Classifying Damage Level Disaster-Induced Images Via Computer Vision

Shirley Li*, Kofi Mireku*, and Brent Thorne*

*Data200S, UC Berkeley MSSE

Abstract

In time-sensitive disaster situations, it is imperative for emergency response efforts to have access to quick and accurate information about an disaster area. Satellite imagery provides the perfect mechanism in which to provide response teams with live, high dimensional data about disaster type and progression across the world. However, the sheer amount of data produced by satellite imagery is impossible to completely review by human eye, generating a need for computational models to help with live and accurate assessment of disaster scenarios. By utilizing image pre-processing techniques, label balancing tools, and various implementations of machine learning model architectures, we produce a deep learning convolutional neural network trained to classify disaster types and damage levels, thus providing a machine learning solution which can thereby significantly increasing first responded efficacy and life expectancy in disaster scenarios across the world.

Keywords: Feature Extraction, Computer Vision, Image Classification

1. Introduction

Natural disasters occur all around us. For the most part, they occur without much notice and leave disaster and destruction in its wake. In lieu of the great danger, loss of life and property, they cause, the need for quick and effective responses from first responders to mitigate their impact has never been so essential especially in this modern era where cities get overpopulated with life and property. Up until recently, most damage assessment for the impact of disasters, whether natural or man-caused have necessitated eye-witness accounts, manual surveys and real time communication, these methods were extremely time consuming, prone to inherent bias and relied too heavily on radio communication which could be ineffective especially during major disasters. Recently, computer vision has emerged as a solution to the drawbacks of traditional damage assessment by offering a novel automated process of damage classification using computer vision and image analysis algorithms to classify the severity of destruction efficiently and accurately.

Computer vision is a field of artificial intelligence (AI) that uses machine learning and neural networks to teach computers and systems to derive meaningful information from digital images, videos and other visual inputs [1]. Alternative applications of computer vision with semantic segmentation models like U-Net, leveraging CNN's for biomedical image segmentation [2]. Other applications have been in urban resilience plan study using remote sensing [3]

This paper aims to explore EDA and algorithmic approaches to disaster type classification and damage level classification that will be relevant to emergency response institutions. Previous papers utilize computer vision models for applications in environmental monitoring problems [3]. However, in this paper, we present a novel application and use-case for these state-of-the-art computer vision machine learning models for use by disaster first responders. We present effective preprocessing and featurization pipelines for tackling efficient computer vision algorithms, classifiers that automatically categorize images derived from the xView2 Challenge Dataset[4][5] to the type of disaster scenario and building damage in 2 tasks:

- Task A: Classifying images from the midwest-flooding disaster and the socal-fire disaster.
- Task B: Classify damage levels for images from the hurricane Matthew disaster.

2. Data Overview

Images used to train the computer vision models were sourced from Maxar/DigitalGlobe Open Data Program [4], which provides open-source high-resolution satellite imagery from a variety of natural disasters. In particular, the models were trained on a subset of data extracted from the xBD Dataset [5]. This subset of data comprises more than 33 thousand RGB satellite images of buildings post various natural disasters taken before and after major crisis events.

75% of these images (denoted as images from the training set) are annotated with one of three disaster categories: Midwest US floods, Socal fire, and Hurricane Matthew. Images were also labeled with building levels of damage, numerically categorized from 0 (no damage) to 3 (destroyed). The remaining 25% of provided images are unlabeled (denoted as images from the test set.)

xBD image data is captured in a NxMx3 numpy array of integers from 0 to 255, where the first two dimensions are variable in size to represent the image size, and where the third dimension captures the pixel intensities of the red, blue, and green channels respectively. These images, despite having black pixel artifacts and occasional blurring possibly from geometric transformations, maintain consistent scale and resolution. Some images are shifted or have a high aspect ratio to highlight specific structures.

Damage Level Disaster Type	0	1	2	3
Fire	7204	69	43	1064
Flood	6734	114	97	59
Hurricane	2631	5236	1544	1740

Table 1: Image Label Distribution in xBD Training Set.

