times 3$  convolution layers followed by a  $2 \times 2$  upsampling layer. During upsampling, the feature map from the corresponding contraction layer is also appended to the input (see Figure 1).

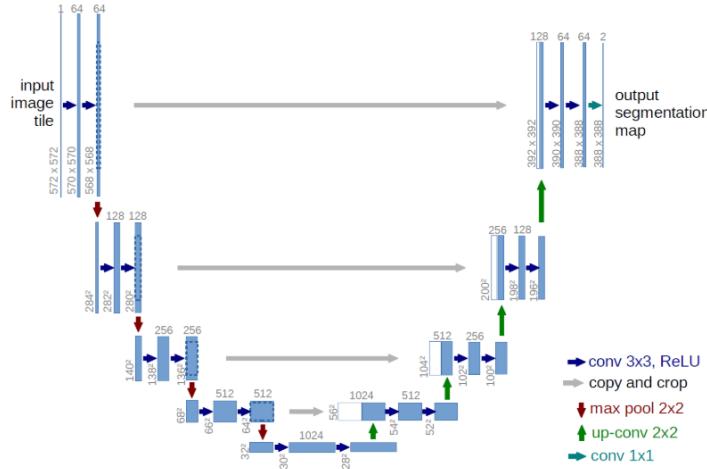


Figure 1: U-Net Architecture. Figure adapted from [6]

FPN is usually used for objection detection and was designed for improved speed. It's composed of a bottom-up pathway (using ResNet) with multiple convolution layers and a top-down pathway with upsampling layers that also has lateral connections (see Figure 2). While FPN is not the traditional choice for semantic segmentation, several papers have suggested that FPN can also be used as an effective, and yet lightweight model for both instance and semantic segmentation [6, 7]. Thus, we decided to implement this network for our project and compare the results with U-Net.

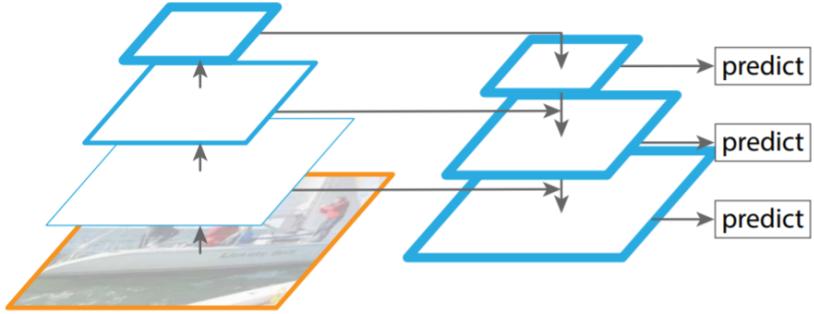


Figure 2: FPN Architecture. Figure adapted from [7]

### 2.3 Proposed Damage Levels and Accessibility Index

Previous attempts to calculate damage levels using satellite imagery have used pixelwise change in the pre-/post-event masks as an approximate indicator for damage levels [4]. Different metrics have been used in the past to quantify the level of damage. However, we were unable to find metrics for accessibility. Therefore, we decided to define our own accessibility index as the fraction of the road that remained intact post event.

Road-pixels are the pixels in the output mask that are marked as 1, indicating the presence of a road. The difference is divided by total number of road-pixels in the pre-disaster to mitigate for how dense the road was and thus how damaged it was.

$$\text{Accessibility Index} = 1 - \frac{\text{Number of road-pixels in (Pre-Event Mask)} - (\text{Post-Event Mask})}{\text{Number of road-pixels in Pre-Event Mask}}$$

