Clustering Rooftops by Construction Material for Natural Disaster Risk Assessment

Tammy Glazer
MSCAPP, The University of Chicago

James Jensen
MSCAPP, The University of Chicago

Github Repository
https://github.com/jamjensen/mapping-disaster-risk

Clustering Rooftops by Construction Material for Natural Disaster Risk Assessment

Tammy Glazer & James Jensen
Github Repository: https://github.com/jamjensen/mapping-disaster-risk

I. Introduction

The goal of our project is to leverage high definition, aerial imagery to identify and label rooftop construction materials in St. Lucia, Guatemala, and Colombia, using color quantization and unsupervised learning techniques. A successful labeling strategy for aerial imagery can facilitate the prioritization of building inspections in areas that are at a heightened risk of natural disasters and can, in turn, ensure resilience.

Supervised learning requires a large number of rows with correct output labels. When working on image classification, it is incredibly time consuming and costly to create labels, which creates an opportunity for an effective unsupervised approach. While supervised deep learning techniques have historically been applied to pre-processed, labeled training image files to predict specific features contained in these images, we are interested in how unsupervised techniques can be used to derive labels when training data is unavailable. Because, in this case, we do have access to image files with associated roof material labels, we hope to compare our unsupervised output with feature labels to assess the effectiveness of our methods.

II. Literature Review

Since the mid-1990s, floods, storms, and earthquakes have made up nearly 80% natural disasters around the world (Myers, 2016). Increasing global surface temperatures are leading to rising sea levels and the evaporation of excess water vapor into the atmosphere, increasing the likelihood of weather-driven disasters (USGS, 2019). As a direct result, locations including the Caribbean, Central America, the South Pacific, and the Himalayas face an elevated risk of experiencing natural hazards (DrivenData, 2019). While a variety of government and development organizations support emergency management and the provision of relief services, far less is being done by way of risk reduction and the development of data-driven resource prioritization strategies.

In the face of a natural hazard, roof design, condition, and material contribute to household resilience. The traditional approach to identifying high-risk buildings and roofs in disaster prone locations involves going door-to-door to visually inspect building conditions, which can be extremely costly and time consuming. For instance, windstorm mitigation inspections are used to assess the appropriateness of a structure's construction in the event of strong winds. An inspector will look for construction features that have been shown to reduce losses in hurricanes, such as concrete block construction, the presence of roof-to-wall attachments, and opening protections. An alternative approach would be to use high-resolution drone imagery to quickly identify, and in turn, prioritize areas with problematic building materials. According to DrivenData, "Mapping

a 10km2 neighborhood with a drone can be done within a matter of days and at a cost of a few thousand dollars at most" (2019).

The World Bank Global Program for Resilient Housing has assembled drone imagery datasets with the specific goal of accurately mapping disaster risk. It is our hope that a proactive assessment of these files could lead to faster, cheaper, and targeted building inspections as a disaster risk reduction strategy.

III. Empirical Strategy

A. Dataset

WeRobotics and the World Bank Global Program for Resilient Housing teamed up to prepare aerial drone imagery of buildings across the Caribbean annotated with characteristics that matter to building inspectors. The data consist of a set of overhead imagery of several locations across three countries (Colombia, Guatemala, and St. Lucia) at 4cm resolution. The features are the images and building footprints, stored as GeoJSONS. Each of the seven images consist of a very large, high-resolution Cloud Optimized GeoTIFF (COG).

Below is an example of a single GeoTIFF image:

Figure 1. High-resolution Cloud Optimized GeoTIFF

A COG is a regular GeoTiff file that one can host and access on an HTTP file server. Hosted on the cloud, COGs enable more efficient workflows by leveraging the ability to issue HTTP Get range requests, accessing only the necessary parts of the file (COGEO, 2019). A COG adheres to the GeoTiff standard and therefore contains all the benefits one would expect from using a GeoTiff file, such as, the ability to embed georeferencing information within raster data.

While we leverage unsupervised machine learning techniques to group buildings with similar roof types, we do have access to training data containing the unique building ID, building footprint, and roof material of each structure. This enables us to conclude our analysis comparing the results of our models with the actual roof labels for each building, to better capture model accuracy and performance.

B. Data Access

The data are downloadable from the DrivenData Open AI Caribbean Challenge website, in the form of a 34G .tar file. We host this file both locally as well as on a HTTP file server with the University of Chiago's Research Computing Center (RCC) for efficient file interaction.

The .tar file contains an individual GeoJSON file for each building footprint. In this case, a GeoJSON is a type of dictionary that stores the building ID, roof material (for testing purposes), and coordinates outlining the building. Additionally, the file contains a single GeoTIFF raster image for each geographic location. A raster image is a bitmap, or a grid of individual pixels, that collectively compose an image. In order to begin working with these files, we wrote a Python script that reads a GeoJSON and GeoTIFF file, extracts the relevant information for a single selected rooftop, and displays the single rooftop as an image. This script contains functions to load a GeoJSON in a usable format, extract an abbreviated dictionary of relevant information about each building, and execute an affine transformation on coordinates provided for each building based on the coordinate reference system (CRS) from the corresponding GeoTIFF file. This is a necessary step because coordinates map differently on different locations of the world based on the earth's natural curvature.

We also write scripts to extract pixel-value arrays for each band of an image given a dictionary containing transformed coordinates for each roof. Each band contains information on surface reflectance from different ranges of the electromagnetic spectrum. In this case, each file contains four bands. The first represents the red band, the second represents the green band, the third represents the blue band, and the fourth band represents alpha, which denotes the level of transparency for each pixel across the three bands. Within each band matrix, each individual value represents a pixel and is assigned a number from 0-255, with larger numbers representing more of that color. Finally, we write scripts to create new, small .tif files for each individual rooftop, as well as to display each raster band with a corresponding colorbar.