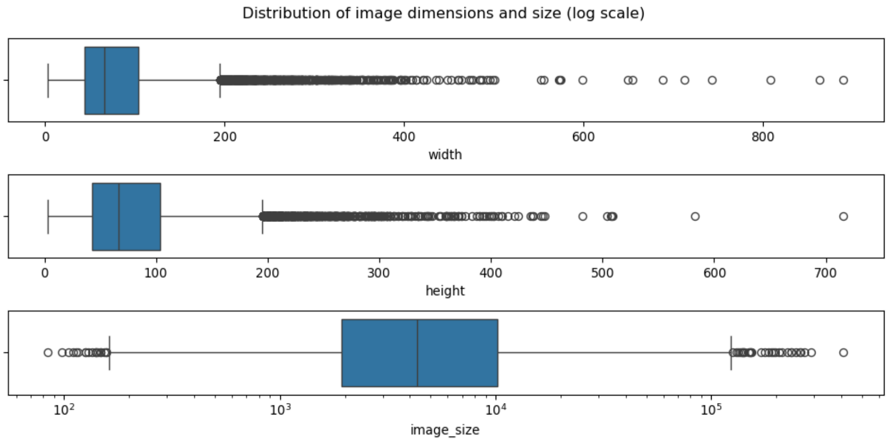

Figure 1: Image size distribution in xBD dataset.

2.1 EDA: Training Label Distribution

The provided subset of xBD training data contains heavy label imbalance across both damage levels and disaster types (Table 1). For instance, there are a disproportionate amount of building images labeled with zero damage. Fire and flood images in particular contain the poorest represented subsets of images, with images labeled with damage levels 1 and 2 capturing less than 350 images in the entire set of more than 15k annotated fire or flood images.

2.2 EDA: Training Image Size Distribution

Image size distribution within the xBD dataset is similarly variable across all disaster types and damage levels (Figure 1), containing image sizes anywhere between 84 and 410464 pixels with similarly variable aspect ratios.

2.3 EDA: Training Image Color Intensity Distribution

When viewing the distribution of average and standard deviation of color intensities across each channel for images in the xBD dataset, it is apparent that there are differences across the images in each category.

In our visual analysis, we sampled images and extracted the average pixel value for each image to see if categories of images are discernable from their aggregate values. When sampling the average color for images across disaster types, we can see a slight visual

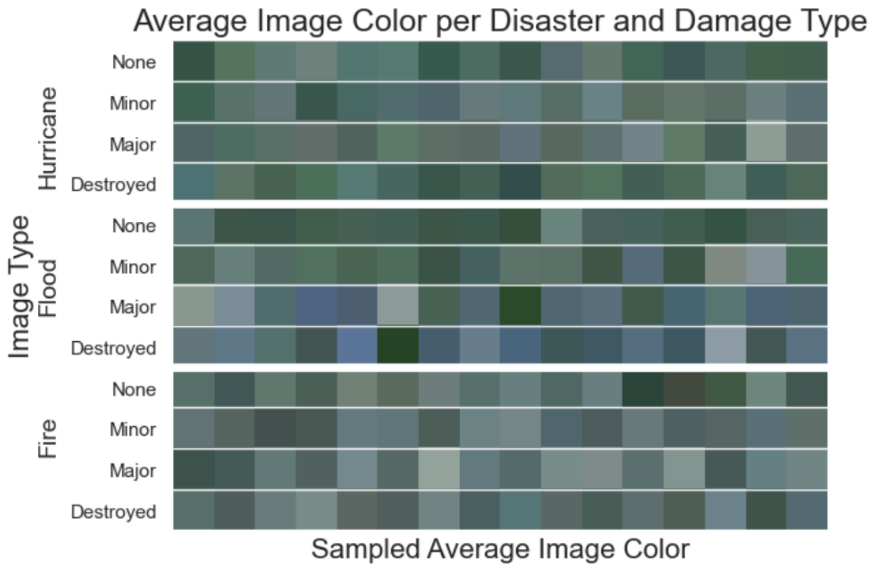

Figure 2: Average image color analysis across label categories.

difference between the categories, especially when comparing against the most extreme damage levels (Figure 2). Images of buildings with higher damage levels (major damage or destroyed) from hurricanes have similar average colors to building images with little to no damage at all. However, the average color for majorly-damaged or destroyed buildings from floods appear much more blue, while those damaged from fire average to greyer colors.

2.4 Training Image Pre-processing

In order to optimize the performance of image classification models, image data was pre-processed prior to model input. For one, images were cropped and resized to a standard size using the Python package cv2, with the final image dimensions depending on the model being trained (see Section 3). This was done in order to standardize model inputs and allow for the models to classify images of all types of sizes. Pixel intensities are further normalized such that values range from 0 to 1 instead of 0 to 255 to improve general model stability and performance.