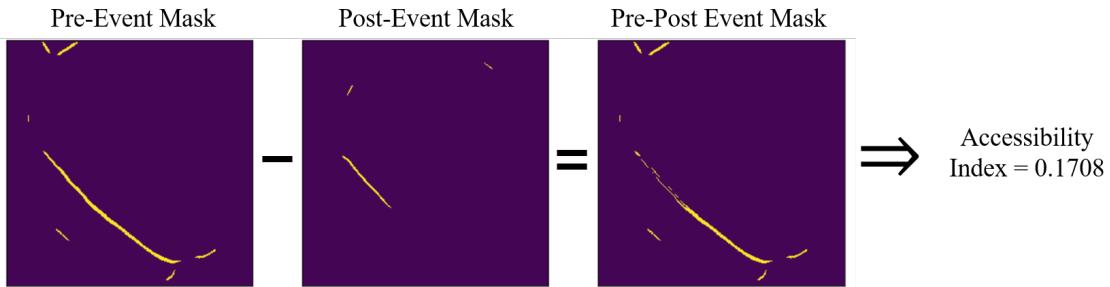


Figure 3: Calculation of Accessibility Index

## 3 Datasets and Processing

### 3.1 Datasets

Two different datasets were utilized for this project. For training neural networks, we used a labeled road satellite image training dataset from [3]. For testing our neural network performance, we used real pre-/post-hurricane satellite images from DigitalGlobe.

#### 3.1.1 DeepGlobe Road Extraction Challenge Training Images

For training our segmentation model, a high-resolution satellite image set from the 2018 DeepGlobe Road Extraction Challenge was used. The training set provided by the challenge contains 6226 satellite images with ground truth labeled masks for roads (see Figure 4). The images were collected from various locations around the world, including Las Vegas, Paris, Shanghai and Khartoum, in order to cover a variety of road sparsity. The image dimensions are  $1024 \times 1024$  pixels.



Figure 4: Example Training Image (Left) and Ground-Truth Mask (Right) Pre-Augmentation

### 3.1.2 DigitalGlobe Pre-/Post-Hurricane Satellite Images

DigitalGlobe provides public access to its extensive database of satellite disaster images to assist with relief and response efforts to these events<sup>1</sup>. From DigitalGlobe, we obtained satellite images from before and after the hurricane (Figure 5), as well as a crowd-sourced, hand-labeled GeoJSON file of damaged regions as ground truth, courtesy of TOMNOD (Figure 6). The label classes included: Blocked Bridge, Flooded/Blocked Road, Flooded/Damaged Building, and Trash Heap. For the scope of this project, we made use of the Blocked Bridge and Flooded/Blocked Road damage classes.



Figure 5: Example Pre-Event Image (Left) and Post-Event Image (Right) for the island of Charlotte Amalie. As pictured here, one challenge of working with satellite images is obstruction from cloud coverage.

The crowd-sourced, hand-labeled ground-truth were incomplete, so we selected pre-/post-event satellite images that overlapped with the regions labeled. We identified the island of Charlotte Amalie, the capital of the United States Virgin Islands as meeting these conditions. Charlotte Amalie was damaged post Hurricane Irma, so we used pre-/post-event images from this hurricane (see Figure 6).

## 3.2 Data Processing

Even at standard resolution, satellite images have very large file sizes. To prepare this data for our neural networks, extensive data processing was applied. For reproducibility, we packaged these steps into a comprehensive data processing pipeline, which includes: data compression, image tiling, road density filtering, and augmentations.

### 3.2.1 Data Extraction and Compression

Python’s BeautifulSoup package was used to extract links to hosted images on DigitalGlobe. Since each TIF file was 1.1GB in size, downloading all 1355 pre-/post-disaster images locally was infeasible. We wrote a pipeline that downloads a single TIF file, applies JPEG compression, and then deletes the uncompressed TIF file. The file compressed images ranged from 10-150 MB. JPEG compression is a form of lossy compression that is commonly used in digital images, that runs the risk of introducing artifacts. Other compression modes supported by this pipeline are PackBits (fast, lossless

---

<sup>1</sup>For more information about DigitalGlobe’s Open Data Program, see their open data protocol