Below are examples of two randomly selected rooftops and three layers of a single rooftop:

Figure 2. Two Rooftops Extracted from the Colombia GeoTIFF File

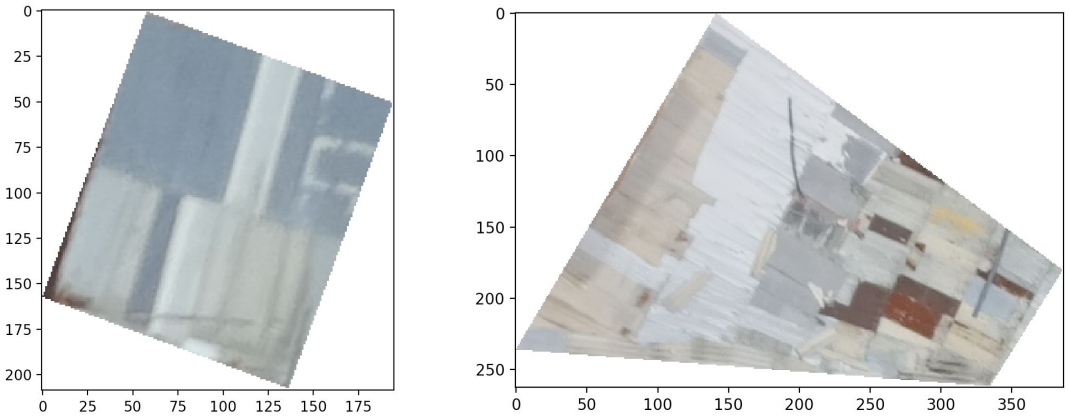
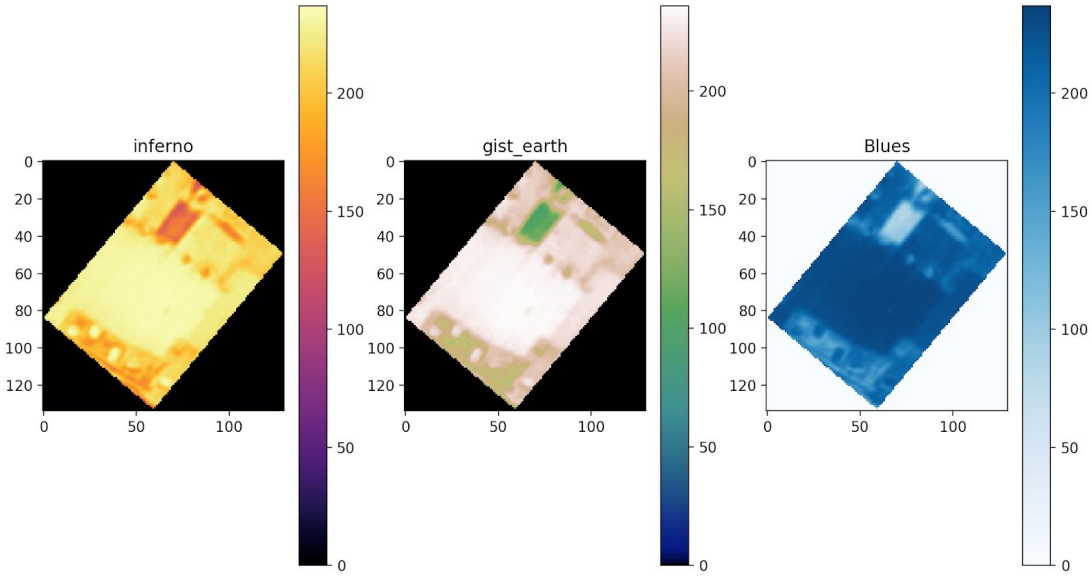


Figure 3. Individual Raster Bands for a Single Rooftop



C. Preprocessing

In order to leverage a set of unsupervised learning technique to group rooftops of similar construction materials, it is necessary to define the parameters of our feature matrix. In this case, each row of data represents a single rooftop or sample. While a matrix of pixel values for a single image can theoretically be presented directly to supervised neural network models in their raw format, we find it helpful to prepare pixel values prior to unsupervised modeling. Each raw image may contain a different number of pixels, hundreds of unique colors, whitespace, and inconsistent rooftop orientations. These characteristics would not only make it computationally expensive to measure pixel distances, but could also introduce unnecessary noise and incorporate empty background space into the model.

In our analysis, we compare the output of different combinations of the following preprocessing and dimensionality reduction steps.

C1. Zonal Statistics

In order to establish a baseline set of results, we begin by calculating a median pixel value for each color band for a single rooftop, and create a simple feature matrix in which the number of features (p) per rooftop (n) is equal to three. This metric is known as calculating Zonal Statistics. For instance, given five rooftops, a feature matrix would contain five rows and three columns, where the first feature vector represents the median pixel value for the red band, the second represents the median pixel value for the green band, and the third represents the median pixel value for the blue band. We write a Python script that calculates zonal statistics for each color band and creates a baseline feature matrix for each country.

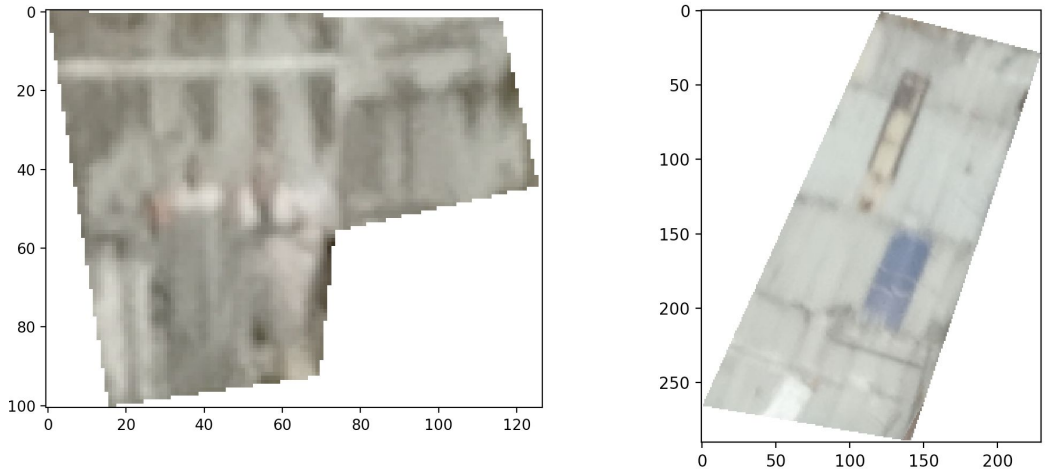
C2. Grayscale, Crop, and Flatten (GCF)

In order to convert a three-layer RGB image into a single layer containing important information from each of the original layers, we leverage grayscaling. Grayscaling images is often sufficient for classification tasks with images, and is far less computationally expensive than leveraging all of the original image layers. Specifically, we apply a linear combination to accomplish this task, in which each pixel in the final output matrix is equal to the sum of 0.299 times the first raster band value, 0.597 times the second raster band value, and 0.144 times the third raster band value. This formula is adopted from Matlab. The resulting matrix has the same size as one original band, but contains important weights from all three bands.

