DEEP LEARNING FOR DAMAGE ASSESSMENT AFTER NATURAL DISASTERS

By

Tristan A. Goers, B.S.

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

in

Interdisciplinary Studies

University of Alaska Anchorage

May 2021

APPROVED:

Caixia Wang, Ph.D., Committee Chair
Kenrick Mock, Ph.D., Committee Member
Tom Ravens, Ph.D., Committee Member
Caixia Wang, Ph.D., Chair
Department of Geomatics
Jennifer Brock, Ph.D., Associate Dean for Academics
College of Engineering
Mary Jo Finney, Ph.D., Dean
Graduate School

ABSTRACT

In recent decades, due to global warming and climate change, we have witnessed ever-growing occurrences of natural disasters such as flooding, tornadoes, earthquakes, and wildfires. As such, it is more important than ever to provide emergency response personnel with accurate and timely information for their effective responses to crises. Among the variety of information which is needed for emergency responses and managements, it is vital that response personnel get informed in a timely manner as to where and how severely a building is damaged so that rescue efforts can be the most effective. However, challenges remain despite a great deal of effort in the field of image classification for disaster response. In this work, a promising approach based on deep learning is proposed to detect damaged buildings on high-resolution satellite imagery. By utilizing generic data augmentation, the proposed approach overcame the problem of limited training data popular in many remote sensing applications. A fine-tuning strategy is proposed to use transfer learning with a pretrained model for the task of interest. The experiments with imagery of Port-au-Prince, Haiti demonstrated the proposed method is effective when training data is limited. The Convolutional Neural Network (CNN) model with augmented training data can achieve 83% accuracy in detecting damaged buildings, greatly improved from 53% with the original training data. Future work would be focused on exploring automated methods to acquire larger training datasets and model generalization by investigating more robust data augmentation techniques.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor and committee chair Dr. Caixia Wang for all her hard work advising me through undergraduate and graduate school as well as advising me on my thesis. Without her advice, this thesis would not be possible. I would also like to thank my committee members Dr. Kenrick Mock and Dr. Tom Ravens for their support and feedback. I would also like to extend my sincere thanks to the Arctic Domain Awareness Center, in particular Executive Director Randy “Church” Kee and Education and Administration Manager Elizabeth “Ellee” Matthews, who supports me to pursue a Master of Science degree.

DEDICATION

This thesis is dedicated to my family without whose support this thesis would not be possible. To my mom and dad, Rebecca Mettert and Matthew Goers, to my stepparents, Jeffery Mettert and Carla Goers, to my siblings, Kyle and Schyler Goers, and to my grandmother, Melissa Goers.

Chapter 1

INTRODUCTION

Remote sensing in disaster management has been maturing since the use of cameras on monoplanes during the First World War (Cable, 2015). In recent decades, due to global warming and climate change, we have witnessed ever-growing occurrences of natural disasters such as flooding, tornadoes, earthquakes, and wildfires. According to the United States Geological Survey, between the years 2000 to 2012, there have been 807 US earthquakes above a magnitude 5.0 and worldwide there have been 23,608 earthquakes with an estimated 789,677 deaths (United States Geological Survey, n.d.). As such, it is more important than ever to provide emergency response personnel with accurate and timely information for their effective responses to crises. For example, Hurricane Harvey, which impacted Texas in 2017, caused an estimated $125 billion in damage while the 2010 Haiti earthquake caused between 220,000 and 250,000 deaths, according to the United Nations (Kolbe et al., 2010). With an estimated 250,000 residential and 30,000 commercial buildings collapsed or destroyed after the Haiti earthquake, remote sensing data was among the most cost effective and accurate sources of data because of its wide coverage, availability, and temporal frequency (E. Hussain et al., 2011). Emergency response is a race against time and when responders lack timely and accurate information, lives are lost (Turker & Sumer, 2008). A report detailing the medical response of the 1995 Kobe, Japan earthquake suggested that there was a drastic reduction in the number of people who survived after being rescued from collapsed buildings as time increased (Comfort, 1996). Information garnered from remote sensing data greatly assists response personnel in rescue and relief efforts and damage assessment efforts (E. Hussain et al., 2011). It is vital that response personnel get informed in a timely manner as to where and how severely a building is damaged so that rescue efforts can be the most effective.

Among the variety of information which can be derived from remote sensing, the locations of damaged buildings and their severity of damage have been demonstrated to be key information for saving lives. Many studies have used satellite or airborne images to detect damaged buildings. For example, Duarte et. al. (Duarte et al., 2018) conducted a study on the use of multi-resolution satellite and airborne imagery, both manned and unmanned. This study developed a convolutional neural network-based approach to detect and classify building damage while achieving an accuracy of between 89.8% and 94.4% (Duarte et al., 2018). Janalipour and Mohammadzadeh explored the use of QuickBird satellite imagery to detect damaged buildings after the Bam Earthquake in Iran with a four-stage approach. First, two preprocessing steps were implemented that included a vector map update, which used pre-event images, and a post-event image geo-referencing with the updated vector map (Janalipour & Mohammadzadeh, 2016). The second stage consisted of a pixel-based classification and segmentation on the post-event image, labeling the segments. The third stage derived the geometric properties such as area, convexity, and rectangular fitting for these segments (Janalipour & Mohammadzadeh, 2016). Finally, in the fourth stage, a decision-making system was developed based on the adaptive network-based fuzzy inference system, or ANFIS. The fuzzy interference system is a hybrid learning process, which integrates both neural network and fuzzy logic principles (Abraham, 2005). Apart from standard logic being limited in concepts that are completely true or completely false, fuzzy logic is more generalized to have a degree of truth between 0.0 and 1.0 to support inherently vague concepts (Elkan, 1999). Thus, the ANFIS model is more adaptive to estimate building damage levels that are inherently vague. A “sensitivity analysis” was proposed to optimize the ANFIS model characteristics generating a building damage map (Janalipour & Mohammadzadeh, 2016).

Furthermore, social media data has become useful in both developing situational awareness and in crisis response planning, as it has become popular and widely available in recent years, though the majority of these studies have been conducted on text-based posts (Alam et al., 2017). One study, investigating the relevant indicators of social media messages, discovered that on-topic social media messages with images are significantly closer to the disaster event than on-topic social messages without images (Peters & Porto De Albuquerque, 2015). Thus, disaster-relevant messages tend to be closer to the disaster event (Porto De Albuquerque et al., 2015), and the classification of damage levels of social media images is very relevant to situational awareness (Peters & Porto De Albuquerque, 2015). An example of this was the use of georeferenced social media images during the 2013 River Elbe flood in Germany. Social media users during a disaster, who are local to an event, are more likely to provide timely and direct data useful for improving situational awareness (Starbird et al., 2012). People in directly affected areas post about their survival related topics while people who are more remote post about "secondary effects" such as transportation issues and nuclear plant risks, illustrating concerns about indirect and future impacts from the disasters (Acar & Muraki, 2011). Albuquerque, et. al. found that relevant tweets related to the flood were 11 times more likely to occur near flood-affected catchment (Porto De Albuquerque et al., 2015). The usefulness of this type of data has been recognized by a number of organizations. For example, fire agencies in Australia are aware that social media imagery can provide significant time-sensitive data to empower them to make critical decisions. However, there are a number of barriers when utilizing social media data for disaster responses ((Lagerstrom et al., 2016),(Starr Roxanne Hiltz et al., 2014)). One study, by Hiltz, et. al. (Starr Roxanne Hiltz et al., 2014), discovered that after several interviews of emergency managers, there were three major barriers for the use of social media. The first major barrier is the lack of personnel and time to

work on the use of social media. A number of interviewees suggested that their emergency response teams are limited in size. With the limited number of personnel available, teams do not have the time or resources to monitor social media sites like Twitter (Starr Roxanne Hiltz et al., 2014). The second major barrier is the lack of policies and guidelines. Many emergency managers have stated that their government organization either does not have any standard operating procedures or that such procedures are outdated when it comes to resourcing social media. Many have also reported that social media accounts like Facebook and Twitter are blocked on their government computers (Starr Roxanne Hiltz et al., 2014). The third major challenge put forth by Hiltz et. al. is the trustworthiness of the data and lack of appropriate technology. One interviewee said that social media information "tends to be inaccurate" and "there is no way to control it" (Starr Roxanne Hiltz et al., 2014). Also, the large amount of social media posts made during a crisis event would be burdensome for processing (Starr Roxanne Hiltz et al., 2014).

Nevertheless, two general challenges remain despite a great deal of effort in the field of image classification for disaster response. One lies in the availability of data resources for analysis, and another is deriving accurate and actionable information. Aerial imagery has been extensively used in the detection of damages due to its high spatial resolution (Joshi et al., 2017), but the image swath can only cover a limited area. Additionally, flight operations may be costly. Satellites, on the other hand, can collect larger swaths of land in a single path. One crucial bottleneck for the acquisition of satellite data is the location of the satellite. If the satellite does not pass over the impacted area, it will have to be programmed to that location. If a satellite location causes large acquisition angles, this may cause image distortion, leading to poor georeferencing and limiting the users ability to co-register the imagery with other important data sets (Rathje & Adams, 2008). During a limited timeframe scenario, a compromise must be reached among response time,

analysis depth, and mapping accuracy (Voigt et al., 2011). High resolution data may be too massive to process for a large region (Rathje & Adams, 2008), often though, sub-meter imagery is used for assessment (Voigt et al., 2011). It is common practice, however, to quickly derive a preliminary damage assessment map with the initial data and then the initial estimates are refined with additional available data (Voigt et al., 2011). With a nadir, or the point at which a vertical line drawn from the sensor to the image plane it intersects (Wolf et al., 2014), view of the ground, satellite and aerial imagery can only capture the rooftops of buildings at a resolution that may be difficult to identify collapsed or partially collapsed roofs ((Rathje & Adams, 2008),(Maruyama et al., 2014)).

To overcome the limitation of the nadir view, many research studies makes use of social media images ((Alam et al., 2017), (Cervone et al., 2016),(Q. Huang & Xiao, 2015),(D. T. Nguyen et al., 2017)). Social media imagery offers emergency response personnel the ability to harness a multitude of on-the-ground data (Cervone et al., 2016). While social media posts are often published without scientific intent, it is still possible to offer important information (Cervone et al., 2016). Social media data can provide timely damage information, help in rescue and relief operations, and help improve situational awareness (Cervone et al., 2016). One of the largest challenges when dealing with social media data is the heterogeneous nature of the data, which overwhelms the capacity of the human analyst ((Cervone et al., 2016),(Q. Huang & Xiao, 2015)).

Because of the timely need for damage information immediately after a natural disaster, the initial analysis of remote sensing data may be compromised, and more time will be needed to make a more in-depth analysis (Rathje & Adams, 2008). Rathje and Adams proposed a multi-level approach in which lower resolution data, which have faster processing times, at a larger scale, are used first to develop rapid observations of damage over a large area. Then, higher resolution data

is used to refine and improve the initial damage estimates. Additionally, they propose that observations from a number of resolutions can lead to a more comprehensive assessment (Rathje & Adams, 2008). Nguyen et.al. (D. T. Nguyen et al., 2017) tried to tackle the issue by using data from previous disasters to train the algorithm to predict damage levels on images of the current disaster. Their machine learning-based approach analyzed imagery from social media feeds from four separate natural disaster events using a convolutional neural network algorithm with domain-specific fine-tuning to classify images. However, two main challenges remain that need improvement for the future: the low prevalence of training data, and the labeling of that data by humans, which consumes valuable time (D. T. Nguyen et al., 2017). Alam et. al. (Alam et al., 2018) proposed to combine human and machine computing to process social media images. They introduced an image processing pipeline where the images are collected from Twitters API in real-time. In the work, two image filters are implemented to reduce noise in the dataset. First, a relevancy filter which discards images that are not relevant to their study and second, a de-duplication filter which discards duplicate or near-duplicate images. One shortcoming discovered from this pipeline approach was the large volume of social media data to be processed and labeled. Deficiencies in labeled data can cause problems in the subsequent automatic classification procedures. The study recruited many volunteers to label images for purposes of quality data labeling. These images then passed through a classification module, along with un-labeled images, to classify the image into classes (Alam et al., 2018).

In the following chapter current state-of-the-art techniques and challenges in detecting damaged buildings will be reviewed and summarized. The purpose of this work is to develop a timely system for the classification of damaged infrastructure after a natural disaster given limited training data. Chapter 2 will review the most current literature in image classification taking a

focus on post-natural disaster imagery. Chapter 3 will discuss the residual network convolutional neural network, common network accuracy assessment found in literature, and model fine-tuning. Chapter 4 discusses the experimental study including data, implementation, results, and model generalization. Finally, Chapter 5 will conclude the thesis with an overview of future work.

Chapter 2

LITERATURE REVIEW

As discussed in the previous section, two main challenges remain concerning image classification for damage assessment: the availability of data resources and the timely gathering of accurate and actionable information. In this section, prevailing work addressing these challenges was reviewed, ranging from the fundamental pixel-based approaches to the advanced deep learning methods. Discussions about further work development has also been included.