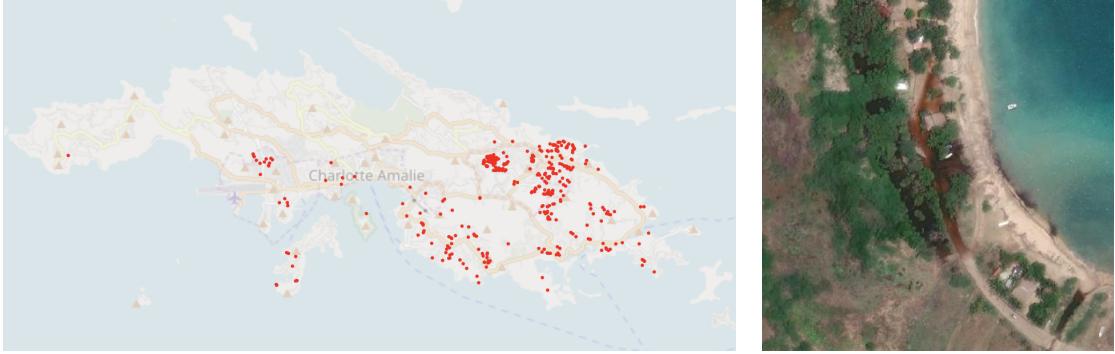


Figure 6: Left: Plot of the centroids of each image tile that were hand-labeled by TOMNOD volunteers, overlayed on a base map of Charlotte Amalie. Right: Example image tile that was labeled Flooded/Blocked Road.

compression for run-length encoding of data), Deflate (lossless compression that uses LZSS and Huffman coding), and LZW (compression using table-based lookup algorithm). We selected JPEG compression over the other methods because of familiarity with this file format, but an area of future work would be to explore the effects of these other compression modes on performance.

### 3.2.2 Image Tiling

The compressed images preserved the resolution of the original satellite TIFs, and were too large to work with. In order to convert the large images to a resolution that is consistent and easier to process, images were spliced into  $512 \times 512$  pixel tiles. The rightmost and bottom tiles were discarded if they were not an even  $512 \times 512$  pixels to preserve consistency.

### 3.2.3 Filtering by Class Balance

To improve training results, our training data was put through an automated filter based on road density in the ground-truth masks. This filter ensured that sparse images with a road density below a certain threshold were discarded. A typical threshold for such a filter is above 15% for larger man-made architectures such as buildings [5]. For our pipeline, since we are training on architectures that are sparser than buildings, the hyper-parameter that we used has threshold of 1%, such that images consisting of less than 1% ground-truth road pixels were not used.

### 3.2.4 Data Augmentation

Image augmentation is a standard method of varying the training data in order to produce a more robust model [10]. Python’s `imgaug` library was imported, and two separate augmenters were used on roughly 10% of the images. The first augmenter consisted of non-noise transformations, and applied a random selection of 0 to 3 of the following operations: scaling hue and saturation, horizontal flip, vertical flip, and affine transformation with symmetric padding. This was applied to both the image and its mask, to maintain label integrity. The second augmenter consisted of noise transformations, and applied one of the following operations: salt and pepper noise, Gaussian blur, and sharpen. This was only applied to the image and the mask was unchanged (see Figures 7 and 8).



Figure 7: Example Training Image (Left) and Ground-Truth Mask (Right) Pre-Augmentation



Figure 8: Example Training Image (Left) and Ground-Truth Mask (Right) Post-Augmentation. In this case, the image and mask received both a horizontal and vertical flip, and the image’s hue and saturation were altered.

## 4 Experiments and Results

### 4.1 Image Matching

In order to produce pre-/post-event images that can be accurately compared pixelwise, we performed feature matching on the image tiles. We tested the OpenCV library’s built-in SIFT (scale-invariant feature transform) and SURF (speeded-up robust features) algorithms to identify features in the pre- and post-event images. It quickly became clear that SURF was both more efficient and accurate, producing a higher number of acceptable homography results than SIFT. Thus, SURF was used as our final feature detector moving forward.

The OpenCV brute-force descriptor matcher (`BfMatcher`) was used to identify the best two matches in the post-event image for each feature in the pre-event image. Feature matches were classified as non-trivial if their two best matches passed Lowe’s Ratio Test [8], where the distance for the first match must be closer than 70% of the distance for the second match. If an image pair yielded at least four non-trivial matches, the homography between the two images could be calculated on the features using Random Sample Consensus (RANSAC). The resulting perspective transformation was then applied to produce a pixelwise matched post-event image (see Figure 9).