In order to prevent over-fitting of model accuracy towards the most represented image types, images were sampled at various rates according to the two separate classification tasks such that the final training image set contained equal proportions of the respective classification category. For instance, the disaster type classification models were trained on flood and fire images which downsampled the number of zero damage images. Damage level classification training data was similarly generated by equally sampling from all four damage level types in the hurricane image set.

To generate and process training and validation data for the model, several steps are followed. Initially, the dataset undergoes preprocessing, including resizing images to 180x180 pixels and normalizing pixel values to a float between 0 and 1. For the Type

Classifier, the level and type are encoded to floats between -1 and 1, while for the Level Classifier, the levels are one-hot encoded into categories from 0 to 4. To address class imbalance, a combination of oversampling and undersampling techniques is employed. The minority class is oversampled using replacement, and the majority class is undersampled by randomly deleting rows to achieve balance. Additionally, data augmentation techniques, such as flipping and rotating images, are applied to increase the diversity of the training set.

3. Methods

To solve each classification task, two types of machine learning models were implemented: a logistic regression model trained on flattened pixel intensity data compressed in PCA space, and a convolutional neural network trained on unflattened images (CNN). The first model takes advantage of the computational simplicity of a logistical regression model; because the optimum feature weights of a logistic regression can be mathematically solved, the model completely avoids potential issues with training hyper-paramterization. At the same time, the first model uses singlar value decomposition to mathematically convert the large input space of flattened image pixel intensities into a much smaller space of orthogonal features. These orthogonal features can then eliminate model input co-linearity such that model performance is further optimized.

While CNNs are deep learning models which require heavier computational power and fine-tuning of training parameters (such as learning rate, optimizer, and batch size) to optimize performance, they are well-suited for image tasks due to their ability to capture spatial hierarchies within images, with popular CNN architecutures such as ResNet50 proven to be successful in various computer vision classification tasks [7]. Because of this, a CNN model was also trained to evaluate both classification tasks.

To train both models, the set of provided xBD annotated images were split into a 90% training and 10% validation set, which model evalaution, selection, and hyperparametrization was conducted by observing changes in validation performance, which the model does not learn to optimize weights from. Training and validation sets were constructed from xBD images using the same workflow as described in Training Image-Preprocessing 2.4.

A detailed walkthrough of code for creating data and training models were developed in Jupyter notebooks found in this GitHub repository:

https://github.com/fractalclockwork/Data200.

3.1 Logistic Regression on PCA-Compressed Features

For all logistic regression classification tasks, RGB images were standardized to a 24x24 size, with intensities normalized between 0 and 1. Subsequently, these normalized images were flattened into an array of length 1728 and further compressed into a feature vector of size 400 using sklearn's principal component analysis (PCA) tool.

Hyperparameter tuning for the model involves determining the optimal number of components to compress the images to. This decision is based on PCA's ability to explain the percentage of variance in the data, ensuring that the selected number of components retains sufficient information for the classification task.

To assess the performance of the model and manage the bias-variance tradeoff, the training accuracy is compared with the validation accuracy. A significant disparity between these metrics suggests potential overfitting, highlighting the importance of ensuring the model's generalizability beyond the training data.

3.2 Convolutional Neural Network

Convolutional Neural Network (CNN) classification models for both tasks were implemented using PyTorch and Tensorflow, enabling the construction and training of deep learning networks. The CNN architecture was leverages the knowledge and weights from the pre-trained image VGG16 [8], which incorporates multiple convolutional layers followed by pooling layers to extract features and reduce dimensionality. The final layer utilizes softmax activation to determine the probability of an input image belonging to the fire category.

In CNN classification tasks, all images were standardized to a size of 120x120, with intensity values normalized between 0 and 1. The chosen loss function to train the model against was sparse categorical crossentropy, optimized using the Adam optimizer. Hyperparameter tuning for the model involved iterating over parameters such as learning rate and number of epochs. Similar to the logistic regression model, performance and accuracy for such hyperparameter iterations is measured by comparing training accuracy and loss to those in the validation set.