Next, to ensure that each image has a uniform size based on the smallest included rooftop and that each rooftop has the same number of features (p), we use a center cropping technique. By extracting a center square of pixels of a set size from each image, we are able to work around rooftops of irregular shapes or that contain excess background space that would add noise to our models. After consulting with experts across the University of Chicago, we determined that 60x60 pixels is a sufficiently large size to expect accurate output from unsupervised learning techniques. To ensure that this size masks a majority of whitespace in our particular image set, we include a check to drop observations that contain greater than 10% whitespace after cropping, and find that only 2% of images are dropped at scale. Therefore, we conclude that 60x60 is an appropriate size for our analysis.

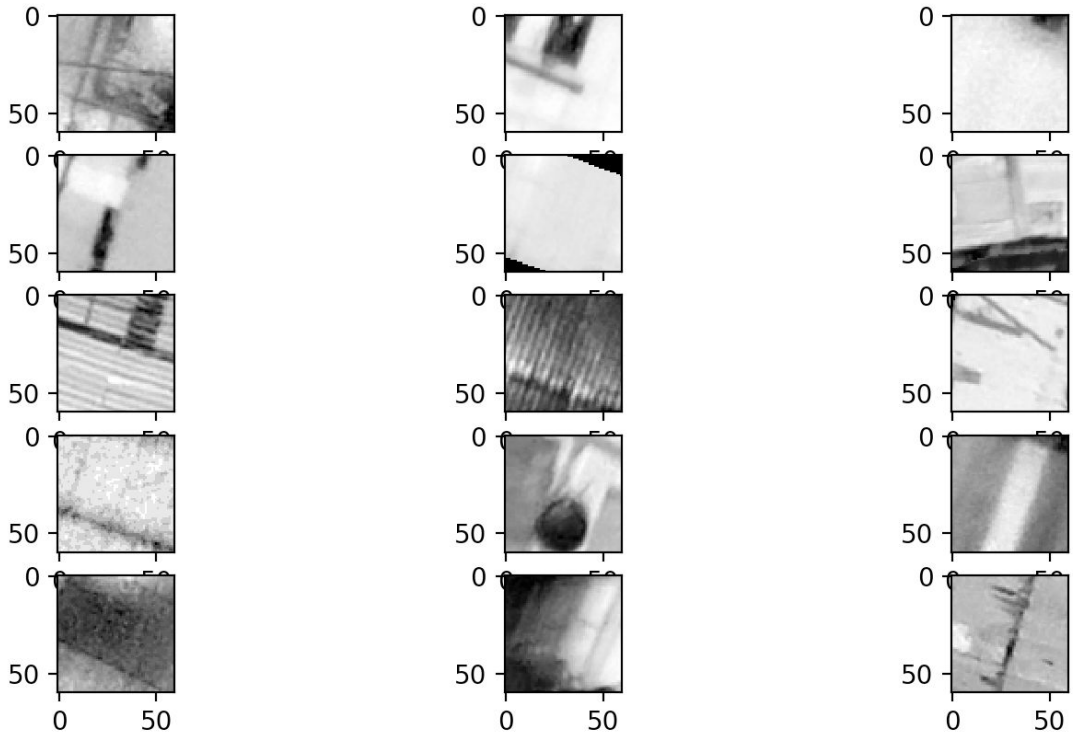
The following images represent a sample irregular rooftop shapes:

Figure 4. Sample of Irregular Rooftop Shapes



Below is the output of a sample of images that have been cropped and converted to grayscale:

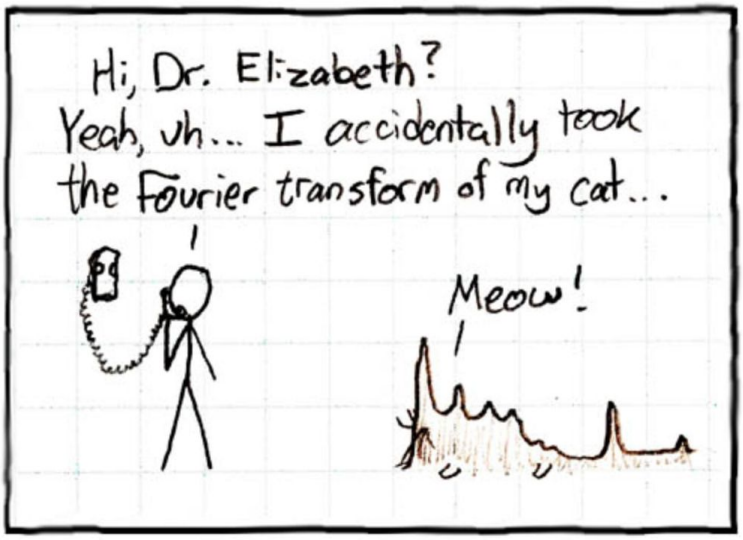
Figure 5. Rooftops Convert to Grayscale and Center-Cropped



Finally, for a single row in the final feature matrix to represent a single rooftop, we flatten the output for each rooftop into an array. To do this, each row of the processed matrix is concatenated to the end of the preceding row, resulting in a single array of length 3600 pixels (originally 60x60).

C3. Fourier Transformation

The Fourier Transformation breaks down an incoming signal into its building blocks. For the purposes of our analysis, we use the Fourier Transformation to convert input images from the spatial domain to frequency domain by decomposing them into their sine and cosine components. With an image represented in the frequency domain, we can display it as a plot of high and low frequencies. Low frequencies provide much greater detail about an image, while high frequencies inform us of points where sharp contrast exists between pixels. These points of high contrast can be interpreted as an edge. In order to focus on distinguishing characteristics, such as the panels on a heavy metal roof, we can use a mask to filter out low frequencies. By excluding low frequencies from the image matrix that we pass to our clustering algorithms, the mask only includes high frequencies (which correspond to high brightness), allowing us to focus on patterns and edges when clustering. For the purpose of our analysis, we cluster on both the full fourier transformation as well as the fourier transformation including a low frequency mask.