Pixel-based techniques rely on the spectral characteristics of surface features of multiple or hyperspectral images. A pixel is the most basic unit of image analysis. In a pixel-based approach, pixels are exploited mostly without considering their spatial context. Therefore statistical operators are used to evaluate a single pixel in an image consisting of possibly thousands or millions of pixels (M. Hussain et al., 2013). Generally, pixel-based classification can be segregated into two categories, unsupervised classification and supervised classification (Li et al., 2014). Using unsupervised classification on an image, pixels are divided into a number of classes based on the pixels' natural grouping. This technique is accomplished without the assistance of any training data or prior knowledge of the study area (Li et al., 2014). Two of the most common algorithms are the k-means (and its variants) and the Iterative Self-Organizing Data Analysis (ISODATA) technique ((Li et al., 2014), (Ouyang et al., 2011)). K-means clustering follows three general steps as described by Zhao et. al. (Zhao et al., 2018). The first starts by initializing k cluster centroids (classes), then assigning each sample (pixel) to its closest centroid, and finally recomputing the centroids based off the assignments in step two. ISODATA is an iterative technique that uses Euclidean distance to measure the similarity of samples to their classes ((Zhao et al., 2018), (Rollet et al., 1998)). The purpose of ISODATA is to assign the pixel spectral pattern

vector into one of several classes. The k-means algorithm is similar, but with additional heuristics that determine the splitting or merging of clusters. If a cluster has a variance that is greater than a pre-selected threshold, the algorithm will split the cluster. Otherwise, it will merge the clusters ((Dhodhi et al., 1999), (Wang et al., 2014)). Supervised classification requires human supervision. An image analyst must select sample sites on an image that is representative of known, human defined classes. The spectral properties of each pixel in the image are then compared to the human-selected training data (Li et al., 2014). The Maximum Likelihood Classifier (MLC) outperforms other supervised classifications algorithms (Sisodia et al., 2014). The classifier generates a probability density function for each class and during the classification phase all pixels are assigned to a class based on the likelihood of that pixel occurring within each probability density function (Sun et al., 2013). Pixel-based approaches are less time-consuming compared to object-based classification (Duro et al., 2012). However, their classification exhibits the so-called "salt-and-pepper" effect ((Ouyang et al., 2011), (Estoque et al., 2015)). This effect can be described as erroneously classified pixels in an area of correctly classified pixels. Additionally, pixel-based approaches only consider spectral information while neglecting textual and contextual information (Qian et al., 2007).

In contrast, object-based image classification does not directly operate on individual pixels but on homogeneous and spatially contiguous groups of pixels, or objects (Qian et al., 2007). Pixel grouping is achieved by an image segmentation process which can be classified into three major image segmentation areas according to Haralick and Shapiro (Haralick & Shapiro, 1985): spatial clustering, thresholding, and region growing. As the accuracy of the image segmentation has a direct influence on the image classification (Qian et al., 2007), the best practice to evaluate the performance of segmentation is for the human analyst to interpret the results ((Estoque et al.,

2015), (Pal & Pal, 1993)). After affirming the segmentation results, the next step is classification of the generated objects. Improved classification accuracy can be obtained by employing a number of different image characteristics such as context and texture, and not solely relying on a series of pixel spectra. Image context, as described by Blaschke (Blaschke, 2004), relies on a summary description of the relationships between multiple pixels or multiple classes, while image texture describes the spatial variation of pixel values. Alongside texture, other information can be analyzed from the immediate surrounding neighborhood of pixels, with result being assigned to a singular, central pixel such as spectral information. The K-Nearest Neighbor (KNN) classifier is one of the most popular object-based classifiers, although it can be utilized for pixel-based classification. The KNN algorithm classifies objects based on the nearest training samples (Cover & Hart, 1967). An object is classified based on the labels of its k nearest neighbors by a majority vote. When the majority votes result in a tie, meaning that the object could be in either of the two classes, the algorithm applies a Euclidean distance calculation. KNN performs well with multi-modal classes. However, it uses all the features equally in computing for similarities leading to classification errors (Kim et al., 2012). A great deal of work has demonstrated that object-based classifiers perform better than pixel-based ones. Platt and Rapoza (Platt & Rapoza, 2008) compared the MLC and KNN algorithms using models that both did and did not incorporate expert knowledge. They showed that the object-based model using both the KNN algorithm and expert knowledge gave the best accuracy results of 78%, while the best pixel-based approach only achieved an accuracy of 64%. Myint et. al. (Myint et al., 2011) compared MLC with KNN using QuickBird imagery to classify urban land cover. They achieved a 90.4% accuracy with their KNN classifier and a 63.3% accuracy with their pixel-based MLC classifier on their original image. They also tested using the same classification strategies on another image with different environmental variables. In this case,

the pixel-based classifier had a much-improved classification accuracy of 87.8%. However, the object-based approach had an accuracy of 95.2%, still better for the classification of urban environments. Yan et. al. (Yan et al., 2006) compared pixel-based and object-based approaches to detect and map surface coal fires to design firefighting plans. Clearly, the object-based approach was found to be superior to that of the pixel-based approach, having an accuracy of 83.25% and 46.48% respectively. Nevertheless, the image segmentation process (Figure 1 (Blaschke, 2004)) is critical for the performance of object-based algorithms and requires proper calibration of segmentation parameters (Estoque et al., 2015). This type of calibration analysis is more time consuming than the pixel-based approach. Sometimes it requires the analyst to go through the trial-and-error process (Duro et al., 2012). In addition, not all object features should be used in classification because information extracted from the segmented image objects usually have a high dimension and are usually highly correlated (Liu & Xia, 2010).

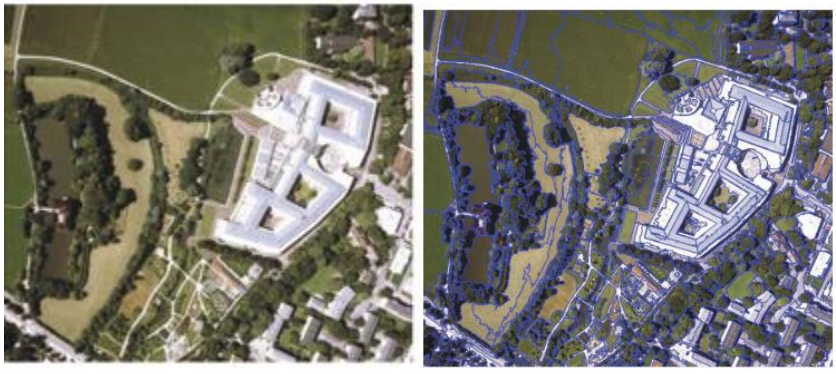

Figure 1. Image segmentation (right) of a small subset of an orthophoto (left) of the city of Salzburg, Austria.

With the significant availability of images with higher spatial resolution, image classification becomes more complex, and thus, requires more versatile and robust classification techniques (M. Hussain et al., 2013). The pixel-based and object-based classification approaches rely on relatively simplistic approaches to describe and classify pixels and image objects. In recent years, because of the advancement of high-performance computation in both hardware and software, machine learning for image classification has enjoyed renewed popularity.

Machine learning is a subdivision of artificial intelligence and deals with the construction of algorithms to learn from data. A variety of algorithms have been developed ranging from Support Vector Machines, to Random Forests, to Artificial Neural Networks. Machine learning can be utilized for both unsupervised or supervised classification and regression of nonlinear systems (Figure 2 (Apruzzese et al., 2018)). These nonlinear systems can involve many data records with thousands of variables or attributes. From a set of data records, or data set, a training data set can be extracted to represent the entire dataset (Lary et al., 2016). In unsupervised machine learning classification, the classifier attempts to identify hidden patterns in unlabeled data by finding similarities between the records of a data set. In supervised machine learning classification, a mathematical model is trained to map input records into a particular known class (Maulik & Chakraborty, 2017). Following model training is a validation process which tests the model with a separate validation data set. The validation data set is used to evaluate the trained model and fine-tune the models' parameters. The test data set is used to evaluate the final model once it is trained and fine-tuned (Shah, 2017). Of the many machine learning algorithms, Support Vector Machine, Random Forest, and Convolutional Neural Networks are among the most commonly used in damage classification ((Taskin Kaya et al., 2011), (Cooner et al., 2016), (D. T. Nguyen et al., 2017)).

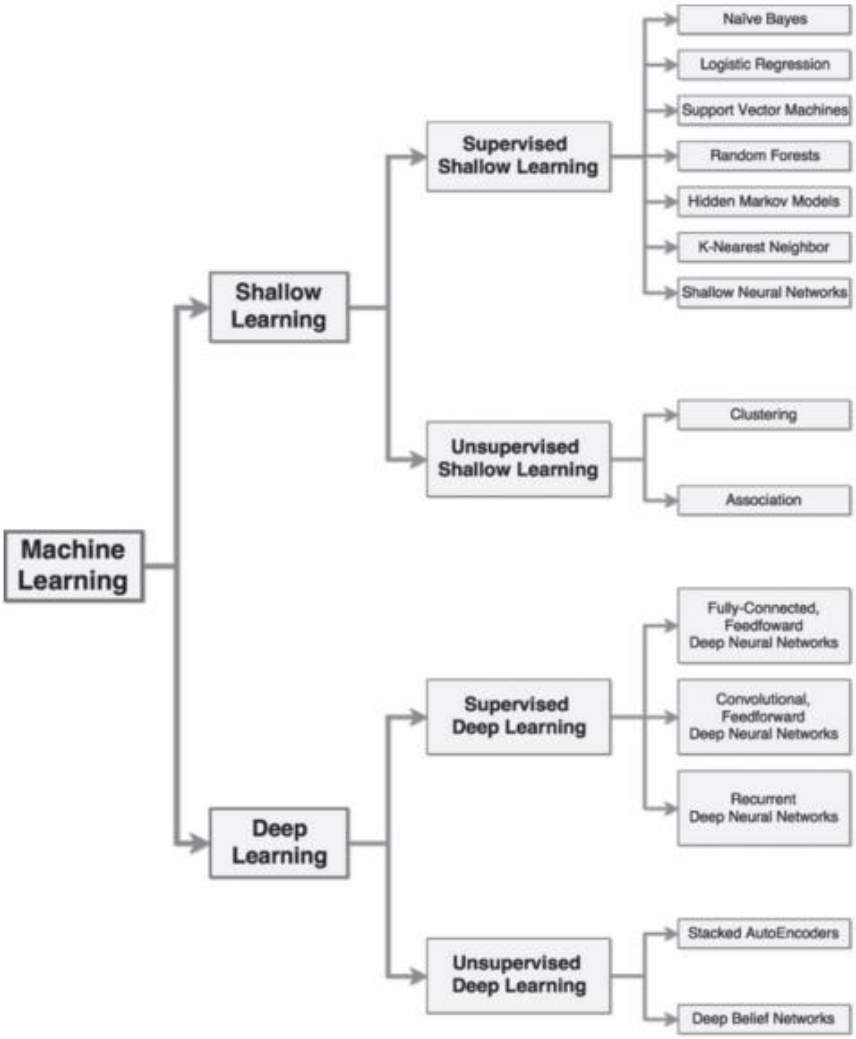

Figure 2. Taxonomy of machine learning techniques.

The Support Vector Machine (SVM) algorithm, in its original form, is an algorithm for two-class classification, meaning that the algorithm, in its infancy, could only classify data into two distinct classes. It maps the input vectors into a high dimensional feature space. In this feature space, a hyperplane is created with properties that ensure a high generalization ability of the network (Cortes & Vapnik, 1995). The optimal separating hyperplane discriminates the data set

into discrete number of classes that minimizes misclassification attained during the training phase (Figure 3 (Kavzoglu & Colkesen, 2009)) (Maulik & Chakraborty, 2017). The implementation of a linear SVM assumes that the feature data are linearly separable. However, in practice, data points of different classes often overlap with each other making the basic linear decision boundaries insufficient. Kernel functions have thus been developed (Mountrakis et al., 2011). Non-linear kernels, like the radial-basis function, map the input to a higher spatial dimensional feature space where a linear decision boundary, via hyperplane, separates the classes. SVMs have the ability to handle small training data sets while often producing higher classification accuracy than traditional methods (Mantero et al., 2005). In addition, the SVM learning process follows a structural risk minimization scheme that minimizes classification error on unseen data without any previous assumptions made on the probability distribution of the data. Another benefit of the SVM model is its ability to strike a balance between accuracy on a limited amount of training data and the ability to generalize unseen data (Mountrakis et al., 2011). Essentially, this means the SVM can strike a reasonable balance between bias and variance. If the classifier has too many adjustable parameters (bias), it will likely learn the training data without difficulty, but it is unlikely that it will generalize properly for the unseen data patterns (Guyon et al., 1992). One challenge of the SVM model is the choice of an optimal kernel. Although there are many kernel functions, some are not optimal for certain remote sensing applications (Mountrakis et al., 2011). Research has shown that kernels such as the radial basis function and polynomial kernels produce different results when applied to satellite imagery (Zhu & Blumberg, 2002).

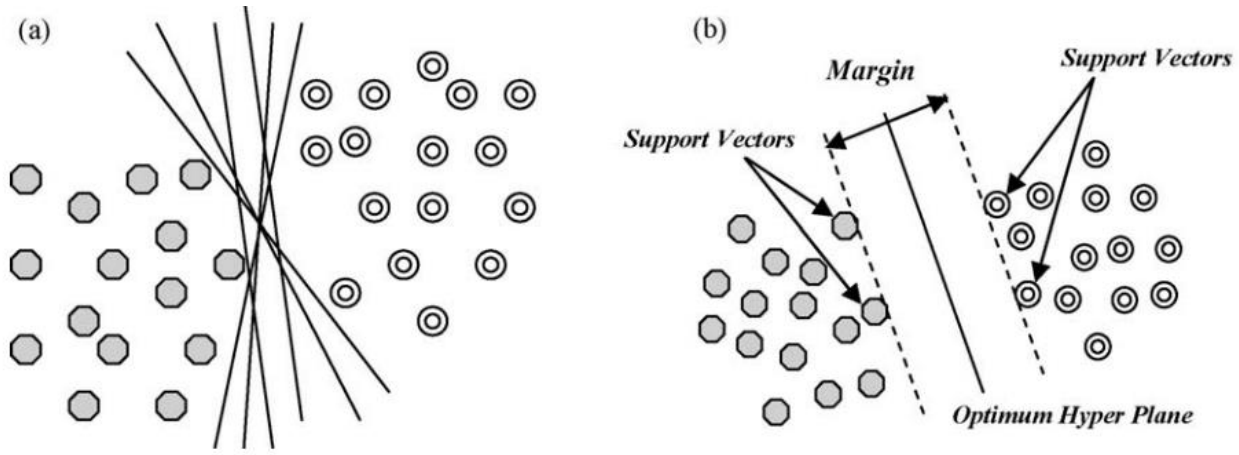

Figure 3. Possible hyperplanes for linearly separating data (a) and the optimum hyperplane and support vectors (b).