4. Results

4.1 PCA Image Compression

With a larger number of components, PCA compressed images can capture a larger proportion of image data. However, to prevent overfitting and bloating of logistic regression model size, the number of PCA components need to be kept at a minimum. In order to decide on the number of features to compress flattened image data in PCA space, cumulative explained variance was calculated across a sample of PCA-transformed images projected into a range of n-dimensional features (see Figure 3). This data suggests that 100 features is sufficient to capture nearly 100% of the variance explained by sampled image, thus supporting our choice of 100 input features for our logistic regression model.

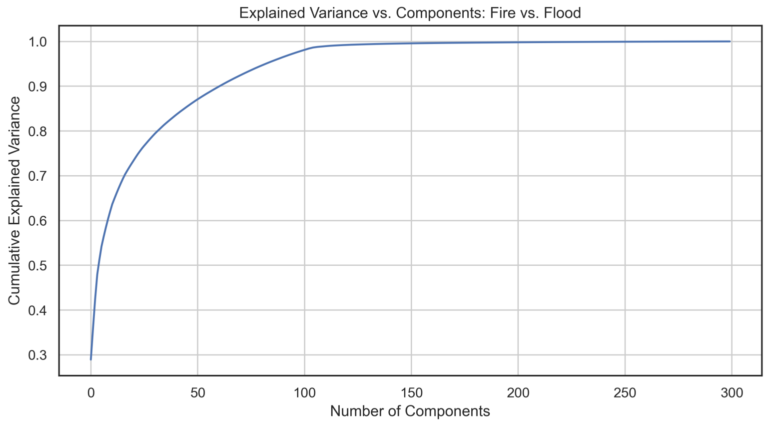

Figure 3: Cumulative variance explained of images compressed into variable number of PCA features.

Precision:	0.8487
Recall:	0.8586
F1-Score:	0.8492
Accuracy:	0.8586

Table 2: Logistic regression model performance on validation set for classification task A.

4.2 Task A: Disaster Type Classification

4.2.1 Logistic Regression on PCA-Compressed Features

Even with using PCA-compressed images as features, the logistic regression model does an adequate job at classifying between fire and flood images, with a precision, recall, and accuracy of approximately 85% (see Table 2). These results were generated from the validation set, suggesting that the logistic regression model is able to generalize from the patterns it observes in PCA space from the training data.

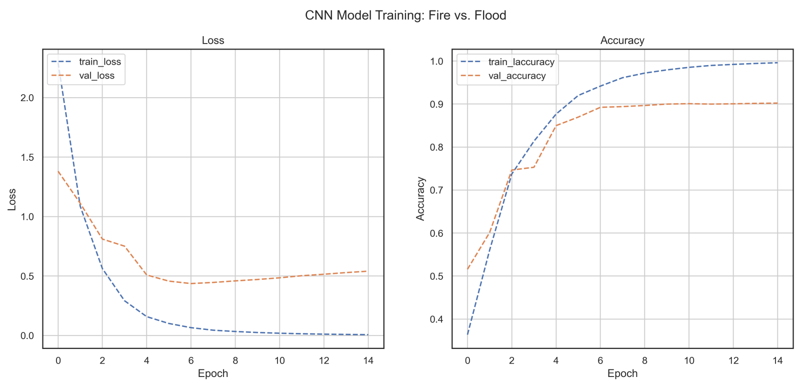

Figure 4: CNN loss and accuracy on training and validation data from on task A classification task.

Damage Level	Precision	Recall	F1-Score	Support
Flood (0)	0.9567	0.9622	0.9594	1401
Fire (1)	0.9682	0.9636	0.9659	1676
Accuracy			**0.9630**	3307
Macro Avg	0.9625	0.9629	0.9627	3307
Weighted Avg	0.9630	0.9630	0.9630	3307

Table 3: CNN Accuracy on validation data for task A classification task.

4.2.2 Convolutional Neural Network

When trained on the xBD image dataset, the CNN model succeeds at classifying fire and flood images. Observing the changes in training and validation loss for this trained convolutional neural network, the model only has a slightly lower training loss than validation loss, suggesting that the model is not egregiously overfitting to the training data to cause higher error in the validation set (see Figure 4).

Indeed, when evaluating the performance of the trained model on the validation set, the CNN produces a very high accuracy, successfully classifying more than 96% of images it was not trained on (Table 3). Additionally, there appears to be no major differences between the precision and recall of fire vs. flood images, insinuating that the model does not contain major blind spots or biases against certain types of fire or flood images. This suggests that the CNN model is able to equally generalize the patterns it learned from in training images into future unseen images, all at a higher capacity than what the original logisitic regression model was able to perform at.