C4. Color Quantization via Self-Organizing Maps

Self-organizing maps (SOMs) are a class of unsupervised learning artificial neural networks used for feature detection and dimensionality reduction. SOMs produce low-dimensional, discretized representations of an input space, where geometric relationships between points indicate their similarity. SOMs apply competitive learning rather than error-correlated learning. In competitive learning, nodes “compete” to respond to the input data, and the map learns to differentiate between features based on similarities. In this algorithm, all connection weights are initialized to random values. Next, output nodes compete to be activated, and the winning neuron is called a “winner-takes-all” neuron. This competition forces neighboring neurons to cooperatively organize themselves. Finally, excited neurons decrease individual values of the discriminant

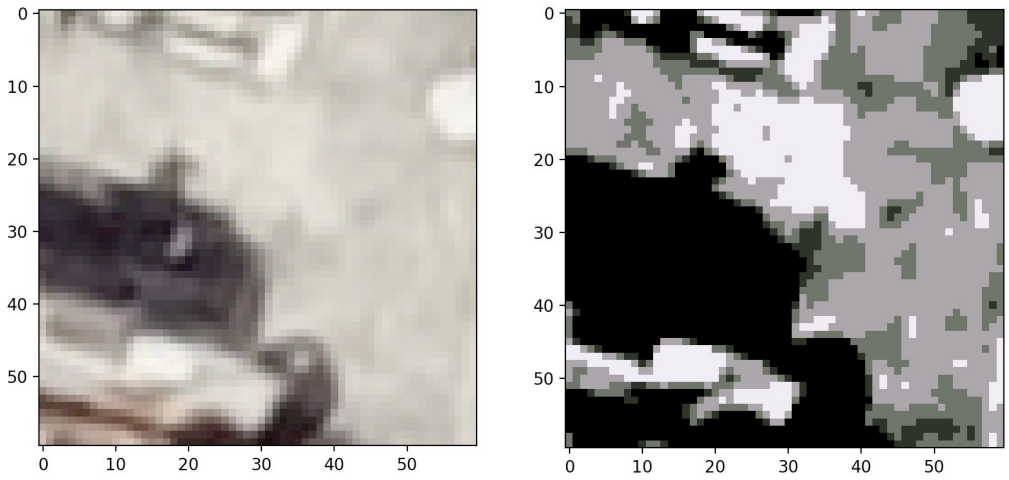
function in relation to the input pattern. The discriminant function is defined as the squared Euclidean distance between the input vector x and the weight vector wj for each neuron j:[1]

$$d_j(\mathbf{x}) = \sum_{i=1}^{D} (x_i - w_{ji})^2$$

To detect patterns in rooftop colors and textures, remove noise, and partition images into distinct segments, we leverage color quantization via SOMs. Color quantization is an example of a technique that is useful in reducing the distinct colors used in an image while making the new image as visually similar as possible to the original image. Specifically, we leverage MiniSom, a numpy implementation for SOMs, to accomplish this goal. We build a 3x3 SOM, meaning that there are 9 final colors in the output of each image. We also specify the radius of the different neighbors in the SOM (sigma) and the learning rate, which determines how much weights are adjusted during each iteration. After several experiments, we find that a sigma of 0.1 and a learning rate of 0.2 produce decent results. While SOM leverages all three layers of an image, the final output can be converted to grayscale as desired.

Below is an example of a cropped rooftop image before and after color quantization:

Figure 6. Results of Color Quantization via Self-Organizing Maps



C5. Other Considerations

We recognize that the selected techniques represent a subset of possible image cleaning and processing strategies used in machine learning. Other possible methods include normalizing, centering, scaling, and rotating images. Given the scale and detail of the dataset, we were unable to construct a generalizable rotation algorithm to detect image direction, whether an image needs to be rotated, and the appropriate number of degrees to rotate it. While it is important that our models do not mistakenly group images by their orientation, we find that image cropping reduces

[1] https://heartbeat.fritz.ai/introduction-to-self-organizing-maps-soms-98e88b568f5d

the extent to which orientation and the resulting background whitespace provide noise. In the future, we are interested to explore data augmentation strategies to compensate for inconsistent orientation. Image data augmentation is a technique that can be used to artificially expand the size of a dataset by creating modified versions of images in the dataset. This process can improve the ability of fit models to generalize what they have learned to new images. In this case, we would consider inputting several rotations of each image into the final training dataset.

D. Clustering Methods

D1. K-Means Clustering

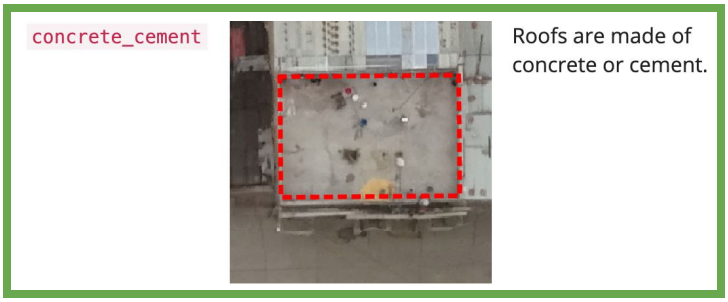
We begin by fitting a k-means algorithm on our data. In k-means clustering, each cluster is represented by its center or centroid, which corresponds to the mean of points assigned to the cluster. We select k-means because it is a hard-partitioning algorithm, where we are interested in strict assignment of observations to clusters, such that every rooftop is a member of only one cluster. Additionally, this method scales well to large datasets, guarantees convergence, and generalizes to clusters of different shapes and sizes.[2] K-means clustering is also particularly prevalent in the image clustering literature. We begin our analysis by defining four to five clusters for each processed dataset. While there are five different rooftop construction material categories labeled in the data, certain .tif files have very low frequencies in the category "other", making four more likely to characterize these particular datasets. We also set the 'nstart' parameter equal to 15 for all iterations. This parameter attempts multiple configurations of the model to find the smallest objective function and reports the best model, generating 15 sets of initialized centroids. As our analysis proceeded, we experimented with setting the number of clusters equal to two instead of five. This decision is reinforced by our purpose of facilitating the prioritization of inspections, which is entirely dependent on a binary classification into "healthy" and "unhealthy" categories. Interestingly, we observe greater accuracy in classifying rooftops in this manner.

Below is a representation of the rooftop construction materials present in the data and how they separate into "healthy" and "unhealthy" categories:

[2] https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

Figure 7. Health vs. Unhealthy Rooftop Construction Materials

Healthy Materials



Unhealthy Materials

D2. Hierarchical Agglomerative Clustering

As a means of comparison and validation, we also fit a Hierarchical Agglomerative Clustering (HAC) model to our processed datasets. HAC iteratively joins together images with strong similarity, and does not require the user to pre-specify the number of clusters. This allows us to iteratively inspect groupings using a bottom-up approach, where each image begins as a singleton cluster at the outset and then successfully merges with similar groupings of clusters until all clusters have been merged. Linkage defines dissimilarity between two groups of observations. In this analysis, we leverage complete linkage, which maximizes inter-cluster dissimilarity and typically yields a balanced dendrogram. Distance refers to the dissimilarity between observations. For our analysis we leverage Canberra distance, which is a weighted Manhattan distance, because it places less emphasis on outliers. This distance metric is

particularly popular for computer security applications, and similarly, and produced interesting results for our analysis as well.