Anniballe et. al. (Anniballe et al., 2018) assessed earthquake damage of very high resolution satellite imagery using the SVM method compared to the Bayesian Maximum A Posteriori method. The results from the two machine learning algorithms were compared to a ground survey. Among the features extracted from each building footprint image, textural and color features were found to be the most effective metric used for damage classification. The SVM algorithm slightly outperformed the Bayesian classifier with the accuracies of 96.8% and 96.2% respectively. Kaya et. al. (Taskin Kaya et al., 2011) proposes the novel "support vector selection and adaptation (SVSA)" approach. The SVSA method is a supervised classification method for the classification of both linearly and nonlinearly separable data. This method consists of the selection and the adaptation stages. In the selection stage, the linear SVM is used to obtain support vectors from the training data set. These support vectors are then classified using the KNN algorithm. If the original label and the predicted label are different, then that support vector is

excluded, and the remaining vectors are called “reference vectors.” In the adaptation stage, the reference vectors are iteratively adapted to make them more representative of the training data by the nearest neighbor rule. In order to adapt the reference vectors, a random sample is selected from the training data and then the distance between this sample and the reference vectors are calculated using Euclidian distance. If the class labels are the same, the reference vector is moved toward the selected sample. Figure 4 below shows the learning flowchart proposed by Kaya et. al. (Taskin Kaya et al., 2011). Using this novel approach to classify damage, an 81.4% accuracy was achieved while only using post-earthquake imagery.

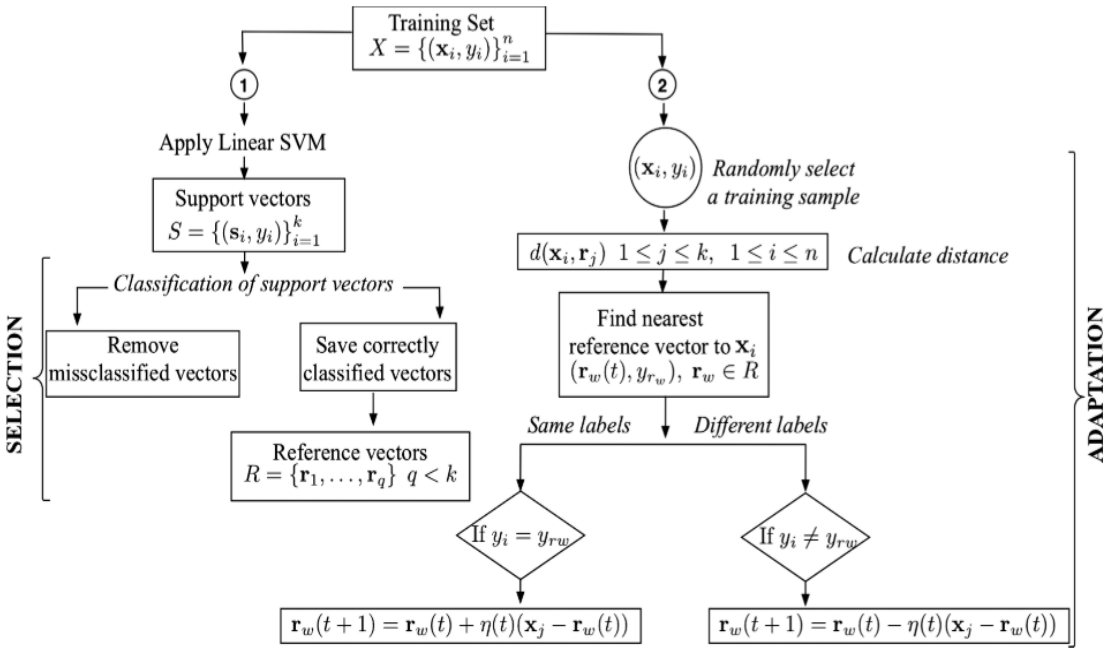

Figure 4. The learning flowchart of the SVSA algorithm, 1 and 2 refer to the selection and adaptation process, respectively.

When classifying damaged buildings after a natural disaster, the Random Forest (RF) algorithm has also been used. It is a machine learning approach that uses many decision trees (DT) (Figure 5 (Donges, 2018)) in order to overcome the disadvantages of a single DT. The RF approach generalizes multiple DT's independently to identify a feature to achieve a more accurate and stable prediction ((Tin Kam Ho, 1995), (Donges, 2018)). A single record can be analyzed through many decision trees until a classification is made. Because there are many decision trees through which a single record is processed, a simple majority vote is taken leading to a final classification (Maxwell et al., 2018). In a single DT, a record starts at the root node traveling to decision nodes, finally ending at a leaf. At each node, a decision is made based on the variables in each record or on a predefined set of splitting rules. Once complete, the DT will assign a single label to the record, thus classifying it (Shalev-Shartz & Ben-David, 2014). A single DT may not be optimal, but by using many trees a more optimal solution may be reached. Each tree is trained on its own random subset of the training data, also only using a random subset of variables. Using only a subset of the training data and a subset of variables, each tree will be less accurate. However, it will also be less correlated thus making the whole RF more reliable (Maxwell et al., 2018). One of the advantages of a RF is that it reduces the danger of overfitting (Shalev-Shartz & Ben-David, 2014). Additionally, benefits of a RF include robustness to outliers and noise and built-in estimates of error and variable importance (Breiman, 2001). One problem of the RF is that by having many random trees, possibly hundreds, the ability to visualize the trees are effectively lost ((Maxwell et al., 2018), (Breiman, 2001)). Moreover, the RF algorithm can become time consuming based on the number and size of the trees estimated by $M(m*n*\log(n))$, where M is the number of trees, m is the total number of variables, and n is the number of samples (Cooner et al., 2016). This means that the more trees or more variables or more samples that exist, the Random Forest runtime will

increase according to the equation shown. Cooner et al. (Cooner et al., 2016) used the RF algorithm, among others, to detect urban damage after the 2010 Haiti earthquake. The RF algorithm was compared to a developed Artificial Neural Network (DANN), which consisted of two hidden layers of 20 neurons with sigmoid activations and a binary softmax output layer, and the radial basis function neural network (RBFNN). After testing each of these algorithms, they showed that the DANN outperformed the RBFNN and RF models. The DANN had a building class omission error rate of 37.7% whereas the RF model had an omission error rate of 71.27%, where omission error rate refers to the portion of the incorrectly classified reference data as compared to the total reference data. The RF had the highest overall accuracy of 76.1%, but it severely underestimated the damage, while also having the longest runtime of 144.9 minutes. While the DANN algorithm outperformed the RF in a building-by-building assessment and required less training time (10.4 minutes), it required a larger number of training samples than the RF algorithm.

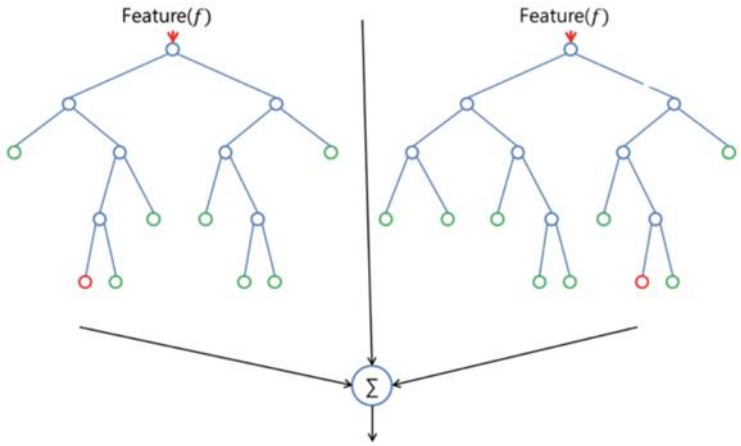

Figure 5. A schematic representation of a Random Forest.

The Artificial Neural Network (ANN) is a computational model inspired by the structure of neural networks in the brain (Shalev-Shartz & Ben-David, 2014). A standard neural network consists of many simple, connected neurons, each of which produces a sequence of activations. These neurons are activated through sensors perceiving the environment through weighted connections from previously activated neurons. The first and simplest type of network was the feedforward neural network (Schmidhuber, 2015). In this type of network, information moves from the input nodes through the hidden nodes and then out the output nodes. There are no loops in this type of network, hence the name feedforward. A series of linked nodes can perform complex calculations (Shalev-Shartz & Ben-David, 2014). A subclass of feedforward neural network, particularly used in image analysis and geoscience problems, is the Convolutional Neural Network, or "CNN" ((Lary et al., 2016), (Valueva et al., 2020), (Matsugu et al., 2003)).

CNNs are emerging as a promising technique in many object recognition applications. CNNs combine three architectural ideas including local receptive fields, shared weights, and spatial or temporal subsampling (LeCun & Bengio, 1995). CNNs have multi-layered interconnected channels with a high capacity for learning the features and classifiers. They also have the capacity to adjust parameters jointly and classify simultaneously. Additionally, with this type of ANN, it has the potential to encode both spectral and spatial information into the classification automatically ((Nogueira et al., 2017), (Mahdianpari et al., 2018)). A neuron takes a vector input and performs an operation which produces an output. The most common function is

$$a = \sigma(wx + b) \qquad \text{(Eq. 1)}$$

where w is a weight vector, b is the bias, and σ is the activation function.

The weight vectors and the biases are the parameters that define the function. The goal of training the network is to find the optimal values for these parameters. Among the most common activation functions is the rectified linear unit (ReLU) (Maggiori et al., 2017). The ReLU takes the linear weighted sum of its input then outputs a non-linear function of the total input (Hinkelmann, n.d.). It is common to organize a CNNs neurons in sets of stacked layers that transforms the outputs of the previous layer into the inputs of the next layer to allow the network to learn hierarchical features. Finding the optimal network comes down to finding the weights and biases that minimize loss between the predicted values and the target values. Once the loss function has been defined, the weights and biases can be solved for, minimizing that loss. Computing the derivative $(\partial L)/(\partial w_i)$ of the loss function is calculated through backpropagation. Backpropagation consists of computing derivatives of the loss with respect to the last layers parameters by using the chain rule to compute, recursively, the rest of the derivatives. To improve training, CNNs have a special feature, that is every neuron has a spatial coordinate associated with it. With this feature, there are a couple of constraints: the connections are only extended to a limited spatial area determined by kernel size and the same filter is applied to each location. Multiple convolution kernels are learned in every layer and the response to every filter is known as the "feature map." By enforcing the constraints, a CNN architecture can include a convolutional layer instead of a fully connected layer. The convolutional layer reduces the number of parameters which results in a regularized loss function to allow easier optimization (Maggiori et al., 2017).

Some CNN architectures include downsampling, the reduction in the resolution of the feature maps (Maggiori et al., 2017). The goal of this process is to increase the receptive field, or the small region which is input into the neuron (Mahdianpari et al., 2018). For the predictions of a CNN to consider a larger spatial area, the higher-level reasoning layers should have a large receptive field. To accomplish this, the convolutional kernel size needs to be increased, resulting in the increase of the number of parameters and memory usage. Alternatively, downsampling to a lower resolution is suggested by introducing pooling layers. The pooling layer (Figure 6 (Dertat, 2017)) simply takes the average or the maximum value within a stride, or filter size. CNNs usually contain a fully connected layer on top of the convolutions/pooling layers. This layer is designed to have many outputs as labels and produces the final classification (Maggiori et al., 2017).

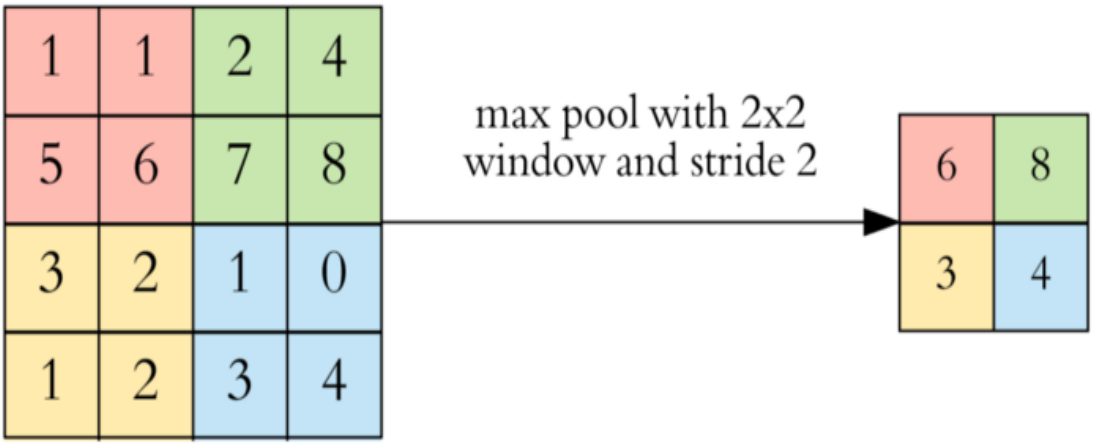

Figure 6. An example of a pooling layer with a max pool of 2x2 window and a stride of 2, used for downsampling.

