AMERICAN UNIVERSITY OF BEIRUT

TOWARDS BUILDING DAMAGE CHANGE DETECTION USING GRAPH CONVOLUTIONAL NETWORKS AND DOMAIN KNOWLEDGE: A CASE STUDY ON THE BEIRUT PORT EXPLOSION

by
ALI MOHAMAD ISMAIL

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
January 2022

AMERICAN UNIVERSITY OF BEIRUT

TOWARDS BUILDING DAMAGE CHANGE DETECTION USING GRAPH CONVOLUTIONAL NETWORKS AND DOMAIN KNOWLEDGE: A CASE STUDY ON THE BEIRUT PORT EXPLOSION

by

ALI MOHAMAD ISMAIL

Approved by:

Dr. Mariette Awad, Associate Professor Electrical and Computer Engineering	Advisor Mariette Awad Mariette Awad (Jan 22, 2022 23:48 GMT+2)
Dr. Wassim Masri, Professor Electrical and Computer Engineering	Member of Committee
Dr. Yaser Abunnasr, Associate Professor Landscape Design and Ecosystem Management	Member of Committee

Date of thesis defense: January 21, 2022

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name:	Ismail	Ali	Mohamad
	Last	First	Middle

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.

January 25, 2022

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I use this opportunity to properly thank those who helped drive this project.

First, I would like to thank my supervisor, Professor Mariette Awad, who was instrumental in shaping and directing this project, for her sound and tactful mentorship.

I would also thank our esteemed committee members Dr. Wassim Masri, Dr. Mona Fawaz and Dr. Yaser Abunnasr for their interest in this work and whose feedback has surely refined this work and improved its quality.

I acknowledge Maxar Technologies for donating satellite images of the Beirut area during dates surrounding the August 4 explosion. More specifically, I thank Mr. Rami Hazime and the team at Maxar for their helpful and prompt cooperation and for enabling the acquisition of the donated data.

I also credit the Open Map Lebanon team for readily providing the building damage annotations for the Beirut Port explosion.

An Abstract of the Thesis Proposal of

Ali Ismail for Master of Engineering
Major: Electrical and Computer Engineering

Title: Towards Building Damage Change Detection using Graph Convolutional Networks and Domain Knowledge: A Case Study on the Beirut Port Explosion

Change detection is a sub-field of remote sensing that aims to detect surface differences between two images taken at different times. It plays a significant part in detecting disaster damage and planning rescue operations. The advent of deep learning has lead to the development of many change detection solutions. Convolutional neural networks are at the core of recent approaches. As with most geographical phenomena, the spread of urban damage is more similar with buildings that in proximity. However, these networks only rely on local features and ignore the interactions and similarities between neighboring buildings. Also, it is important to map damage quickly whenever a new disaster occurs for an efficient response. Therefore it is not practical to wait for data to be annotated and models to be trained. Additionally, many structural building properties that are not based on proximity may impact the degree to which each building is damaged such as age and height. These properties can be very diverse especially in dense and irregularly urbanized cities and are not discernible from overhead imagery. In this work, we present a graph formulation for building damage change detection which enables learning relationships and representations from both local patterns and non-stationary neighborhoods that cannot be captured by traditional neural networks. We propose a novel architecture combining a Siamese convolutional neural network and a graph convolutional network which we train in a semi-supervised framework allowing the task to be performed with a small number of annotations and reducing the time and effort needed to obtain damage assessment. We also investigate a supervised variant and evaluate the possibility of generalizing to unseen disasters. We train and validate this approach on the xBD dataset. We also demonstrate this method on the Beirut Port Explosion

and show that performance is improved by incorporating domain knowledge from building meta-features.

Contents

Acknowledgements **iv**

Abstract **v**

1 Introduction **1**

2 Literature Review **5**
2.1 Physical Principles of Remote Sensing 5
2.2 Technology of Remote Sensing 6
2.3 Applications of Remote Sensing 7
2.4 General Background on Change Detection 7
2.4.1 Image Modalities in Change Detection. 8
2.4.2 Classical Methods for Change Detection 8
2.4.3 AI for Change Detection 9
2.4.4 AI for Building Damage Change Detection 11
2.4.5 Methodologies for Fast Change Detection 12
2.5 Graph Convolutional Networks 15
2.5.1 Concepts Behind Graph Convolutions 15
2.5.2 Graph Convolutional Networks in Computer Vision 18
2.6 State-of-the-Art Work 19
2.7 General Research Trends 21