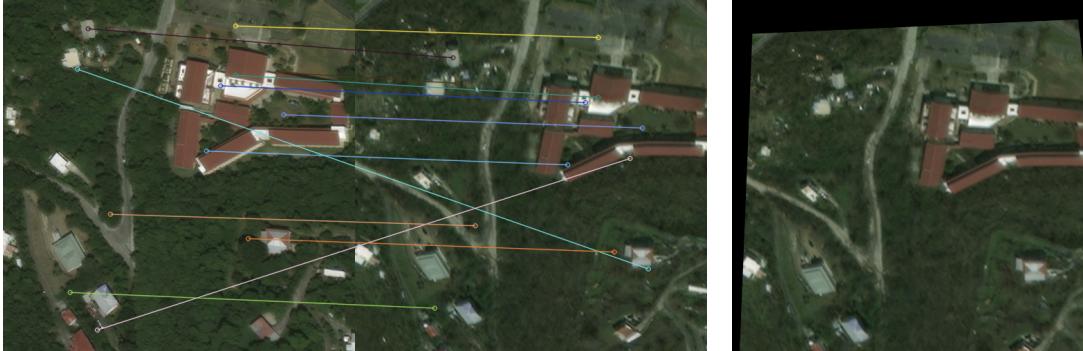


Figure 9: Example Feature Matches (Left) and Perspective Transformation (Right)

$$\text{Example Homography Matrix } H = \begin{bmatrix} 8.55353494e - 01 & -4.14295732e - 02 & 3.22717815e + 01 \\ -4.46778864e - 02 & 7.99831349e - 01 & 5.34838104e + 01 \\ -1.21451547e - 04 & -2.76006367e - 04 & 1 \end{bmatrix}$$

### 4.2 Semantic Segmentation

Two U-Nets and one FPN were trained and tested for semantic segmentation. All the models were trained using 3448 training, 1246 validation, and 1245 testing images from the Deep Globe Road Extraction dataset. The images were resized to  $512 \times 512$  pixels and grayscaled.

#### 4.2.1 U-Net Trained from Scratch

Our model follows the exact model architecture described in [9], with an input size of  $512 \times 512$  pixels. The model<sup>2</sup> was trained for 40 epochs using dice loss as its loss function. However, this model did not train well and had a dice loss of 0.95 even after training. We decided to discard this model for the next steps (see Figure 10).

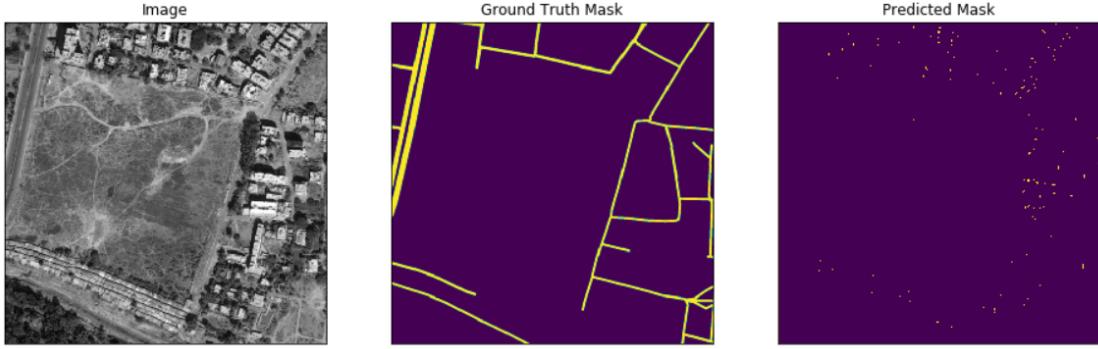


Figure 10: Example Training Result of U-Net

#### 4.2.2 Pretrained FPN

To avoid training from scratch, we used a FPN<sup>3</sup> with a ResNet-34 backbone pretrained on ImageNet. Binary cross entropy was used as the loss function and Intersection over Union (IoU) was used as the performance evaluation metric. The final model had an IoU of 0.55 (see Figure 11).