Nguyen et. al. (D. T. Nguyen et al., 2017) shows that the fine-tuning of CNNs outperforms other techniques when it comes to assessing damage from social media imagery data after

disasters. It is common to use a pre-trained CNN as a feature extractor on a new dataset. One popular approach is to use the existing weights of a CNN as an initialization, which is referred as fine-tuning. In this approach, the last layer of the CNN is adapted to suit the task at hand and the pre-trained network is fine-tuned according to the training images from the new dataset. Nguyen et. al. (D. T. Nguyen et al., 2017) applied the commonly used cross-entropy loss function and fine-tuned the VGG-16 CNN model to classify images as severe, mild, and little-to-no damage. The fine-tuned model outperformed the Bag of Visual Words and the not-fine-tuned CNN model in every respect and in every test case (D. T. Nguyen et al., 2017). Alam et. al. (Alam et al., 2017) used a CNN to filter social media images to determine relevancy using the VGG-16 model. A pre-trained VGG-16 was also used for image classification. This model was fine-tuned using social media images and by adapting the last layer of the CNN to comply with the number of classes selected by the user.

Duarte et. al. (Duarte et al., 2018) compared three CNN configurations, trained with multi-resolution image samples against two benchmark networks where only satellite images are used. One benchmark network is trained from scratch while the other is fine-tuned on generic satellite image samples. All three multi-resolution test networks have been created to best combine features from each image resolution level. All the networks use residual connections and dilated convolutions. Because of the hierarchical stacking of convolutional layers that allow the network to learn from lower-level features to high level abstractions, a given layer may need information not only from the immediately previous layer, but also from other previous layers. Residual connections enable this process to occur, effectively allowing every level of a residual network to contribute to the final recognition task. Another component of these networks is dilated convolutions, which allows the network to capture spatial context of multi-resolution images. It is

proven that the neural networks that use multi-resolution imagery outperform the two benchmark networks by 4%.

Chapter 3

DEEP CNN FOR DAMAGE DETECTION

With the discussion in Chapter 2, fine-tuning CNN models which were fitted to computer vision datasets outperform other techniques for damage detection using high-resolution satellite images. This refinement of models can be seen as transferring learned skills to apply them in a different context, also known as transfer learning. Therefore, this work is developed to fine-tune the ResNet152 model, the most widely used backbone architecture in computer vision related designs, to classify damaged and non-damaged buildings on satellite imagery. This model was compared to other ResNet models of varying depths including ResNet34 and ResNet50. ResNet152 was chosen because it gave the best testing accuracy when tested on the same sample set. Additionally, higher-depth ResNet models do not have lower training accuracies than their shallower counterparts by design ((He et al., 2016a), (He et al., 2016b)). The proposed method consists of four primary steps, as shown in Figure 7. It starts with preparing the data by extracting training data for the task of interest (i.e., building footprints and manually labeling them as "Damaged" or "No Damage") after the satellite imagery has been pre-processed (atmospheric correction and orthorectification). This data is then separated into three categories: training, validation, and testing. Model fine-tuning follows, which modifies the classification layers of the existing architecture and trains the filter weights of the neural network. Finally, output accuracy from the fine-tuned classifier will be assessed using metrics widely available in literature.

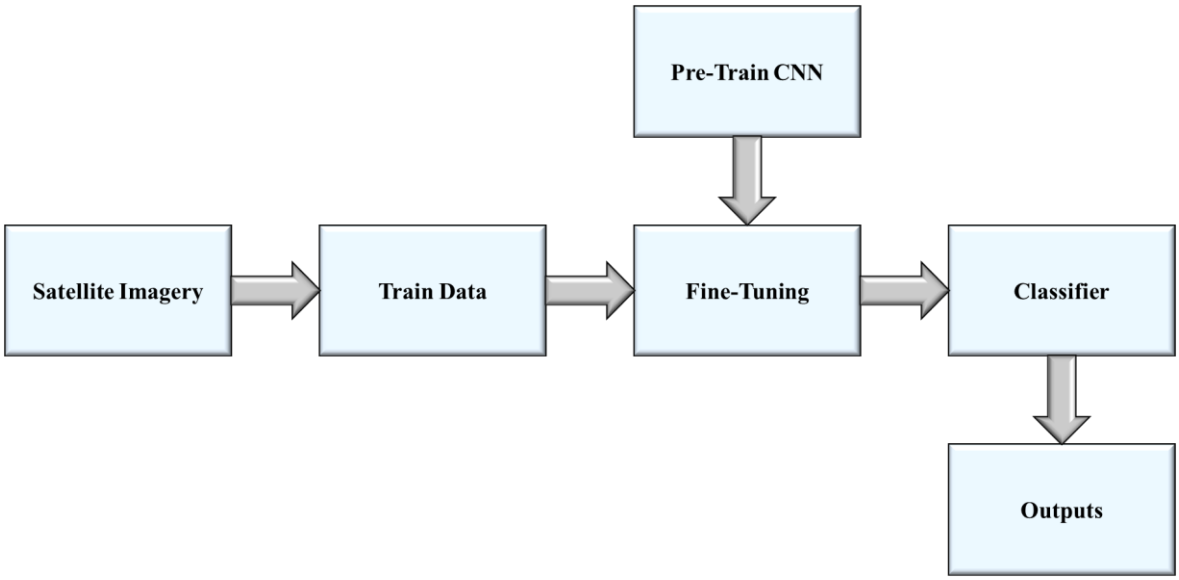

Figure 7. Overview of the proposed methodology.

In this chapter, the convolutional backbone, ResNet152, will be introduced and discussed first. Its overall architecture will be briefly discussed in section 3.2. In section 3.3, the introduction of the metrics used for accuracy assessment, which are widely used in literature, are discussed. Finally, section 3.4 will focus on the model fine-tuning scheme.

3.1 Residual network

The residual network (ResNet) was introduced by He et. al. (He et al., 2016b) to solve a problem of training deep neural nets. Research suggests that neural network depth is of high importance when training models (Simonyan & Zisserman, 2014). However, when dealing with high-depth networks, a number of issues can arise in the learning process including the problem of vanishing or exploding gradients which curtail convergence ((He et al., 2016b),(G. Huang et al., 2016)). This problem has been addressed by normalized initialization and intermediate

normalization layers. However, when a deep network starts to converge, a degradation problem emerges where accuracy gets saturated and degrades quickly. Degradation occurs when a model gets deeper, and it becomes more difficult for layers to propagate information from the shallower layers causing information loss. This degradation problem is not caused by overfitting the data. Instead, adding additional layers to an already deep model leads to higher training error (He et al., 2016b). ResNet tackled the problem by introducing a so-called "identity shortcut connection," and a heavy use of batch normalization.

ResNet is constructed from residual building blocks to address the degradation problem. Instead of skipping a few stacked network layers and fitting an underlying mapping, this model fits a residual mapping. Let's use *H(x)* to represent the output mapping function:

$$H(x) = F(x) + x \qquad \text{(Eq. 2)}$$

Where *x* is the identity function (input=output) and *F(x)* represents the residual function (He et al., 2016b). Consider two consecutive layers as shown in Figure 8 below (He et al., 2016b).

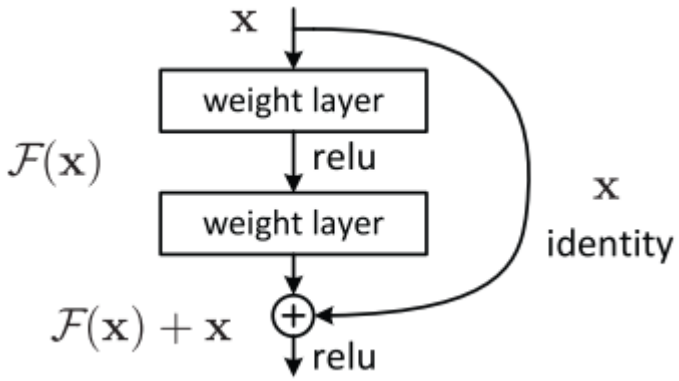

Figure 8. Residual learning: a building block of convolution layers.

H(x) can be considered an underlying mapping to be fit by a few of the stacked layers and *x* representing the input to the first of these layers. If we agree that multiple nonlinear layers can approximate complex functions, then equivalently, it should hold that multiple nonlinear layers can approximate a residual function. Because of this, He et. al. (He et al., 2016b) hypothesize that it is easier to learn the residual function *F*. If the added identity mapping layers of a deeper model are constructed, they should have a training error no larger than the model's shallower counterpart. With a residual learning system, if identity mappings are optimal, solvers may drive the layer weights towards zero to come close to identity mappings (He et al., 2016b).

Identity mappings are vital parts of the residual learning block. In a ResNet model the core idea is to learn the additive residual function F with respect to $h(x_l)$ with a key choice of identity mapping $h(x_l) = x_l$. This is achieved by attaching the identity skip connection (or shortcut) to the ResNet model. He et. al. (He et al., 2016a) expresses the building blocks, or "Residual Units," and identity mappings with the following equations where x_l and x_{l+1} represent the input and output of the lth unit respectively, F is a residual function to be learned, $h(x_l) = x_l$ representing an identity mapping, f is a rectified linear unit (ReLU) activation function, W_l is a set of weights and biases associated with the lth Residual Unit, and *K* representing the number of layers (e.g. 2, 3, 4…).

$$y_l = F(x_l, W_l) + h(x_l) \qquad \text{(Eq. 3)}$$

$$x_{l+1} = f(y_l) \quad \text{(Eq. 4)}$$

$$W_l = \{W_{l,k} | 1 \leq k \leq K\} \quad \text{(Eq. 5)}$$

If both $h(x_l)$ and $f(y_l)$ are identity mappings, the "signal could be directly propagated from one unit to another in both the forward and backward passes" (He et al., 2016a). The shortcut connections do not introduce extra parameters to the equation, nor do they introduce additional computational complexity. The dimensions of h and F must be equal and if they are not, a linear projection, W_s, must be calculated in the shortcut equation as follows (He et al., 2016b):

$$y_l = F(x_l, W_l) + W_s h(x_l) \quad \text{(Eq. 6)}$$

This projection shortcut is done by 1x1 convolutions. Another option is to zero-pad entries to increase dimension so that the input and output of a layer have the same dimensions (He et al., 2016b).

Training a deep ResNet CNN can be done with Stochastic Gradient Descent (SGD) with back propagation. Deep neural networks come with a set of issues including vanishing gradients in backward propagation and long training times. As gradient information is back-propagated, repeated convolutions with small weights renders the gradient information ineffectively small in earlier layers (G. Huang et al., 2016). Although many attempts have been made at improving the training deep neural networks, a notable contribution was made with the creation of Batch

Normalization (BN) (Ioffe & Szegedy, 2015). BN standardizes the mean and variance in hidden layers thereby reducing the vanishing gradient problem (G. Huang et al., 2016). Learning with stochastic depth is accomplished in a similar fashion as ResNets. Skip connections are introduced and for each mini batch of training data, a set of layers is randomly selected, and their transform function are removed keeping only their identity skip connections.

3.2 Basic Architecture of ResNet152

The ResNet152 model starts with a 128x128x3 input image to be processed into a 7x7 convolution layer feeding into a 3x3 max pooling layer with both having a stride of 2, then comes a 3-layer residual building block, the design of which is a bottleneck. The first layer of the building block is a 1x1 convolution, the second layer is a 3x3 convolution, and the third layer is another 1x1 convolution. The 1x1 convolution can reduce the number of parameters while not degrading the performance of the network as much (Tsang, 2018). As shown in Figure 9 below, the residual shortcut passes the 3-layer building block (Tsang, 2018). 50 of the residual learning building blocks are used using a total of 150 layers.

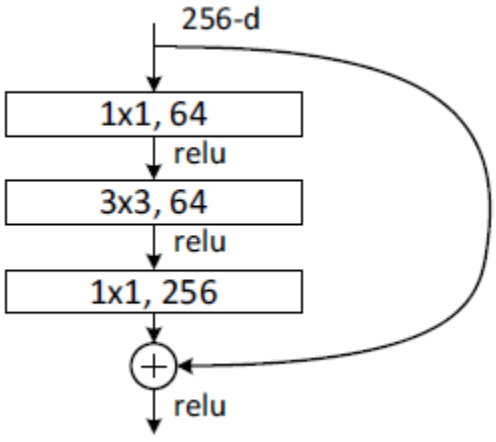

Figure 9. ResNet152 building block.

The layers have the same number of filters for the same output feature size. However, if the output feature size is halved, the number of features is quadrupled. Downsampling is performed directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer with a 1000-way fully connected layer with softmax with an output size of 1x1. The 152-layer network architecture is shown in Figure 10 below (L. D. Nguyen et al., 2018).