D3. Partitioning Around Medoids (PAM) Clustering

As a final validation technique, we run several iterations PAM or k-medoids clustering on individual .tif files, which represent country-specific samples of the data. This technique is similar to k-means, as it is a hard partitioning algorithm that aims to minimize within cluster variation given a set of non-overlapping clusters k. In k-medoids, a medoid is a data point whose average dissimilarity to all other data points is minimal, making it the most centrally located point in the data. Again, we begin our analysis be defining four to give clusters for each processed dataset, and experiment with setting the number of clusters equal to two instead of give, driven by our motivation to distinguish between "healthy" or "unhealthy" rooftops.

D4. Other Considerations

After visualizing our data points, we initially considered leveraging Density-Based Spatial Clustering of Applications with Noise (DBSCAN), which is a density based clustering algorithm. This method forms optimal clusters based on point densities and spatial densities, and treats distant points as outliers. Given the shape of most of our processed datasets, which typically feature a central grouping of observations and a sparse outer radius of additional points, we had hoped to find unique configurations that could more accurately characterize this complex space. Unfortunately, however, this method did not yield unique, interpretable, or helpful results, so we opted to leave it out of our final analysis.

IV. Findings

This analysis is conducted in both Python and R and implemented on the University of Chicago's Research Computing Center Cluster, Midway2. After running a script to read all GeoJSON and GeoTIFF files, extract building-specific reference dictionaries, execute the appropriate coordinate transformations, flatten each pixel-array into a single row, and union all rows together, we determine that the data include 22,506 rooftops across Colombia, St. Lucia, and Guatemala. We proceed to process this full dataset using seven distinct combinations of processing methods, in order to establish a baseline and compare our clustering accuracy across dimension reduction and transformation techniques. It is important to note that we remove rooftop labels prior to learning any models. Below is a summary of our image processing methods:

- Baseline (3 features per image):
    1. Zonal Statistics
    2. Zonal Statistics & Crop
- Low Dimension Methods (3,600 features per image):
    3. Grayscale, Crop, & Flatten
    4. Grayscale, Crop, Flatten, & Fourier Transformation with Mask
    5. Grayscale, Crop, Flatten, & Fourier Transformation without Mask

6. Grayscale, Crop, Flatten, & Self-Organizing Map

- High Dimension Method (10,800 features per image):
  7. Crop, Flatten, & Self-Organizing Map

After running each of the above image processing methods on both the full dataset as well as smaller regional samples, we apply k-means, HAC, and PAM clustering in R, to visually inspect and numerically assess clustering accuracy, given our original set of construction material labels. Our final data access and processing script (Python) and clustering script (R) can be run directly from the terminal and are available for inspection in our GitHub repository. The following represents a summary of our most prominent findings across these specifications.

Finding 1: The rooftop image data are highly clusterable after pre-processing

To diagnose clusterability, we standardize all input values and construct an ordered dissimilarity matrix (ODI) using a variety of distance measures, including Manhattan, Euclidean, and Canberra. After ordering by spatial proximity, all baseline and low-dimension ODI plots make a strong case for clusterability in the data and confirm that there is likely a cluster structure (see Figure 8). There are clear dark squares along the diagonal, indicating groupings of similar vectors (low dissimilarity), as well as lighter sections on either side demonstrating high dissimilarity between these groupings for all processed files. Some ODI plots highlight approximately four clusters, while others highlight two main clusters. It is interesting to contrast these results with that of the one high-dimension iteration we run using SOM without grayscaling, which displays no clear clusters (see Figure 9). This suggests that dimension-reduction techniques help to remove noise and highlight underlying groupings in the data. While we were able to run the ODI script on each regional .tif file, we ran into computational setbacks running it on the full dataset. Below represents a reduced-dimension, pre-processed ODI in comparison to an ODI run on a high-dimension dataset:

Figure 8. GCF Processing; Canberra Distance
Colombia, n=3151

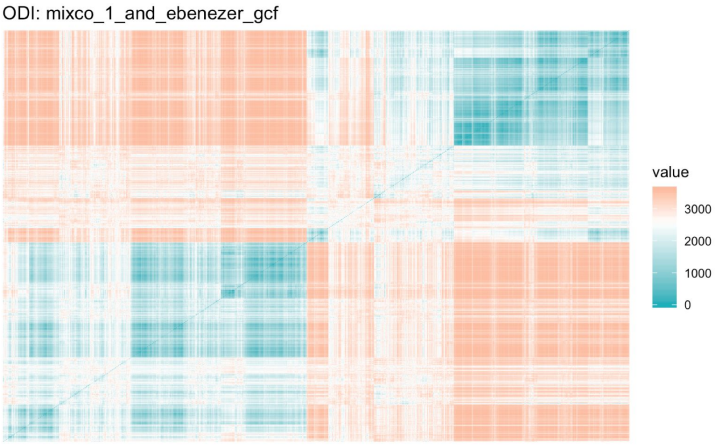
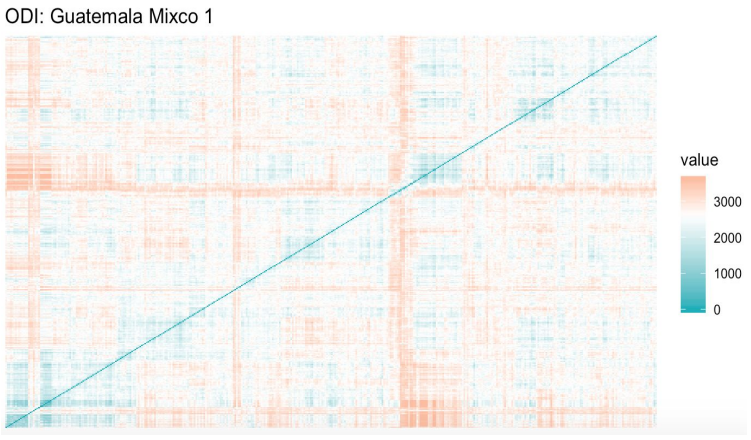


Figure 9. 3-Layer SOM; Canberra Distance
Guatemala, n=400



Finding 2. Binary "healthy" and "unhealthy" classification yields higher accuracy and interpretability than grouping by construction material

We begin our analysis by running k-means, HAC, and PAM with the number of clusters set to 4, with the intention of parsing out up to five construction materials in the dataset: healthy metal, irregular metal, concrete/cement, other, and incomplete[3]. Because approximately 75 percent of the sample consists of healthy metal, we initially find a considerable spread of healthy metals across all clusters, as well as densely clustered points, making it difficult to interpret any underlying distinctions. Figures 10 and 11 represent a sample of our initial output:

Figure 10. Zonal Statistics Clusters, k=4
Full Data, n=22,506

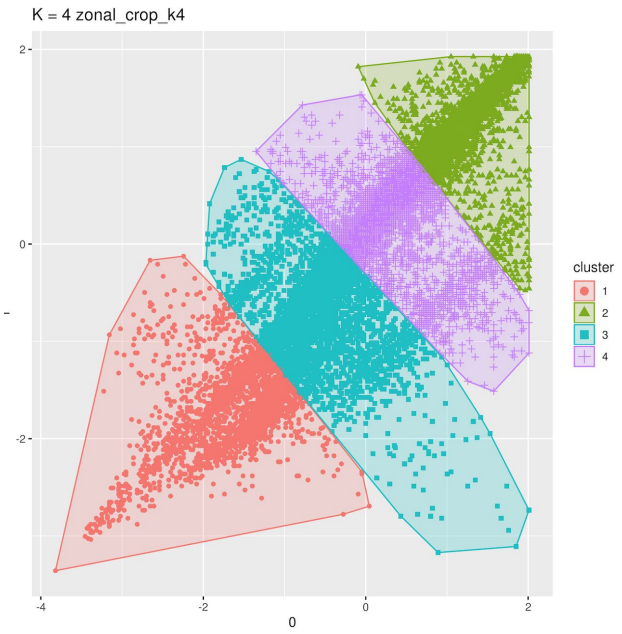
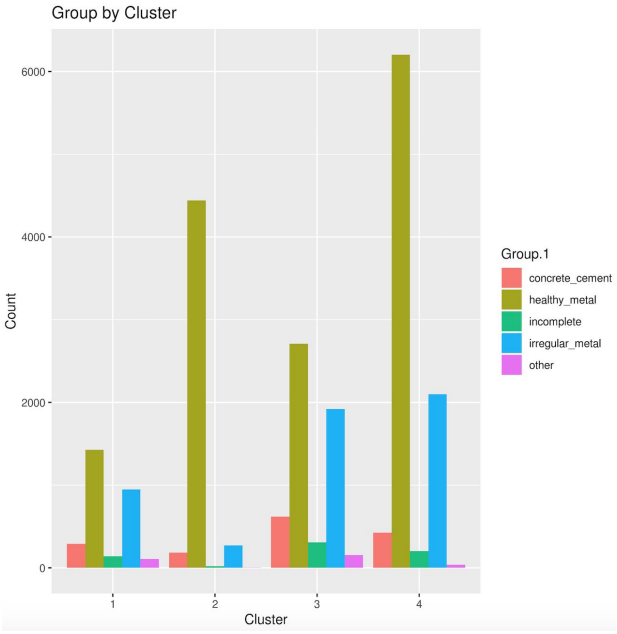


Figure 11. Zonal Statistics Classifications, k=4
Full Data, n=22,506



As our analysis progressed, however, we began to observe clear distinctions between healthy and unhealthy materials, even across specific materials. For instance, in several groups, two clusters would contain a high volume of unhealthy materials, while two clusters would contain a high volume of healthy materials. Given that the primary objective of our study is to assist organizations in prioritizing building inspections in areas at high risk of natural disaster, we pivoted our analysis with this new insight and ran all models using two clusters.

With this relatively minor update, we see dramatic improvements in our output, both in accuracy and interpretability. For instance, in Figure 12 below, which represents baseline zonal statistics run on the entire dataset, cluster one is the dominant cluster for each of the building materials, which provides little actionable insight. In comparison, Figure 13, calculated on the entire dataset with GCF pre-processing, yields striking improvements. Specifically, cluster two captures a majority of healthy metals, while cluster one captures a majority of irregular metals, all else equal. Cluster two contains 82.6% healthy rooftops and only 17% unhealthy rooftops,

[3] Note: "Other" has a very small sample size in the full data, and as a result, we experiment with k=4.

demonstrating a high accuracy score when cluster one is labeled as healthy. While cluster one contains a relatively even distribution of healthy and unhealthy rooftops, this is still an interesting finding given that unhealthy rooftops make up less than one-quarter of the total observations. Therefore, using a list constructed from this specification, a building inspector would have a far greater likelihood of identifying a vulnerable household than by conducting random inspections. Additional improvements in accuracy scores can be viewed in Appendix A.

Figure 12: Zonal Statistics (Baseline), k=2
All locations, n=22,506

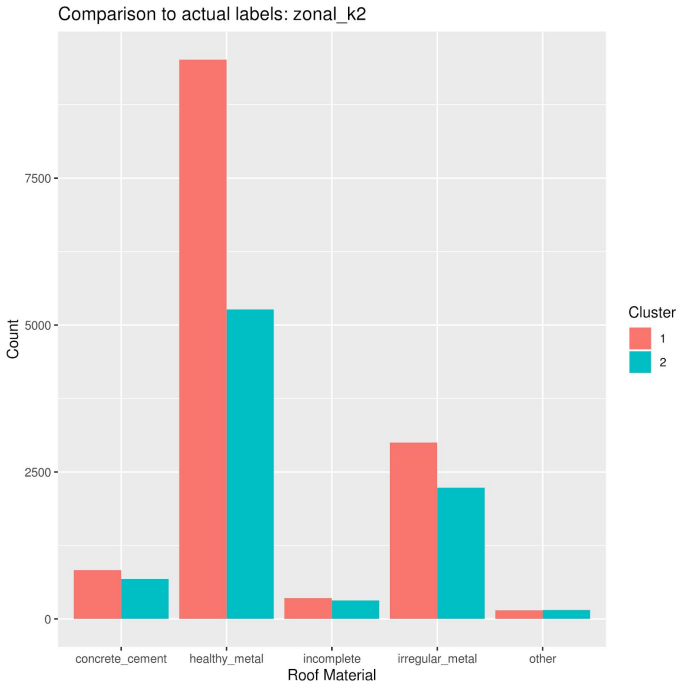
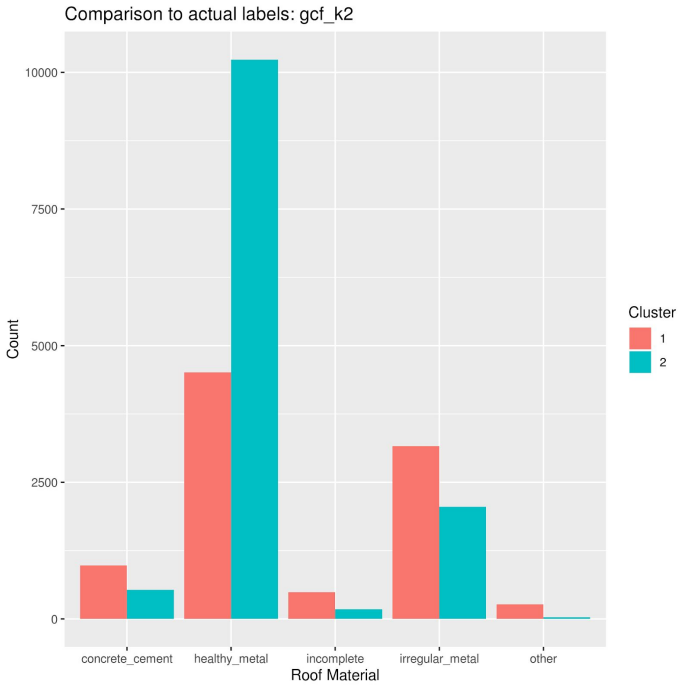