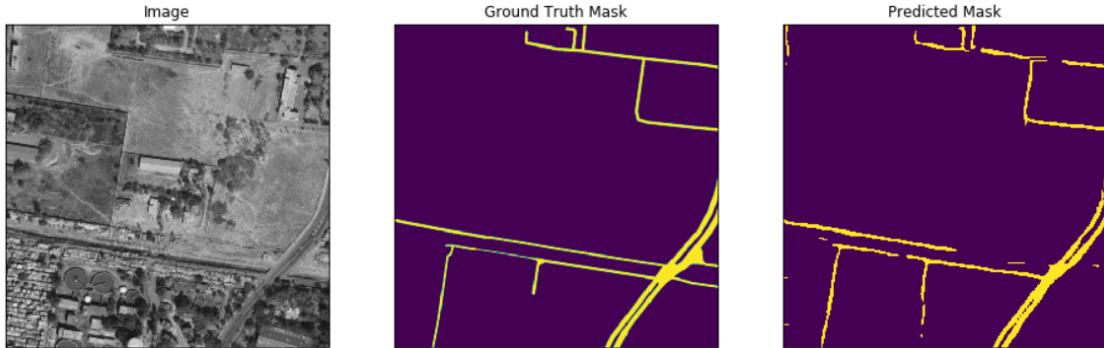


Figure 11: Example Training Result of FPN with ResNet-34 Backbone

#### 4.2.3 Pretrained U-Net

A U-Net<sup>3</sup> with a ResNet-34 backbone pretrained on ImageNet was also trained using binary cross entropy as the loss function and IoU as the performance evaluation metric. The final model had an IoU of 0.54 (see Figure 12).

### 4.3 Road Extraction and Disaster Region Identification

While our initial goal was to use our model to identify regions of damage, due to the incomplete nature of the TOMNOD ground-truth labels, we decided to only test our model for regions that have been clearly labeled as damaged. We used both pre-trained U-Net and the pre-trained FPN on the pre-/post-event satellite images to visually verify whether the trained models could successfully extract the roads from the satellite imagery.

<sup>2</sup>Adapted from <https://github.com/usuyama/pytorch-unet>

<sup>3</sup>Adapted from [https://github.com/qubvel/segmentation\\_models.pytorch](https://github.com/qubvel/segmentation_models.pytorch)



Figure 12: Example Training Result of U-Net with ResNet-34 Backbone

#### 4.3.1 Pre-trained FPN

The FPN was good at detecting roads with obvious contrasts but did not fare as well on blurry images that we often had for post disaster events. Examples of this malfunction can be seen in that sometimes the post event image had more identified roads on the masks than the pre-image due to contrast and sharpness of the image. Although we pre-processed the pre-/post-disaster images with increased contrast, it did not improve the situation (see Figure 13).

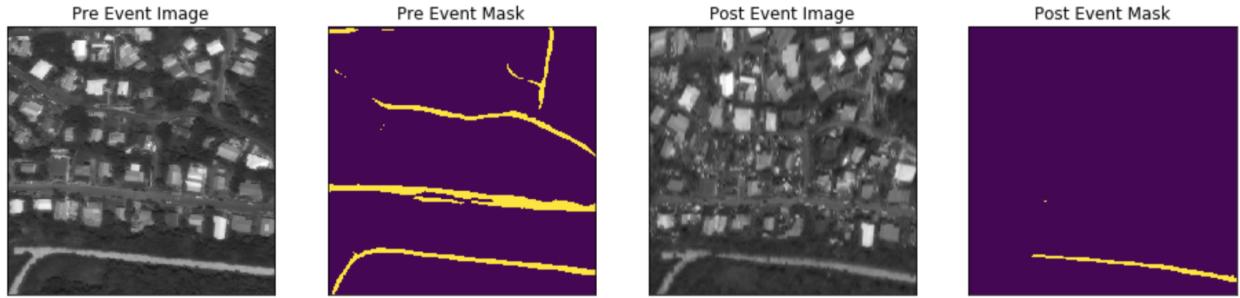


Figure 13: Example Pre/Post-Event Road Extraction Result using FPN

#### 4.3.2 Pre-trained U-Net

Overall the pre-trained U-Net produced better results on the post-disaster image as compared to the pre-trained CN, although both had similar IoU after being trained for the same number of epochs. This was mostly shown in definitely capturing more roads in the pre-disaster image as opposed to the post-disaster images without even having to apply the transformations such as contrast mentioned in the FPN section (see Figure 14).