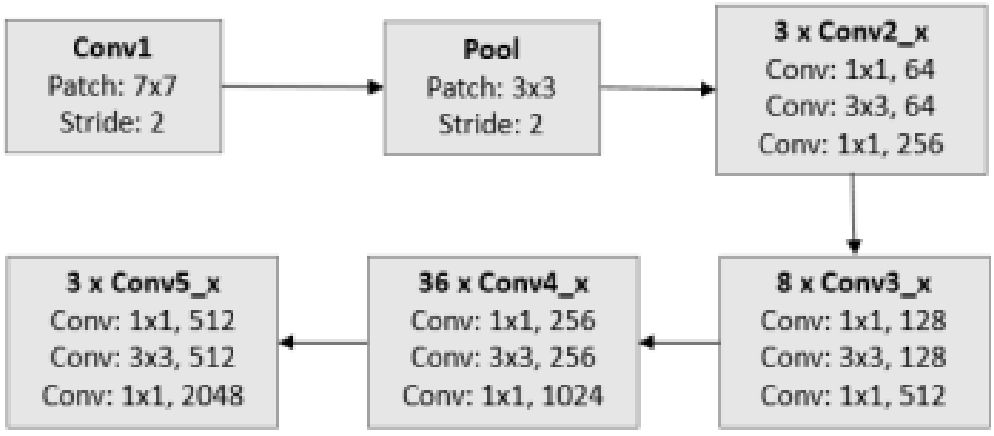

Figure 10. The basic architecture of ResNet152

3.3 Accuracy Assessment

A number of assessments were conducted to assess the accuracy of the ResNet152 model classifying damaged and non-damaged buildings including accuracy, precision, sensitivity, specificity, F1 score, Cohen's Kappa, and the Matthews Correlation Coefficient (MCC).

Accuracy, or the percentage of correctly classified buildings, can be calculated as (Shin, 2020):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{(Eq. 7)}$$

Where TP refers to the true positive outcome of a confusion matrix, TN as the true negative outcome, FP as the false positive, and FN as the false negative outcome. Positive results are referred to as the "Damage" classification and negative results are referred to as the "No Damage" classification.

Precision, or the proportion of Damage or No Damage predictions that are correctly predicted, can be calculated as (Shin, 2020):

$$Precision\ (Damage) = \frac{TP}{TP + FP} \quad \text{(Eq. 8)}$$

$$Precision\ (No\ Damage) = \frac{TN}{TN + FN} \quad \text{(Eq. 9)}$$

Sensitivity, or true positive rate, is the proportion of actual positives ("Damage") identified correctly. Specificity (or the true negative rate) measure the proportion of negatives ("No

Damage") that are correctly identified as negatives (Shin, 2020). Sensitivity and specificity, also known as recall, can be calculated as follows:

$$\text{Recall (Damage)} = \frac{TP}{TP + FN} \quad \text{(Eq. 10)}$$

$$\text{Recall (No Damage)} = \frac{TN}{TN + FP} \quad \text{(Eq. 11)}$$

The F1 Score is a measure of a test's accuracy, the harmonic mean of precision and sensitivity. It measures the robustness of the model having a maximum score of 1, meaning perfect precision and recall, and having a minimum score of 0 (Shin, 2020).

$$F1\ Score = \frac{2 * (Precision * Sensitivity)}{Precision + Sensitivity} = \frac{2 * TP}{(2 * TP) + FP + FN} \quad \text{(Eq. 12)}$$

Cohen's Kappa Score is a measure of inter-rater agreement, entailing how much the prediction and the ground truth labels agree (Sim & Wright, 2005).

$$p_o = \left(\frac{TP + TN}{TP + FN + TN + FP}\right) \quad \text{(Eq. 13)}$$

$$p_{Yes} = \left(\frac{TP + FN}{TP + FN + TN + FP} * \frac{TP + FP}{TP + FN + TN + FP}\right) \quad \text{(Eq. 14)}$$

$$p_{No} = \left(\frac{FP + TN}{TP + FN + TN + FP} * \frac{FN + TN}{TP + FN + TN + FP}\right) \quad \text{(Eq. 15)}$$

$$p_e = p_{Yes} + p_{No} \quad \text{(Eq. 16)}$$

$$Cohen's\ Kappa = \frac{p_o - p_e}{1 - p_e} \quad \text{(Eq. 17)}$$

The Matthews Correlation Coefficient (MCC) is used to measure the quality of a binary classification and is essentially a correlation coefficient between the true and the predicted binary classifications. A value of 1 represents a perfect prediction, 0 represents the predictions that are no better than random, and -1 represents total disagreement between predictions and true values. The MCC can be calculated as follows (Shin, 2020):

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad \text{(Eq. 18)}$$

3.4 Model Fine-Tuning

Training deep models typically requires a dataset of sufficient size and training time (weeks or months) (He et al., 2016b). Instead of training deep models from scratch, this work follows the

common practice for solving computer vision problems, which is using transfer learning with a pretrained model for the task of interest. The model selected for this work is ResNet152, a popular backbone architecture in computer vision related designs. It is pretrained on the ImageNet dataset which contains 1.2 million images with 1000 categories. This work focuses on fine-tuning the pretrained ResNet152 model on damaged buildings extracted from satellite imagery. Although this ResNet model will be used on a satellite image, generic image descriptors can be extracted from an unrelated dataset, such as a nature image dataset, and then applied to a different image classification task via transfer learning (L. D. Nguyen et al., 2018). In transfer learning, the base model is trained on a base dataset and task. These learned features are then transferred to a second target network to be trained on the target dataset and task. When the target dataset is smaller than the base dataset, transfer learning can be a powerful tool to train large ResNet networks without the possibility of overfitting (Yosinski et al., 2014). In transfer learning there is a choice of whether to fine-tune the first n layers of the target network. The number of buildings extracted from the satellite image is relatively small compared to the ImageNet nature dataset where the pre-trained ResNet152 is being trained. Therefore, the first layers of the target model will be left "frozen" to keep from overfitting the data. The shallow part of the model tends to learn filters that correspond to more general image features, such as edges and corners. They are recognized as fixed features for images from different application fields. By transferring the pretrained base model to a target model, the weights from the first n layers are proven to be superior to that of an un-pretrained model with random weights. The further away from the first n layers the data goes through, the less useful the pretrained layers will be (Yosinski et al., 2014).

To overcome the issue that collection of a large corpus of training data is often expensive and laborious, data augmentation (DA) is used to improve the model performance by artificially

inflating the original training set with label preserving transformations (Taylor & Nitschke, 2018). Geometric-based transformations methods alter the geometry of the image by mapping the individual pixel values to new destinations, such as flipping, rotation, and cropping schemes. Photometric-based methods alter the RGB channels by shifting each pixel value to new pixel values with pre-defined heuristics, such as color jittering, edge enhancement and fancy PCA. In this work, the rotation scheme was tested by rotating each extracted building θ degrees all the way to 360 degrees as shown in Figure 11 to increase the amount of training data fivefold. The transformation function is:

$$T = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\,\theta \end{pmatrix} * S \qquad \text{(Eq. 19)}$$

Where S is the original image, T is the transformed image, and θ is the rotation angle.

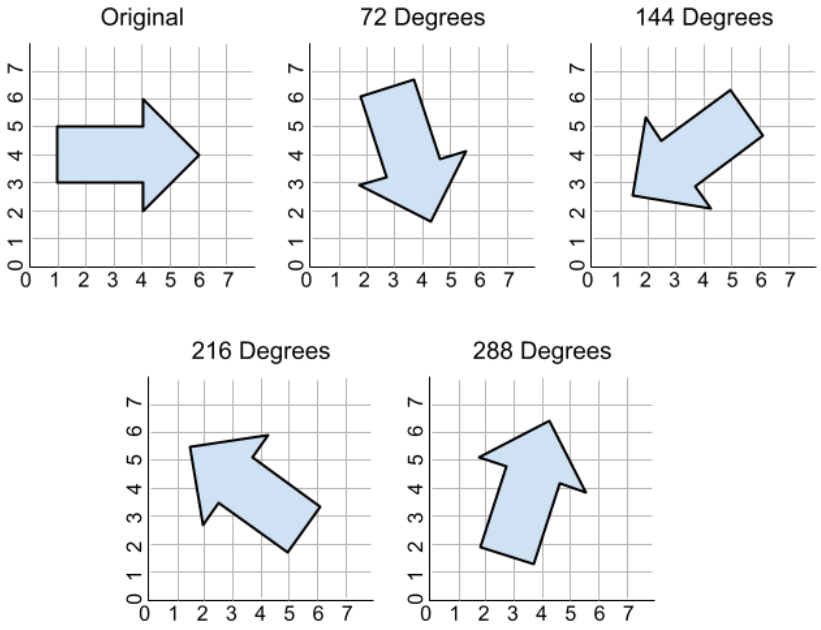

Figure 11. Example of augmented images after the original sample image is rotated 72 degrees consecutively.

A few small tests were conducted to test how the degree of rotation affected the results. For example, a 90-degree rotation was applied to the training sets, however the testing accuracy was lower. A balance between accuracy and training time was struck therefore no more than a 72-degree rotation was applied. Also, in addition to being rotated, all images sampled were also georeferenced. The transfer learning principle allows the model to learn the weights of the specific damaged building classification task more effectively. Additionally, increasing the amount of training data used in the chosen method also increased the accuracy of the model and helped the model to generalize better, which will be demonstrated in Chapter 4.

Chapter 4

EXPERIMENTAL STUDY

In this section, the performance of the proposed method introduced in Chapter 3 is demonstrated. The experiments have been implemented under the Jupyter Notebook environment. The WorldView-2 (WV2) satellite imagery of Port-au-Prince, Haiti is provided by Maxar Technologies, acquired on 15 January 2010 after the magnitude 7.0 Haitian earthquake of 12 January 2010. It has a spatial resolution of 1.84 meters (Satellite Imaging Corporation, n.d.) and is freely available under the Maxar open data program. The data consists of 4 images mosaicked together with 3-bands each, in the red, green, and blue spectrums. Ground truth data was collected from the products in support of the Post Disaster Needs Assessment and Recovery Framework (PDNA), produced jointly by the United Nations Institute for Training and Research (UNITAR) Operational Satellite Applications Programme (UNOSAT), the European Commission (EC) Joint Research Centre (JRC), and the World Bank.

4.1 Sample Datasets

Building footprints were extracted from the WV2 satellite imagery and manually labeled as “Damage” or “No Damage” based on the ground truth data. They were randomly split into 322 training samples, 46 validation samples, and 93 testing samples. Figure 12 shows these sample datasets in each category and their distribution in the area. These areas were chosen because 1) detailed building damage info is publicly available; and 2) buildings in these areas are representative in terms of density, size, and structure throughout the commune of Port-au-Prince.

Figure 12. Sample data locations and categories (orange - training, yellow - validation, red - testing A, blue - testing B).

The ResNet152 model was first fine-tuned with the extracted 322 training samples. A second experiment was conducted with the augmented training samples (training^{+}) that are generated from the rotation transformation. Both experiments consisted of using the zero padded training images extracted from the WV2 satellite imagery. In the data augmentation (DA) process, each of these 322 buildings, as previously discussed in Chapter 3, was rotated 72 degrees continuously until 360 degrees shown in Figure 13, thereby increasing the amount of training data

fivefold. Table 1 summarizes the amount of labeled sample data used for each stage of the process. Training data is used for learning the process and to fit the weights for the ResNet152 classifier. Validation data works to tune the parameters of the ResNet152 classifier and to determine, for example, if the training data is sufficient. Finally, the testing data is used to assess the performance of the fully trained classifier, the final model.

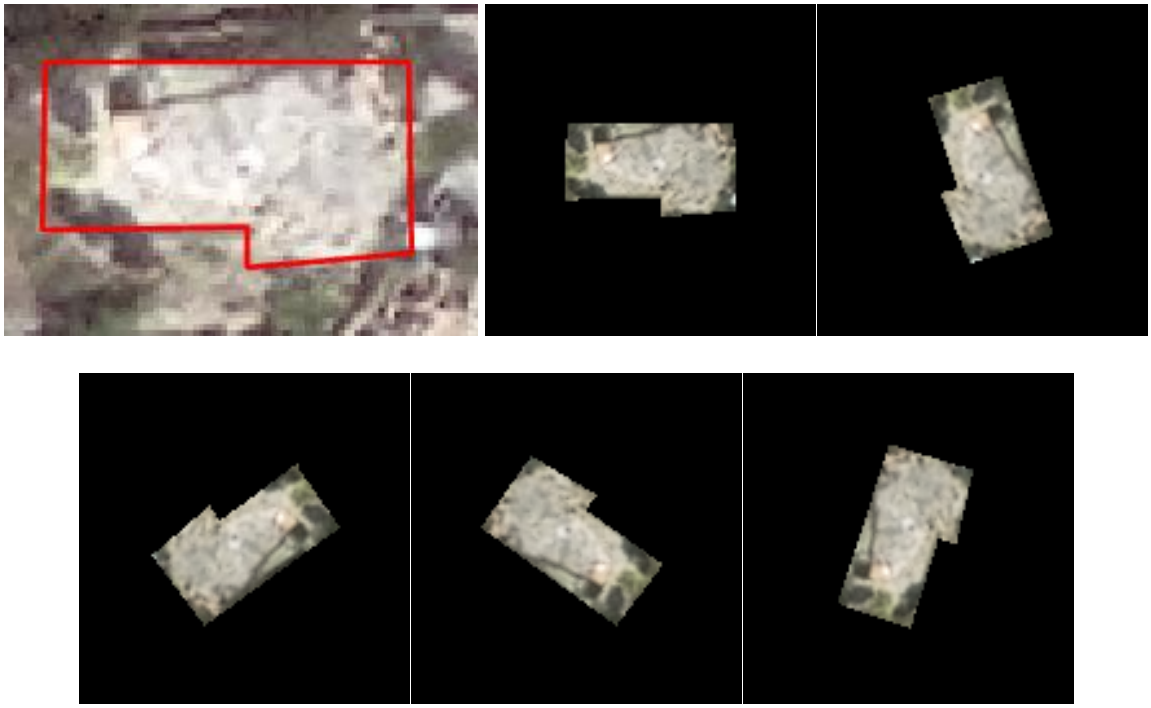

Figure 13. The top left image is the original image from the WV2 satellite with the boundary used for image chip extraction. The top middle image is that of the extracted and padded image. The other images show the DA results with 72-degree rotation consecutively.