Figure 13: GCF (Low-Dimension), k=2
All locations, n=22,506



Finding 3. Edge detection yields improved results over grayscale and flattening alone (for both k=2 and k=4)

We compare two of our data processing methods -- Grayscale, Crop, and Flatten (GCF), and Fourier Transformation with a high pass filter -- on the same satellite image: Colombia-border-rural.tif. For this analysis, we focus on Hierarchical Agglomerative Clustering. As seen in Figure 14, we do not observe a clear difference among clusters in this particular .tif file after processing the data using GCF on its own. The roofs that belong in our two priority classes (incomplete and irregular metal) appear to be equally present across all four clusters. While we do see that a majority of roofs that contain healthy metal come from cluster four, the same could be said among the majority of roofs that contain irregular metal. This inability to clearly distinguish among priority classes proves that HAC paired with GCF is insufficient.

Figure 14: HAC, GCF
Colombia, n=1971

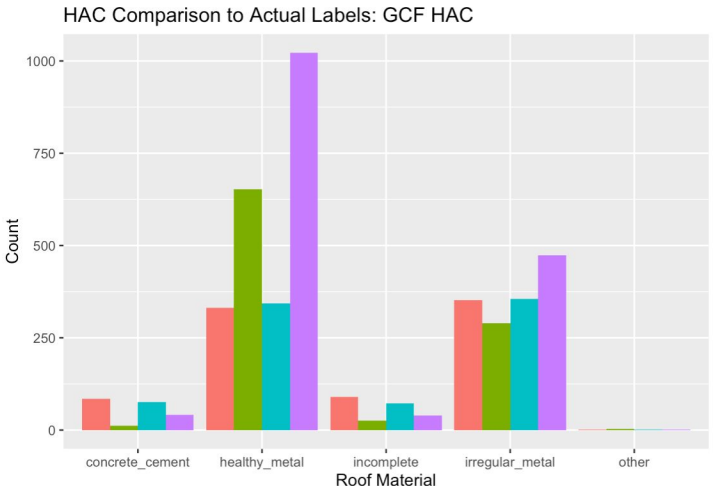
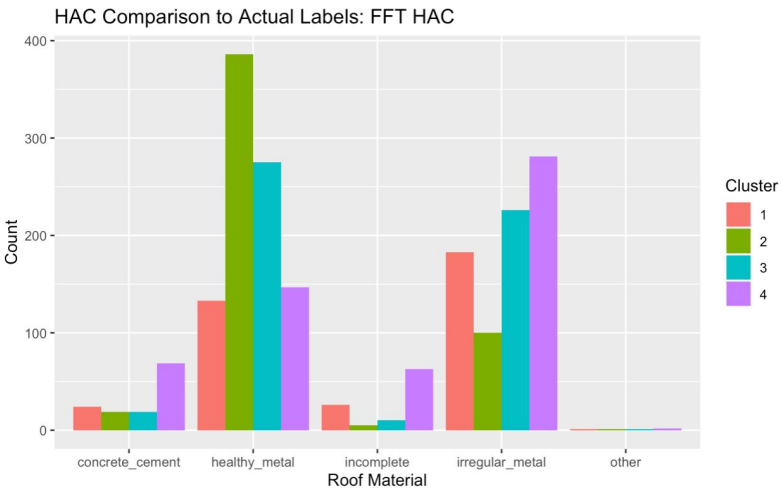


Figure 15: HAC, Fourier Transformation, High Pass Filter
Colombia, n=1971



Compared to the results in Figure 14, we see that the HAC paired with the Fourier Transformation much more clearly identifies our priority classes. We find the majority of healthy metal roofs and the fewest irregular metal roofs in custer two. Conversely, we observe that the majority of irregular metal roofs and very few healthy metal roofs are found in cluster four. We believe this proves the value of including edge detection and pattern recognition when designing an image feature matrix. After we conduct the Fourier Transformation on each of our images, we add a high pass filter that blocks low frequencies. We pass the resulting matrix that primarily contains pixel values for points of high contrast to the hierarchical clustering algorithm. The focus on high frequencies, which represent edges and other large shifts between pixel values, appears to improve our ability to identify priority classes.

Finding 4. At scale, image pre-processing improves the interpretability and accuracy of clusters

As demonstrated across our baseline results leveraging zonal statistics, a majority of figures yield nearly proportional construction material breakdowns within each cluster, demonstrating weak partitioning. This result can be difficult to interpret. In Figure 16 below, which does not involve extensive pre-processing and assesses a small dataset, cluster one is evenly distributed across healthy and irregular metals, and no clear patterns immediately emerge from the results. Because this project aims to parse out subtle differences between healthy and irregular materials in order to help prioritize inspections and failure mitigation efforts, our analysis layers in detail in the form of image segmentation and an enhanced feature matrix based on the outlined processing steps. In comparison, Figure 17 below, which leverages GCF pre-processing and a larger dataset, displays a clear distinction between healthy and irregular metals that is far more actionable for inspectors focused on identifying high risk households. Both the GCF and the Fourier Transformation (without mask) models demonstrate significant improvements in discerning between materials using k-means, HAC, and PAM.