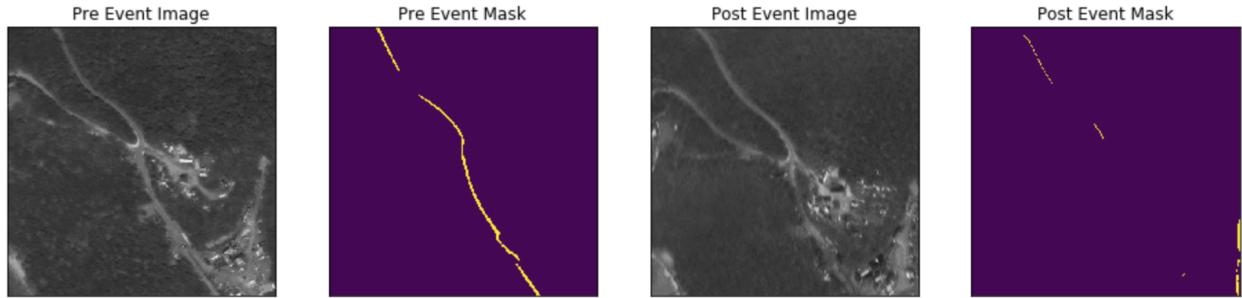


Figure 14: Example Training Result of U-Net with ResNet-34 Backbone

#### 4.4 Damage Levels and Accessibility Index

Damage level and accessibility index computed by both networks are shown on a select subset of the pre-/post-disaster images (see Figures 15 and 16). For the method of calculation, please refer to Section 2.3.

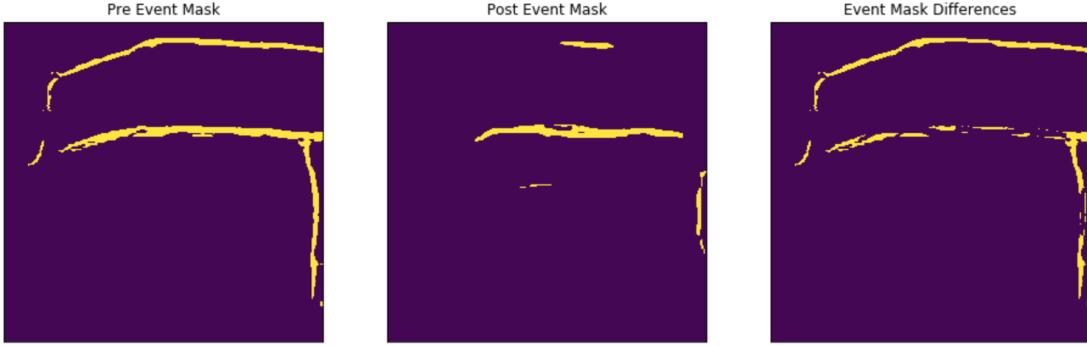


Figure 15: Accessibility index computed From FPN masks. Accessibility Index = 0.3188.



Figure 16: Accessibility index computed From U-Net masks. Accessibility Index = 0.1709.

## 5 Conclusion and Future Work

In this project, we showed that image segmentation techniques can be used to detect levels of damage following natural disasters, which can be used to calculate region accessibility. Using U-Net and FPN we were able to successfully train semantic segmentation models. However, the satellite images we used were of lower resolution than our training images, which frequently resulted in our model not being able to identify all the roads.

As part of this work, we focused on roads due to time constraints and data access, but these results can easily be extended to other man-made and natural structures by adding classes to the networks to provide a more holistic quantification of disaster impact.

### 5.1 Future Work

Obstruction from cloud coverage was a common issue encountered when processing pre-/post-hurricane satellite images. However, cloud obstruction was not consistent across images. For some, the clouds created a complete blackout of the underlying geography, while in other images, the clouds were wispy. A future direction of this work would include automating cloud removal for images. Images with high cloud concentration would show a right shift in an intensity histogram of the grayscale images. Using this observation, Nonnegative Matrix Factorization could be used to perform dimensionality reduction on the data to determine which images are cloudy and exclude images above a certain cloudiness threshold from our dataset [12].

We also noticed that in certain cases, it was hard to observe significant changes between the pre-/post-hurricane images, even with visual inspection. This could be due to the nature of hurricanes, in which areas can have damaged buildings and debris, and yet still have intact roads. In order to use roads as the feature for determining the level of damage as we did for this project, a more appropriate natural disaster would be floods or earthquakes.