Table 1. Summary of the amount of training, validation, and testing samples to be used in the fine-tune process.

Sample Types	Total	Labeled as Damage	Labeled as No Damage
Training (Original)	322	182	140
Training^{+} (after DA)	1610	910	700
Validation	46	26	20
Testing	93	53	40

4.2 Experiment Implementation

The ResNet152 residual network model was fine-tuned with training samples in two separate experiments (with original training samples and with training^{+} samples). To assess the results of the training, validation, and testing modules, the respective samples were predicted using the trained model to enable the construction of confusion matrices. The assessment matrices of accuracy, precision, sensitivity, the F1 score, Cohen's Kappa, and the Matthews Correlation Coefficient (MCC) are derived afterwards.

One critical issue in machine learning is to avoid overfitting or underfitting a model. The purpose of training is to make the validation loss zero. However, if the validation loss is larger than the training loss, overfitting is indicated, and when the opposite is true there is underfitting. If the validation loss and the training loss are equal to each other, robust fitting is achieved (Brownlee, 2019b). A technique called "early stopping" was applied. Using the early stopping technique, the model stops being trained when the validation loss stops improving for five epochs

straight. The purpose of early stopping is to improve generalization of deep neural networks and to keep a model from overfitting (Brownlee, 2019a).

With early stopping, the model with training^{+} data applied stopped improving validation loss on the 15th epoch, entailing the model trained for a total of 20 epochs shown in Table 2. When testing with original training data applied, the model was only trained for six epochs. Therefore, for the two experiments to have a common framework of early stopping, the 20-epoch DA result will be compared to the six-epoch original training data result. Figure 14 shows the results of the ResNet152 models epochs comparing the original training data to the training^{+} data. The figure shows at the 15th epoch there is the lowest validation loss for the DA model, which is the goal, however, the difference between validation loss and training loss at that point is one of the greatest in the model. The 17th epoch shows that validation loss is higher than training loss by .01, however, at this point validation loss has peaked. The model stops at the 20th epoch because of early stopping. It is also the epoch where there is a difference of .09 between validation and training, making it the epoch with the second lowest difference and a location with the third lowest validation loss.

Table 2. DA model training return showing training and validation loss.

epoch	train_loss	valid_loss	accuracy	time
1	1.042376	0.903653	0.515528	0:36
2	0.936403	0.731321	0.720497	0:22
3	0.874362	0.575533	0.770186	0:21
4	0.783001	0.508756	0.776398	0:21
5	0.767506	0.537253	0.782609	0:21
6	0.751107	0.613045	0.782609	0:21
7	0.729267	0.525315	0.751553	0:21

8	0.694654	0.46184	0.807453	0:21
9	0.660684	0.433513	0.807453	0:21
10	0.653519	0.441762	0.801242	0:21
11	0.605963	0.488908	0.813665	0:21
12	0.583887	0.500495	0.782609	0:21
13	0.562885	0.400379	0.819876	0:21
14	0.534634	0.310225	0.869565	0:21
15	0.523058	0.288835	0.850932	0:21
16	0.521369	0.366022	0.826087	0:21
17	0.498879	0.51325	0.807453	0:21
18	0.474929	0.376632	0.832298	0:21
19	0.440805	0.320855	0.869565	0:21
20	0.406367	0.316084	0.863354	0:21

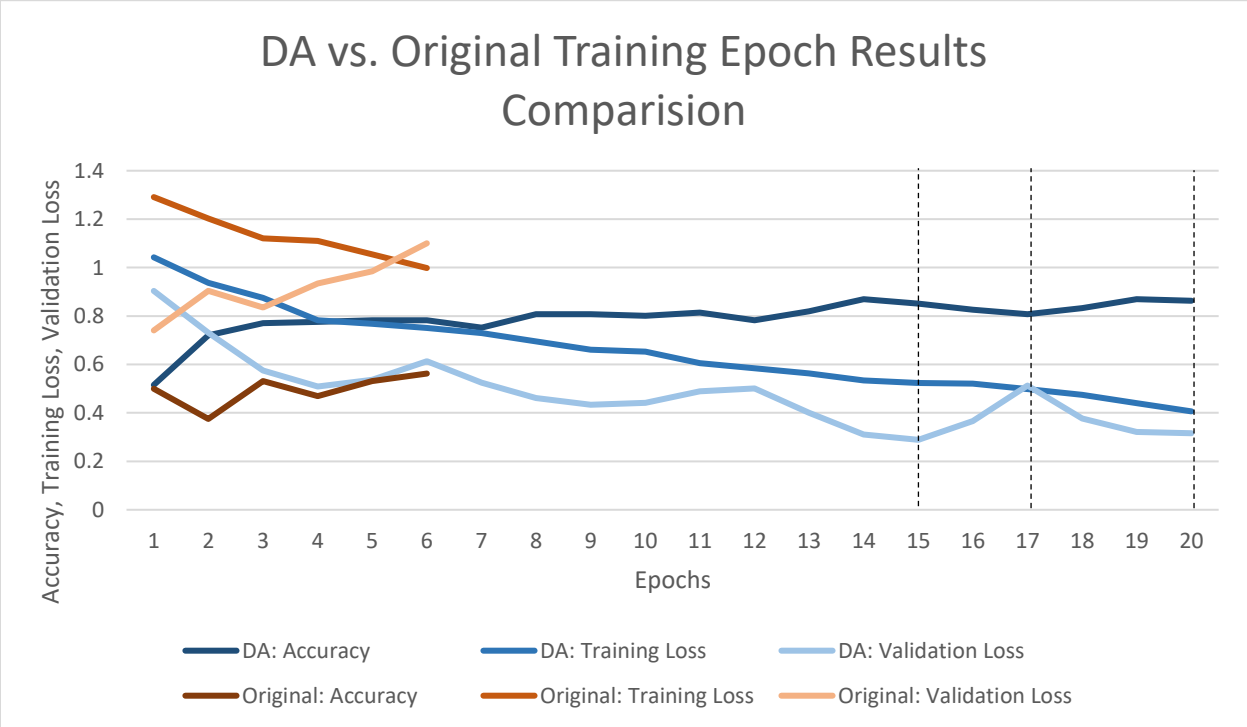

Figure 14. Graph comparing data augmentation and original model training results.

Using the early stopping result of 20 epochs with the training^{+} data is the optimal model for a variety of reasons. The amount of time it takes to train the 20-epoch model is significantly less. It took 7.27 minutes to train 20 epochs on training^{+} images compared to 12.48 minutes and 18.37 minutes to train 34 and 50 epochs respectively. Secondly, the training and validation loss were the closest at 20 epochs while also taking into consideration the low validation loss. Thirdly, the early stopping feature will not only prevent overfitting but also allow for better generalization for future unseen instances (Brownlee, 2019a). Finally, when comparing accuracies between the 20, 34, and 50-epoch models, the 34 and 50-epoch models only show a slight increase in accuracy at the cost of a significant amount of time.

A confusion matrix (Table 3) can be derived for the results, where four different outcomes, true positive (TP), true negative (TN), false positive (FP), and false negative (FN), will be counted. In this work, positive refers to the "Damage" classification and negative refers to "No Damage." TP and TN refer to the outcome where the model correctly predicts the intended class. FP, or type 1 error, and FN, or type 2 error, refer to an outcome where the model incorrectly predicts the class (Shin, 2020). The sum of all numbers in the matrix represents the total number of data samples.

Table 3. Structure for a 2x2 Confusion Matrix.

		Predicted	
		Damage (P)	No Damage (N)
True	Damage (P)	TP	FN
	No Damage (N)	FP	TN

4.3 Result Discussions

The two experiments conducted show that the one with training^{+} is superior to the one with original training samples. Figure 15 shows a comparison of the results from the two experiments as well as comparing training, validation, and testing accuracies. When DA is applied to the training images, there are training, validation, and testing accuracy of 92%, 85%, and 83% respectively. Using the original training data, the training, validation, and testing accuracies are 50%, 43%, and 53% respectively. On average, the DA accuracies are 38% more accurate when compared to the model with original training samples. There is also an average of a 33% and a 46% improved accuracy result when analyzing the damage and no damage precisions from the original training data and training^{+} data, respectively. Additionally, the recall assessment for damage and no damage show improvement for the DA result, increasing an average of 29% and 49% respectively. Looking at the measure of the test's accuracy, or the F1 score calculated from precision and recall, across the board, the test for classifying damaged buildings is better than the test for classifying no damage buildings. The F1 score also shows that the average difference of the two models is between 31% for damage and 47% for no damage. Cohen's Kappa, the measure of the inter-relatability between the prediction classes and ground truth classes, shows that the model with DA has an average of .78 more inter-relatability than that of the original model. Lastly, MCC values can range from -1 to 1, where -1 indicates total disagreement between prediction and ground truth, 0 indicates the model is random, and 1 means the model has perfect prediction. The MCC score average of the model with DA is .72 and the model with original training data is -.06, demonstrating the model with DA performs much better than the one with original samples.

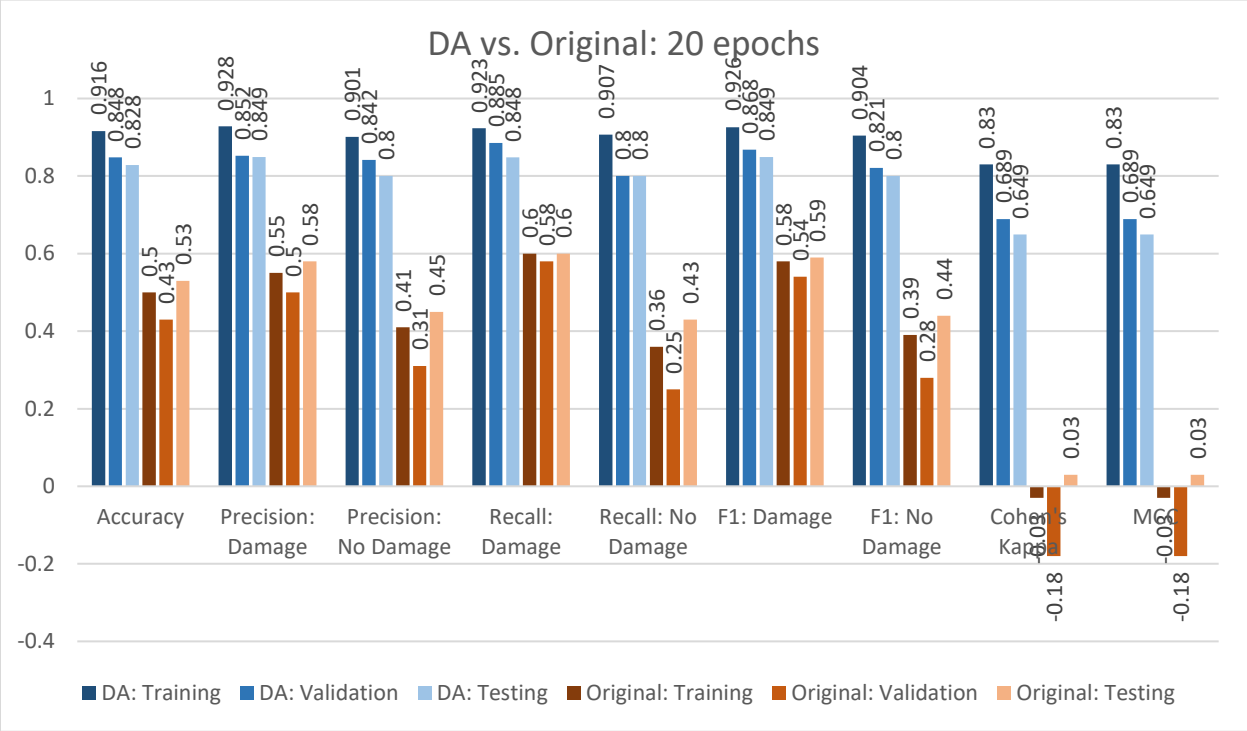

Figure 15. Graph comparing DA and original model statistics.

Table 4. Confusion matrices of training+, validation, and testing predictions (DA).

Training		Predicted Label	
True Label		Damage	No Damage
	Damage	168	14
	No Damage	13	127
Validation		Predicted Label	
True Label		Damage	No Damage
	Damage	23	3

	No Damage	4	16
Testing	Predicted Label		
True Label		Damage	No Damage
	Damage	45	8
	No Damage	8	32

The results show that when applying DA to the training samples and training the ResNet152 model on that data, there is a remarkably improved performance compared with the model with original training samples. Accuracy, precision, and recall are all higher between the two models suggesting that when training a model, data augmentation on the training data, if the training sample data is limited, is an effective way to improve the model performance.

4.4 Generalization Performance Test of the Fine-Tuned DA Model

In this subsection, the DA fine-tuned model will be tested in different areas for its generalization performance. This section focuses on how building density and building size will affect the trained ResNet152 using a different set of test samples taken from the same image but at a different location and disposition. The first of the two testing data sets is randomly generated as seen in the previous subsection 4.1, and is labeled as ‘Testing A,’ The other is the new set, which is to test the generalization performance of the model and is labeled as ‘Testing B’.