4.3 Task B: Damage Level Classification

4.3.1 Convolutional Neural Network

Based on the results of the CNN model's performance on the first fire-flood classification task, the same architecure was chosen to evaluate damage level classifications on hurricane images for task B.

Observing the changes in training and validation loss for this trained convolutional neural network, the model has a significantly lower training loss than validation loss, suggesting that there is some major overfitting to the training data (see Figure 5). This may occur if the model learns to memorize the training data, or if the training data is not representative of what is seen in the validation set. In future iterations we plan to implement regularization techniques such as dropout and play around with different hyperparameter values to mitigate overfitting.

Looking the trained CNN's performance on the task B validation set, our accuracy is also significantly lower than that of task A (Table 4), with a combined overall accuracy of 0.55. When comparing the classification accuracy of this CNN split by damage level,

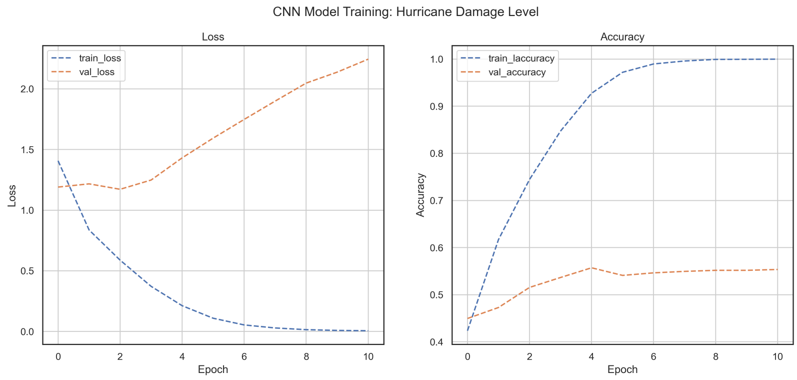

Figure 5: CNN loss and accuracy on training and validation data from on task B classification task.

Damage Level	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.50	0.44	0.47	526
1	0.61	0.70	0.65	1048
2	0.29	0.18	0.22	309
3	0.57	0.62	0.59	348
Accuracy			**0.55**	2231
Macro Avg	0.49	0.48	0.48	2231
Weighted Avg	0.53	0.55	0.54	2231

Table 4: CNN accuracy on validation data for task B classification.

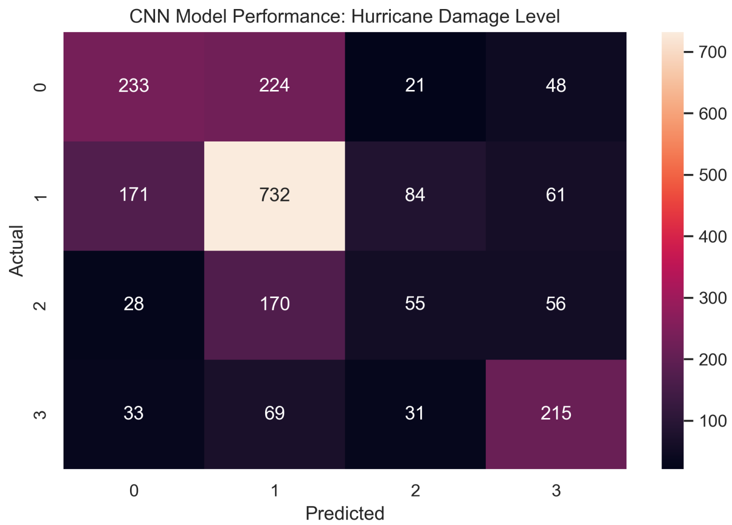

Figure 6: Confusion matrix of CNN model performance on task B classification task.

it is apparent that this decrease in performance is due to the model's inability to classify specific levels which were not well represented in the original xBD dataset (Figure 6. While the CNN was able to accurately predict images labeled as damage level 1, with relative success for damage level 0 and 3, the model had very low precision for images in damge leve 2. This implies that our model needs more fine tuning on generalized unseen data, and that further image label balancing methods need to be conducted to improve the CNN's overall performance.