To provide a more complete accessibility index, the damage level for buildings and other features should be incorporated. A more complete accessibility index would be defined using a weighted sum of building and road damage within a given image tile. While we intended to train a building segmentation model, we were not able to obtain access to the necessary dataset. Knowing the distribution and level of buildings could also inform first responders on which areas have higher priority for evacuation resources.

## 5.2 Project Reflections

We gained a better understanding of PyTorch and the OpenCV library, also on setting up scripts to train and run deep learning models. We received practical experiences working with pre-trained models and running deep learning VMs using Google Cloud. A majority of our project time was spent on obtaining and pre-processing our dataset. This opened our eyes to the importance of a high-quality dataset in computer vision projects.

### For Future Computer Vision Students and Instructors

We were not aware of how ambitious our project was and the technical difficulties involved with implementing and training a deep neural network. If you don't have prior experience with deep learning models and/or have never worked with Google Cloud or other cloud services, make sure to allocate extra time to familiarize yourself with these tools. Also, make sure to do a thorough search of existing resources and utilize them as much as possible.

Additionally, our project required a significant amount of data processing. We retrieved full resolution satellite images from DigitalGlobe and had to compress, tile, and match the images ourselves before running the images through a pre-processing pipeline for our neural networks. Fortunately, we started working on our project far in advance of the deadline, and accomplishing these tasks were feasible. Computer vision tasks, especially if you're using deep learning methods, are very time-intensive to train since the input data (images) are so large, so make sure to allow sufficient time for yourselves.

More feedback on proposals and the progress report from the instructors would have been helpful, especially in ensuring that the scope of our project was not too ambitious. Additional instructions on PyTorch and setting up deep learning models from scratch would also benefit future students.

## Acknowledgements

We are very grateful to Dr. Gregory Hager, Dr. Wei Shen, and the TAs Jin Bai and Huiyu Wang for their help in this project. We would also like to thank DigitalGlobe and DeepGlobe for making their satellite imagery available, Pavel Yakubovskiy for his `segmentation_models.pytorch` library, and the Disaster Damage Detection team from the 2018 Data Science for Social Good fellowship at the University of Washington eScience Institute for project inspiration.

## References

- [1] Y. Bai, E. Mas, and S. Koshimura, "Towards Operational Satellite-Based Damage-Mapping Using U-Net Convolutional Network: A Case Study of 2011 Tohoku Earthquake-Tsunami," *Remote Sensing*, vol. 10, no. 10, p. 1626, Oct. 2018.
- [2] Q. D. Cao and Y. Choe, "Building Damage Annotation on Post-Hurricane Satellite Imagery Based on Convolutional Neural Networks," *arXiv:1807.01688 [cs]*, May 2019.
- [3] I. Demir et al., "DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images," *arXiv:1805.06561 [cs]*, May 2018.
- [4] J. Doshi, S. Basu, and G. Pang, "From Satellite Imagery to Disaster Insights," p. 5.
- [5] D. Duarte, F. Nex, N. Kerle, and G. Vosselman, "Satellite Image Classification Of Building Damages Using Airborne And Satellite Image Samples In A Deep Learning Approach," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2, pp. 89–96, May 2018.
- [6] A. Kirillov, R. Girshick, K. He, and P. Dollar, "Panoptic Feature Pyramid Networks," [https://arxiv.org/abs/1901.02446 \[cs\]](https://arxiv.org/abs/1901.02446), Jan. 2019
- [7] T. Lin et al., "Feature Pyramid Networks for Object Detection," [https://arxiv.org/abs/1612.03144 \[cs\]](https://arxiv.org/abs/1612.03144), Dec. 2016

- [8] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," arXiv:1505.04597 [cs], May 2015.
- [10] Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data* 6, 60 (2019) doi:10.1186/s40537-019-0197-0
- [11] A. Trekin, G. Novikov, G. Potapov, V. Ignatiev, and E. Burnaev, "Satellite imagery analysis for operational damage assessment in Emergency situations," arXiv:1803.00397 [cs], Feb. 2018.
- [12] M. Xu, X. Jia, M. Pickering, and A. J. Plaza, "Cloud Removal Based on Sparse Representation via Multitemporal Dictionary Learning," *IEEE Trans. Geosci. Remote Sensing*, vol. 54, no. 5, pp. 2998–3006, May 2016.