Testing B comprises of 189 samples collected from seven city blocks as shown in Figure 16. These samples, when compared with the 93 samples collected for Testing A, are not spread out over a large area and are not sparsely sampled. Additionally, the building footprints in the Testing B samples are significantly smaller. Table 5 shows that the samples collected for Testing

B are, on average, significantly smaller compared to the samples collected for training in Testing A.

Figure 16. Testing B samples. Red shows damaged buildings and blue shows buildings with no damage.

Table 5. Comparison of building footprint areas.

Area (sqm)	Training	Testing A	Testing B
Average	291.7	360.1	197.0
Median	197.0	200.0	150.3
Min	30.1	18.9	33.1
Max	2058.0	1912.2	1092.7

The accuracy of Testing B was calculated to be 63% compared to that of Testing A's accuracy of 83%. Precision, or the portion of damage or no damage predictions that are correctly predicted, turned out to be 50% and 80% respectively. The damage precision is 50%, far lower than Testing A's 85%. Interestingly, however, no damage's classification for Testing A and Testing B were found to be the same, 80%. Looking at the recall metrics for Testing A and Testing B, we can see that actual positives ("Damage"), or ground truth positives, that are identified correctly, are 85% and 76% respectively. Looking at the recall figure for the no damage classification, comparing Testing A to Testing B, the recall was calculated to be 80% and 55%, respectively. The recall for Damage from both testing areas are superior to that of no damage. The F1 score, or the measure of preciseness and robustness of the model, turned out to be 85% and 60% for Testing A and Testing B's damaged building classification. It also shows 80% and 65% F1 score for the no damage classification. These measurements show that for the purposes of generalization, the trained ResNet152 model is not robust for generalizing on unseen instances. Finally, Cohen's Kappa and the Matthews Correlation Coefficient indicate Testing A far outperforms Testing B. Cohen's Kappa is the measure of how much the prediction and the ground truth labels agree. Testing A has a Cohen's Kappa of .65 whereas Testing B has a Kappa of .28.

The MCC is a measure of the correlation between the ground truth and the predicted binary classification. Testing A received a value of .65 and Testing B received a value of .30. The Testing B value of .30 entails that the correlation between the ground truth and predicted classifications is more random than that of Testing A's MCC value.

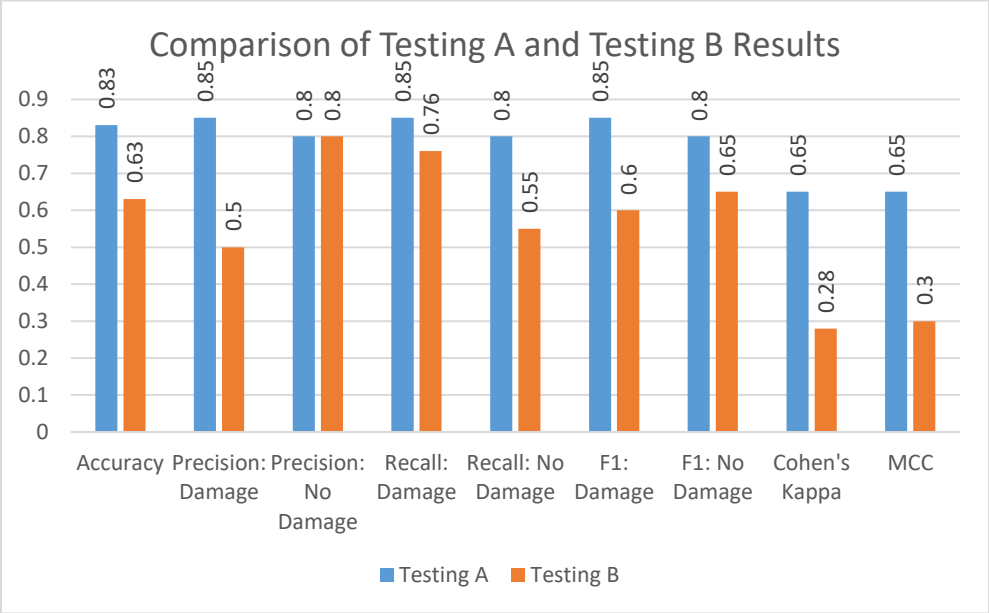

Figure 17. Comparison of Testing A to Testing B.

Table 6. Testing B confusion matrix.

Testing B	Predicted Label		
True Label		Damage	No Damage
	Damage	53	17
	No Damage	53	66

The relatively low generalization performance of the trained ResNet152 model can be caused by a couple of factors. One of the factors is the building footprint area of Testing B compared to that of the training building footprints. In Table 5 above, the Testing B average building footprint area is 197.0 sqm compared to the building footprint area of 291.7 sqm for training and 360.1 sqm for Testing A. The small area of the Testing B footprints could have a significant impact on the generalization results since textures included would be smaller, resulting in less features available for classification by the model. Another factor for the relatively poor generalization performance, is the damaged building texture, notably shown in the precision results. The no damage building footprints were taken from buildings with no damage, that is, the rooftops of the buildings are relatively similar throughout the satellite image in the area, whether they are from Testing A or Testing B. Therefore, the precision value for the no damage classification is remarkably the same, 80%. With damaged building footprints, there are many variations to the samples depending on the amount of damage, the type of building rooftop material, and the general non-uniform nature of collapsed buildings. It is difficult to train a model to classify unseen patterns of damage with limited training samples. Therefore, it is reasonable that the Testing B precision results are far less, 50%, then that of Testing A, 85%. Precision, in this case, is the portion of predicted damaged buildings that were correctly predicted as such. The classifier mistook 53 non-damaged rooftops with having damage suggesting that the model needs further work on classifying damaged buildings, determining what is and what is not a damaged rooftop.

Chapter 5

CONCLUSIONS AND FUTURE WORK

This work presented a promising approach to classify damaged and non-damaged buildings on high-resolution satellite imagery using a deep neural network when a large training dataset is not feasible or laborious to collect. With the proposed generic data augmentation based on geometric transformation, the training set can be artificially inflated with labels preserved. As the experiments demonstrated, the DA scheme further boost CNN performance and prevents over-fitting. Additionally, I proposed a fine-tuning strategy with transfer learning that fine-tunes high-level layers of pre-trained ResNet152 model, while keep the low-level layers unchanged. My experimental results indicate that using pre-trained architectures, although not pre-trained on the application of this work (i.e., damaged or no-damaged buildings), is highly effective and results in promising classification (85% and 80% in Testing A precision and recall for classifying damaged and no-damaged buildings respectively). Moreover, the fine-tuning process converged with 20 epochs of training iterations where the validation loss is close to the training loss, which shows the effectiveness of our method.

Future work would be focused on exploring the automating methods to extract building footprint, label training data, and investigating model generalization techniques. It is not in doubt that a larger training dataset would further improve the model performance. However, manual extraction is slow, costly, and laborious. Building detection ML models have been developed to identify buildings from high resolution satellite images, e.g., Fast-RCNN by Ren et. al. (Ren et al., 2017). The challenge remains in automating their labeling. Model generalization is a non-trivial task. Typically, a simple way to improve generalization in ML models is to increase training data size and variation. This is challenging, especially in the damage assessment domain due to a very

small number of past disasters where high-resolution satellite imagery and manual damage assessments are available. This means the model is trained with data that have limited variation in building characteristics, lighting conditions, terrain types, image quality, and camera angles. The model can easily be overfit on the training data and perform poorly out-of-sample. The experiment shown in Section 4.4 illustrated such a need, particularly for classifying damaged buildings due to the lack of data variability in the trained damaged building samples. Further work should investigate techniques to make the model more robust to data flaws. For example, more data augmentation techniques should be investigated to be included in the fine-tuning process.

REFERENCES

Abraham, A. (2005). Adaptation of Fuzzy Inference System Using Neural Learning. In N. Nedjah & L. Macedo Mourelle (Eds.), *Fuzzy Systems Engineering* (Vol. 181, pp. 53–83). Springer. https://doi.org/10.1007/11339366_3

Acar, A., & Muraki, Y. (2011). Twitter for crisis communication: lessons learned from Japan's tsunami disaster. *International Journal of Web Based Communities*, *7*(3), 392–402. https://doi.org/10.1504/IJWBC.2011.041206

Alam, F., Imran, M., & Ofli, F. (2017). Image4Act: Online Social Media Image Processing for Disaster Response. *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 - ASONAM '17*, 601–604. https://doi.org/10.1145/3110025.3110164

Alam, F., Ofli, F., & Imran, M. (2018). Processing Social Media Images by Combining Human and Machine Computing during Crises. *International Journal of Human–Computer Interaction*, *34*(4), 311–327. https://doi.org/10.1080/10447318.2018.1427831

Anniballe, R., Noto, F., Scalia, T., Bignami, C., Stramondo, S., Chini, M., & Pierdicca, N. (2018). Earthquake damage mapping: An overall assessment of ground surveys and VHR image change detection after L'Aquila 2009 earthquake. *Remote Sensing of Environment*, *210*, 166–178. https://doi.org/10.1016/j.rse.2018.03.004

Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., & Marchetti, M. (2018). On the Effectiveness of Machine and Deep Learning for Cyber Security. In T. Minárik, R. Jakschis, & L. Lindström (Eds.), *2018 10th International Conference on Cyber Conflict (CyCon)* (pp. 371–389). NATO CCD COE Publications. https://doi.org/10.23919/CYCON.2018.8405026

Blaschke, T. (2004). Object-based contextual image classification built on image segmentation. *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data, 2003*, 113–119. https://doi.org/10.1109/WARSD.2003.1295182

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Brownlee, J. (2019a, August 6). *A gentle introduction to early stopping to avoid overtraining neural networks*. Machine Learning Mastery. https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/

Brownlee, J. (2019b, August 6). *How to use learning curves to diagnose machine learning model performance*. Machine Learning Mastery. https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

Cable, S. (2015). *Aerial photography and the First World War*. The National Archives. https://blog.nationalarchives.gov.uk/aerial-photography-first-world-war/

Cervone, G., Sava, E., Huang, Q., Schnebele, E., Harrison, J., & Waters, N. (2016). Using Twitter for tasking remote-sensing data collection and damage assessment: 2013 Boulder flood case study. *International Journal of Remote Sensing*, *37*(1), 100–124. https://doi.org/10.1080/01431161.2015.1117684

Comfort, L. K. (1996). *Self Organization in Disaster Response: The Great Hanshin, Japan Earthquake of January 17, 1995*. https://www.researchgate.net/publication/228364362_Self-organization_in_Disaster_Response_The_Great_Hanshin_Japan_Earthquake_of_January_17_1995

Cooner, A., Shao, Y., & Campbell, J. (2016). Detection of Urban Damage Using Remote Sensing and Machine Learning Algorithms: Revisiting the 2010 Haiti Earthquake. *Remote Sensing*, *8*(10), 868. https://doi.org/10.3390/rs8100868

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Dertat, A. (2017, November 8). *Applied deep learning - Part 4: Convolutional neural networks*. Towards Data Science. https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Dhodhi, M. K., Saghri, J. A., Ahmad, I., & Ul-Mustafa, R. (1999). D-ISODATA: A Distributed Algorithm for Unsupervised Classification of Remotely Sensed Data on Network of Workstations. *Journal of Parallel and Distributed Computing*, *59*(2), 280–301. https://doi.org/10.1006/jpdc.1999.1573

Donges, N. (2018). *The Random Forest Algorithm – Towards Data Science*. Towards Data Science. https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

Duarte, D., Nex, F., Kerle, N., & Vosselman, G. (2018). SATELLITE IMAGE CLASSIFICATION OF BUILDING DAMAGES USING AIRBORNE AND SATELLITE IMAGE SAMPLES IN A DEEP LEARNING APPROACH. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(2), 89–96. https://doi.org/10.5194/isprs-annals-IV-2-89-2018

Duro, D. C., Franklin, S. E., & Dubé, M. G. (2012). A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery. *Remote Sensing of Environment*, *118*, 259–272. https://doi.org/10.1016/j.rse.2011.11.020

Elkan, C. (1999). *What is "fuzzy logic"? Are there computers that are inherently fuzzy and do not apply the usual binary logic?* Scientific American. https://www.scientificamerican.com/article/what-is-fuzzy-logic-are-t/

Estoque, R. C., Murayama, Y., & Akiyama, C. M. (2015). Pixel-based and object-based

classifications using high- and medium-spatial-resolution imageries in the urban and suburban landscapes. *Geocarto International*, *30*(10), 1113–1129. https://doi.org/10.1080/10106049.2015.1027291

Guyon, I., Vapnik, V., Boser, B., Bottou, L., & Solla, S. A. (1992). Capacity control in linear classifiers for pattern recognition. *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, *2*, 385–388. https://doi.org/10.1109/ICPR.1992.201798

Haralick, R. M., & Shapiro, L. G. (1985). Image Segmentation Techniques. *Computer Vision, Graphics, and Image Processing*, *29*(1), 100–132. https://www.sciencedirect.com/science/article/abs/pii/S0734189X85901537

He, K., Zhang, X., Ren, S., & Sun, J. (2016a). *Identity Mappings in Deep Residual Networks*. 1–15. http://arxiv.org/abs/1603.05027

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hiltz, Starr Roxanne, Kushma, J., & Plotnick, L. (2014). Use of Social Media by U.S. Public Sector Emergency Managers: Barriers and Wish Lists. In S. R. Hiltz, M. S. Pfaff, L. Plotnick, & P. C. Shih (Eds.), *Proceedings of the 11th International ISCRAM Conference on nformation systems for crisis response and management* (pp. 602–611). The Pennsylvania State University. https://doi.org/10.13140/2.1.3122.4005