4.4 Overall Analysis

Analyzing these results, it appears that additional data augmentation, label balancing and sampling, or model hyperparamatrization is required to best improve our model's performance across both task A and task B. The CNN model appears to outperform the Logistic Regression model for Task A classification - however, it is unclear if the CNN model's overfitting to training data is causing deleterious effects on validation accuracy. Comparison between CNN and logistic regression performance for the task B classification problem remains to be seen, but initial data suggests that task B will require additional improvements for both models to see adequate validation accuracy.

5. Discussion

In conclusion, singular value decomposition and image-oriented deep learning models can be utilized to perform on a variety of image classification tasks. In further iterations, the performance of these models can still be further improved; for instance, these models could be trained on additional image data. The xBD dataset provides more satellite imagery of other disaster types, increasing the amount of data trained on and possibly decreasing effects of label imbalance on the amount of dropped data. There are also a variety of other machine learning architectures that could be used to also help classify images such as the U-Net [2] CNN model, which can produce a condensed latent vector (similar to the PCA-compressed image feature) for images based on an image-to-image training regimen.

One limitation in the implementation of these computer vision classification models was the compute power required for training deep learning models such as the convolutional neural network. Without access to stronger compute or GPU resources, hyperparameter iterization was limited by the speed of local CPU resources. Greater computational power would have also allowed for the testing of even larger CNN architectures with a greater number of layers or kernels, which may also boost performance.

During improving model performance, it was found that adding non-storm images to the storm-based damage type classification model helped to improve the model's accuracy. Although the images were unrelated to the disaster type used for the classification task, it appears as if the additional of more image data to train on was able to inform the convolutional neural network how best to capture information from images in general. For instance, the incorporation of fire or flood images may have provided greater color intensity diversity for the model to observe. This discovery corroborates previous findings in the field of machine learning and computer vision, where CNN models can leverage large unspecific images to train a bulk of the model, and later fine-tune to specific tasks with smaller image datasets [6].

While developing these computer image classificaiton models, it is imperative to also consider on the societal and ethical effects of these models. For one, the classification models developed in our research are trained on a dataset consisting of specific disaster types prevalent in U.S. locations and terrains. Consequently, these models are likely overfitted towards U.S. terrain characteristics and may not effectively generalize to evaluate satellite imagery from other geographical regions or different types of disasters. Our model is also developed to evaluate damage specifically to buildings. This approach neglects damage assessment for critical infrastructure and natural elements (e.g., bridges, streets, rivers, forests) that are vital for community resilience and disaster recovery worldwide. By prioritizing building-centric evaluations, our classification model may inadvertently overlook key aspects of disaster impact that are essential for comprehensive recovery efforts in diverse regions and contexts.

An essential aspect of our study involves the annotation of image data. The labels indicating the degree of damage to buildings are human-curated and thus inherently subjective and potentially biased based on the annotators' perspectives. This introduces a moral and ethical concern, as misclassifications or biases in the annotated data may propagate into the classification models. Such biases could lead to response teams underestimating the impact of natural disasters on human lives, affecting the allocation of resources and emergency response efforts.

In summary, while our research contributes to advancing disaster classification techniques using CNNs, it is essential to acknowledge and address the ethical implications and potential societal biases associated with dataset composition, annotation processes, and the scope of damage assessment. Future efforts should prioritize inclusive and diverse datasets, thorough model interpretation, and holistic disaster impact assessments to ensure the ethical and equitable deployment of AI technologies in disaster response and recovery initiatives.

5.1 References

1. IBM, What is computer vision? https://www.ibm.com/topics/computer-vision

2. Ronneberger, O., Fischer, P., Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18 (pp. 234-241). Springer International Publishing.

3. Kerle, Norman Ghaffarian, Saman Nawrotzki, Raphael Leppert, Gerald Lech, Malte. (2019). Evaluating Resilience-Centered Development Interventions with Remote Sensing. Remote Sensing. 11. 2511. 10.3390/rs11212511.

4. Gupta, R. et. al, (2019). xBD: A Dataset for Assessing Building Damage from Satellite Imagery. Retrieved from https://arxiv.org/pdf/1911.09296

5. xView2. Computer Vision for Building Damage Assessment using satellite imagery of natural disasters. Retrieved from https://xview2.org/

6. Reyes, A.K., Caicedo, J.C., Camargo, J.E. (2015). Fine-tuning Deep Convolutional Networks for Plant Recognition. Conference and Labs of the Evaluation Forum

7. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)

8. Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.