Figure 16. Baseline Clustering Results
Colombia, n=400

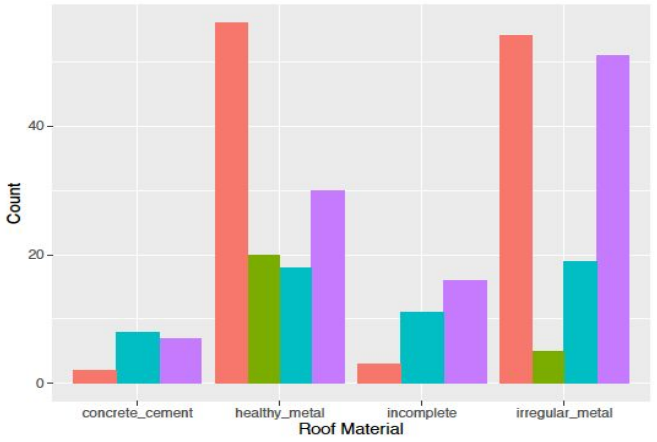
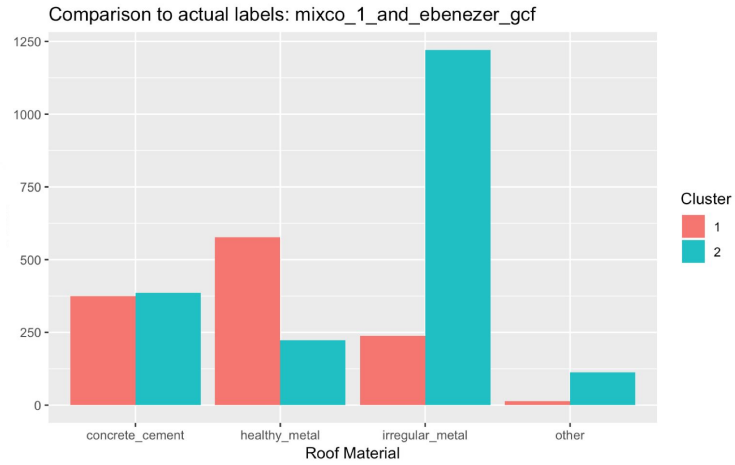


Figure 17. Pre-processed clustering results
Guatemala, n=3,151



V. Conclusion

While we begin our analysis focusing on leveraging high definition, aerial imagery to identify and label rooftops by construction material using color quantization and unsupervised learning techniques, clear distinctions between "healthy" and "unhealthy" materials emerge throughout our analysis, prompting a two-cluster approach. This insight spans across k-means, HAC, and PAM clustering methods. We consider this finding to be a significant contribution to current strategies of random household selection and manual inspection. Throughout this analysis, we are also able to identify a strong case for clusterability in the DataDriven rooftop dataset, which is enhanced by dimension reduction and image pre-processing techniques.

We demonstrate that leveraging grayscale, cropping, and flattening outperforms the calculation of image-layer aggregate statistics, such as median value, in clustering algorithms. Additional processing techniques, such as Fourier Transformation and Self-Organizing Maps, further improve our clustering accuracy. We believe that we can continue to improve upon this analysis by exploring additional image-processing methods, including data augmentation, rotation detection, and deep neural networks, as well as density-based clustering algorithms given the shape of our data. We conclude that unsupervised learning techniques can significantly contribute to disaster mitigation efforts where labeled data is ultimately impractical to collect.

Appendix A. Accuracy Score by Processing and Clustering Method

Model	Clustering Method	Cluster Counts & "Accuracy"
Zonal Statistics	K-Means, k=2	
Zonal Statistics + Crop	K-Means, k=4	
Grayscale, Crop, + Flatten	K-Means, k=2	
Grayscale, Crop, Flatten, + Fourier w/o Mask	K-Means, k=2	
Grayscale, Crop, Flatten + Self-Organizing Map	K-Means, k=2	

	status	cluster	count	percent
1	healthy	1	10353	74.7131414
2	unhealthy	1	3504	25.2868586
3	healthy	2	5946	68.7557817
4	unhealthy	2	2702	31.2442183

	status	cluster	count	percent
1	healthy	1	1718	58.9972527
2	unhealthy	1	1194	41.0027473
3	healthy	2	4624	94.0219601
4	unhealthy	2	294	5.97803985
5	healthy	3	3328	58.3041345
6	unhealthy	3	2380	41.6958655
7	healthy	4	6629	73.9266198
8	unhealthy	4	2338	26.0733802

	status	cluster	count	percent
1	healthy	1	5491	58.3838384
2	unhealthy	1	3914	41.6161616
3	healthy	2	10770	82.6617545
4	unhealthy	2	2259	17.3382455

	status	cluster	count	percent
1	healthy	1	10485	83.2671537
2	unhealthy	1	2107	16.7328463
3	healthy	2	5386	58.0137872
4	unhealthy	2	3898	41.9862128

	status	cluster	count	percent
1	healthy	1	5012	71.4061832
2	unhealthy	1	2007	28.5938168
3	healthy	2	3173	75.3323837
4	unhealthy	2	1039	24.6676163
5	healthy	3	4867	71.9331954
6	unhealthy	3	1899	28.0668046
7	healthy	4	3171	72.0845647
8	unhealthy	4	1228	27.9154353

VI. References

COGEO. “Cloud Optimized GeoTIFF” COGEO. https://www.cogeo.org/ (accessed October 16, 2019).

DrivenData. “Open AI Caribbean Challenge: Mapping Disaster Risk from Aerial Imagery” Driven Data. https://www.drivendata.org/competitions/58/disaster-response-roof-type/page/144/ (accessed October 16, 2019).

Ivanov, Ivan-Assen. “Image Compression with Vector Quantization” Informatech. https://www.gamasutra.com/view/feature/131499/image_compression_with_vector_.php

Myers, Joe. “Which natural disasters hit most frequently?” World Economic Forum. https://www.weforum.org/agenda/2016/01/which-natural-disasters-hit-most-frequently/ (accessed October 16, 2019).

Penatti, Otavio A. B., Keiller Nogueira, and Jefersson A. dos Santos. 2015. “Do Deep Features Generalize from Everyday Objects to Remote Sensing and Aerial Scenes Domains?” CVPR2015 Workshop Paper. Computer Vision Foundation, June 2015. https://www.researchgate.net/publication/301921752_Do_deep_features_generalize_from_everyday_objects_to_remote_sensing_and_aerial_scenes_domains.

USGS. “How can climate change affect natural disasters?” USGS. https://www.usgs.gov/faqs/how-can-climate-change-affect-natural-disasters-1?qt-news_science_products=0#qt-news_science_products (accessed October 16, 2019).

Wasser, Leah, Chris Holdgraf, and Martha Morrissey. “Lesson 4. About the Geotiff (.tf) Raster File Format: Raster Data in Python” EarthLab. https://www.earthdatascience.org/courses/earth-analytics-python/lidar-raster-data/intro-to-the-geotiff-file-format/ (accessed October 16. 2019)

VII. Contribution Statement

We both contributed to nearly all facets of this project -- including the research proposal, data processing, analysis, and interpretation -- and thus feel that our efforts have been equal.