Hinkelmann, K. (n.d.). *Neural Networks*. Retrieved January 30, 2020, from http://didattica.cs.unicam.it/lib/exe/fetch.php?media=didattica:magistrale:kebi:ay_1718:ke-11_neural_networks.pdf

Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. (2016). *Deep Networks with Stochastic Depth*. 1–16. http://arxiv.org/abs/1603.09382

Huang, Q., & Xiao, Y. (2015). Geographic Situational Awareness: Mining Tweets for Disaster Preparedness, Emergency Response, Impact, and Recovery. *ISPRS International Journal of Geo-Information*, *4*(3), 1549–1568. https://doi.org/10.3390/ijgi4031549

Hussain, E., Ural, S., Kim, K., Fu, C.-S., & Shan, J. (2011). Building Extraction and Rubble Mapping for City Port-au-Prince Post-2010 Earthquake with GeoEye-1 Imagery and Lidar Data. *Photogrammetric Engineering and Remote Sensing*, *77*(10), 1011–1023. https://doi.org/10.14358/pers.77.10.1011

Hussain, M., Chen, D., Cheng, A., Wei, H., & Stanley, D. (2013). Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, *80*, 91–106. https://doi.org/10.1016/J.ISPRSJPRS.2013.03.006

Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by*

Reducing Internal Covariate Shift. 1–11. http://arxiv.org/abs/1502.03167

Janalipour, M., & Mohammadzadeh, A. (2016). Building Damage Detection Using Object-Based Image Analysis and ANFIS From High-Resolution Image (Case Study: BAM Earthquake, Iran). *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *9*(5), 1937–1945. https://doi.org/10.1109/JSTARS.2015.2458582

Joshi, A. R., Tarte, I., Suresh, S., & Koolagudi, S. G. (2017). Damage Identification and Assessment using Image Processing on Post-Disaster Satellite Imagery. *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, 1–7. https://doi.org/10.1109/GHTC.2017.8239286

Kavzoglu, T., & Colkesen, I. (2009). A kernel functions analysis for support vector machines for land cover classification. *International Journal of Applied Earth Observation and Geoinformation*, *11*(5), 352–359. https://doi.org/10.1016/j.jag.2009.06.002

Kim, J., Kim, B.-S., & Savarese, S. (2012). Comparing Image Classification Methods: K-Nearest-Neighbor and Support-Vector-Machines. In *Proceedings of the 6th WSEAS International Conference on Computer Engineering and Applications, and Proceedings of the 2012 American Conference on Applied Mathematics* (pp. 133–138). World Scientific and Engineering Academy and Society.

Kolbe, A. R., Hutson, R. A., Shannon, H., Trzcinski, E., Miles, B., Levitz, N., Puccio, M., James, L., Noel, J. R., & Muggah, R. (2010). Mortality, crime and access to basic needs before and after the Haiti earthquake: a random survey of Port-au-Prince households. *Medicine, Conflict and Survival*, *26*(4), 281–297. https://doi.org/10.1080/13623699.2010.535279

Lagerstrom, R., Arzhaeva, Y., Szul, P., Obst, O., Power, R., Robinson, B., & Bednarz, T. (2016). Image Classification to Support Emergency Situation Awareness. *Frontiers in Robotics and AI*, *3*(54), 1–11. https://doi.org/10.3389/frobt.2016.00054

Lary, D. J., Alavi, A. H., Gandomi, A. H., & Walker, A. L. (2016). Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, *7*(1), 3–10. https://doi.org/10.1016/J.GSF.2015.07.003

LeCun, Y., & Bengio, Y. (1995). Convolutional Networks for Images, Speech, and Time-Series. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 1–14). MIT Press. https://www.researchgate.net/publication/2453996

Li, M., Zang, S., Zhang, B., Li, S., & Wu, C. (2014). A Review of Remote Sensing Image Classification Techniques: the Role of Spatio-contextual Information. *European Journal of Remote Sensing*, *47*(1), 389–411. https://doi.org/10.5721/EuJRS20144723

Liu, D., & Xia, F. (2010). Assessing object-based classification: advantages and limitations. *Remote Sensing Letters*, *1*(4), 187–194. https://doi.org/10.1080/01431161003743173

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transactions on Geoscience and*

Remote Sensing, *55*(2), 645–657. https://doi.org/10.1109/TGRS.2016.2612821

Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., & Zhang, Y. (2018). Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sensing*, *10*(7), 1119. https://doi.org/10.3390/rs10071119

Mantero, P., Moser, G., & Serpico, S. B. (2005). Partially Supervised classification of remote sensing images through SVM-based probability density estimation. *IEEE Transactions on Geoscience and Remote Sensing*, *43*(3), 559–570. https://doi.org/10.1109/TGRS.2004.842022

Maruyama, Y., Tashiro, A., & Yamazaki, F. (2014). Detection of collapsed buildings due to earthquakes using a digital surface model constructed from aerial images. *Journal of Earthquake and Tsunami*, *8*(1), 1–13. https://doi.org/10.1142/S1793431114500031

Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, *16*(5–6), 555–559. https://doi.org/10.1016/S0893-6080(03)00115-1

Maulik, U., & Chakraborty, D. (2017, March). Remote Sensing Image Classification: A survey of support-vector-machine-based advanced techniques. *IEEE Geoscience and Remote Sensing Magazine*, *5*(1), 33–52. https://doi.org/10.1109/MGRS.2016.2641240

Maxwell, A. E., Warner, T. A., & Fang, F. (2018). Implementation of machine-learning classification in remote sensing: an applied review. *International Journal of Remote Sensing*, *39*(9), 2784–2817. https://doi.org/10.1080/01431161.2018.1433343

Mountrakis, G., Im, J., & Ogole, C. (2011). Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, *66*(3), 247–259. https://doi.org/10.1016/j.isprsjprs.2010.11.001

Myint, S. W., Gober, P., Brazel, A., Grossman-Clarke, S., & Weng, Q. (2011). Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. *Remote Sensing of Environment*, *115*(5), 1145–1161. https://doi.org/10.1016/j.rse.2010.12.017

Nguyen, D. T., Ofli, F., Imran, M., & Mitra, P. (2017). Damage Assessment from Social Media Imagery Data During Disasters. *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 - ASONAM '17*, 569–576. https://doi.org/10.1145/3110025.3110109

Nguyen, L. D., Lin, D., Lin, Z., & Cao, J. (2018). Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5. https://doi.org/10.1109/ISCAS.2018.8351550

Nogueira, K., Penatti, O. A. B., & dos Santos, J. A. (2017). Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*,

61, 539–556. https://doi.org/10.1016/j.patcog.2016.07.001

Ouyang, Z. T., Zhang, M. Q., Xie, X., Shen, Q., Guo, H. Q., & Zhao, B. (2011). A comparison of pixel-based and object-oriented approaches to VHR imagery for mapping saltmarsh plants. *Ecological Informatics*, *6*(2), 136–146. https://doi.org/10.1016/j.ecoinf.2011.01.002

Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, *26*(9), 1277–1294. https://doi.org/10.1016/0031-3203(93)90135-J

Peters, R., & Porto De Albuquerque, J. (2015). Investigating images as indicators for relevant social media messages in disaster management. In Palen, Büscher, Comes, & Hughes (Eds.), *Proceedings of the ISCRAM 2015 Conference*.

Platt, R. V., & Rapoza, L. (2008). An Evaluation of an Object-Oriented Paradigm for Land Use/Land Cover Classification. *The Professional Geographer*, *60*(1), 87–100. https://doi.org/10.1080/00330120701724152

Porto De Albuquerque, J., Herfort, B., Brenning, A., & Zipf, A. (2015). A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management. *International Journal of Geographical Information Science*, *29*(4), 667–689. https://doi.org/10.1080/13658816.2014.996567

Qian, J., Zhou, Q., & Hou, Q. (2007). COMPARISON OF PIXEL-BASED AND OBJECT-ORIENTED CLASSIFICATION METHODS FOR EXTRACTING BUILT-UP AREAS IN ARIDZONE. *ISPRS Workshop on Updating Geo-Spatial Databases with Imagery & The 5th ISPRS Workshop on DMGISs*, 163–171. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.221.8137&rep=rep1&type=pdf

Rathje, E. M., & Adams, B. J. (2008). The Role of Remote Sensing in Earthquake Science and Eengineering: Opportunities and Challenges. *Earthquake Spectra*, *24*(2), 471–492. https://doi.org/10.1193/1.2923922

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Rollet, R., Benie, G. B., Li, W., Wang, S., & Boucher, J.-M. (1998). Image classification algorithm based on the RBF neural network and K-means. *International Journal of Remote Sensing*, *19*(15), 3003–3009. https://doi.org/10.1080/014311698214398

Satellite Imaging Corporation. (n.d.). *WorldView-2 satellite sensor*. Satellite Imaging Corporation. Retrieved March 24, 2021, from https://www.satimagingcorp.com/satellite-sensors/worldview-2/

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Shah, T. (2017, December 6). *About train, validation and test sets in machine learning*. Towards

Data Science. https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

Shalev-Shartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press. http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning

Shin, T. (2020, May 1). *Understanding the confusion matrix and how to implement it in python*. Towards Data Science. https://towardsdatascience.com/understanding-the-confusion-matrix-and-how-to-implement-it-in-python-319202e0fe4d

Sim, J., & Wright, C. C. (2005). The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, *85*(3), 257–268. https://doi.org/10.1093/ptj/85.3.257

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR 2015*, 1–14. http://arxiv.org/abs/1409.1556

Sisodia, P. S., Tiwari, V., & Kumar, A. (2014). Analysis of Supervised Maximum Likelihood Classification for Remote Sensing Image. *International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2014*, 1–4. https://doi.org/10.1109/ICRAIE.2014.6909319

Starbird, K., Muzny, G., & Palen, L. (2012). Learning from the Crowd: Collaborative Filtering Techniques for Identifying On-the-Ground Twitterers during Mass Disruptions. In L. Rothkrantz, J. Ristvej, & Z. Franco (Eds.), *Proceedings of the 9th International ISCRAM Conference* (pp. 1–10). Simon Fraser University. https://nlp.stanford.edu/~muzny/docs/smp-iscram2012.pdf

Sun, J., Yang, J., Zhang, C., Yun, W., & Qu, J. (2013). Automatic remotely sensed image classification in a grid environment based on the maximum likelihood method. *Mathematical and Computer Modelling*, *58*(3–4), 573–581. https://doi.org/10.1016/j.mcm.2011.10.063

Taskin Kaya, G., Musaoglu, N., & Ersoy, O. K. (2011). Damage Assessment of 2010 Haiti Earthquake with Post-Earthquake Satellite Image by Support Vector Selection and Adaptation. *Photogrammetric Engineering & Remote Sensing*, *77*(10), 1025–1035. https://doi.org/10.14358/PERS.77.10.1025

Taylor, L., & Nitschke, G. (2018). Improving Deep Learning with Generic Data Augmentation. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1542–1547. https://doi.org/10.1109/SSCI.2018.8628742

Tin Kam Ho. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 278–282. https://doi.org/10.1109/ICDAR.1995.598994

Tsang, S.-H. (2018, September 15). *Review: ResNet — Winner of ILSVRC 2015 (image classification, localization, detection)*. Towards Data Science. https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8

Turker, M., & Sumer, E. (2008). Building-based damage detection due to earthquake using the watershed segmentation of the post-event aerial images. *International Journal of Remote Sensing*, *29*(11), 3073–3089. https://doi.org/10.1080/01431160701442096

United States Geological Survey. (n.d.). *New earthquake Hazards program: Lists, maps, and statistics*. Retrieved May 19, 2020, from https://www.usgs.gov/natural-hazards/earthquake-hazards/lists-maps-and-statistics

Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V., & Chervyakov, N. I. (2020). Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, *177*, 232–243. https://doi.org/10.1016/j.matcom.2020.04.031

Voigt, S., Schneiderhan, T., Twele, A., Gähler, M., Stein, E., & Mehl, H. (2011). Rapid Damage Assessment and Situation Mapping: Learning from the 2010 Haiti Earthquake. *Photogrammetric Engineering & Remote Sensing*, *77*(9), 923–931. https://doi.org/10.14358/PERS.77.9.923

Wang, Q., Li, Q., Liu, H., Wang, Y., & Zhu, J. (2014). An improved ISODATA algorithm for hyperspectral image classification. *Proceedings - 2014 7th International Congress on Image and Signal Processing, CISP 2014*, 660–664. https://doi.org/10.1109/CISP.2014.7003861

Wolf, P. R., Dewitt, B. A., & Wilkinson, B. E. (2014). *Elements of Photogrammetry with Applications in GIS* (Fourth Edi). McGraw-Hill Education.

Yan, G., Mas, J. -F., Maathuis, B. H. P., Xiangmin, Z., & Van Dijk, P. M. (2006). Comparison of pixel-based and object-oriented image classification approaches—a case study in a coal fire area, Wuda, Inner Mongolia, China. *International Journal of Remote Sensing*, *27*(18), 4039–4055. https://doi.org/10.1080/01431160600702632

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems 27*, 3320–3328. http://arxiv.org/abs/1411.1792

Zhao, W. L., Deng, C.-H., & Ngo, C.-W. (2018). k-means: A revisit. *Neurocomputing*, *291*, 195–206. https://doi.org/10.1016/j.neucom.2018.02.072

Zhu, G., & Blumberg, D. G. (2002). Classification using ASTER data and SVM algorithms; *Remote Sensing of Environment*, *80*(2), 233–240. https://doi.org/10.1016/S0034-4257(01)00305-4