3 Methodology **24**
3.1 Graph Data Formulation 24
3.2 Model Architecture 26

4 Experiments, Results and Discussion **28**
4.1 Datasets and Evaluation Metrics 28
4.1.1 xBD Dataset 28
4.1.2 Beirut Explosion Dataset 29
4.2 Model Evaluation 34
4.2.1 Performance Metrics 34
4.2.2 Model Benchmarking 36

4.2.3 Null Hypothesis Testing 37
4.3 Experimental Setup 38
4.4 Semi-supervised GCN with the xBD dataset 39
4.4.1 Benchmarking with Multiresolution Autoencoder 40
4.5 Semi-supervised GCN with the Beirut Explosion dataset 42
4.6 Supervised change detection and domain shift with Graph SAGE 44
4.7 Computational and performance considerations for best practices 46

5 Conclusion and Future Work **48**

A xBD Dataset Tier 1 Class Distribution **51**

B Abbreviations **53**

Bibliography **54**

List of Figures

2.1 Electromagnetic spectrum (Image credit: `https://www.nist.gov/image/designuassspectrum154169990hrjpg`). 5
2.2 Zachary's Karate Club [1], a common social relation problem represented by a graph. Each node is a student and the edges represent their social relations. Different colors represent how the students split into two factions after a conflict between the club owner and their coach. 15
2.3 High level depiction of 2D convolution (left) and graph convolution (right) (Image credit: [2]). 16
2.4 Different subcategories of the collected publications. 21
2.5 Publication year and venue type of the collected works. 22
2.6 Top 10 publication venues of the reviewed literature. 22
2.7 Evolution of CD methods over the years. 23

3.1 Concept of the Graph Formulation. Different node colors indicate nodes' different classes. 25
3.2 Subgraph implemented on a sample from the Joplin Tornado set. 26
3.3 Architecture of the proposed model. Objects in dotted lines indicate meta-feature injection if these features are available. 27

4.1 xBD disaster distribution across the regions (Image credit: [3]) . . 29
4.2 xBD Tier 3 number of buildings per class. 30
4.3 xBD Tier 3 chips with at least one damaged building. 31
4.4 Overview of the selected area. 32
4.5 Class distribution of the selected samples. 33
4.6 Before (left) and after (right) georeferencing. 33
4.7 A polygon covering only part of the building. 34
4.8 Buffered mask (red) around the original polygon (orange). 35
4.9 Counts of Beirut heritage buildings and Beirut buildings by function. 36
4.10 Semi-supervised autoencoder architecture (Image credit: [4]) . . . 37
4.11 Image chips of Joplin Tornado as squares of different color. 39

4.12 Above: Joplin Tornado damage as provided in the xBD dataset. Below: Joplin Tornado trajectory (Image credit: https://www.nist.gov/image/aerialimageofjoplinmissouritornadodamagejpg). 40
4.13 Model performance with respect to the training data size. 41
4.14 Performance difference with respect to the autoencoder for all three disasters. 42
4.15 GCN node embeddings for Joplin tornado. 43
4.16 Distribution of the performance metrics obtained after 30 runs with different data samples. 44
4.17 Difference between training and hold scores for the Siamese CNN and Graph SAGE. The x-axis indexes indicate the different experiments with different train - test configurations: 1) Fire - Fire, 2) Flooding - Fire, 3) Flooding + Tornado + Volcano - Fire and 4) Flooding + Tornado + Volcano + 10% Fire - Fire. 46

A.1 xBD Tier 1 building per class. 51
A.2 xBD Tier 1 chips with at least one damaged building. 52

List of Tables

2.1 Qualitative Comparison with State-of-the-art Work. 21

4.1 Summary Statistics of the Beirut Meta-features. 34
4.2 Comparison between our Semi-supervised GCN and the Multiresolution Autoencoder (AE) with Bold Italics Indicating Better Performance. 42
4.3 Shannon Equitability Index for the Chosen xBD Sets and the Difference between Specificity and Recall. 43
4.4 Comparison between the GCN with or without Meta-features on the Beirut Data with Bold Italics for Best Metrics. 44
4.5 Comparison between the Graph SAGE and the Siamese CNN. . . 45

Chapter 1

Introduction

Remote sensing (RS) is the process of capturing data using a sensory platform operating at a distance from the sensed target. Furthermore, the term remote sensing is predominantly used in reference to geospatial remote sensing where the objective is to collect data about the surface of the Earth. RS data can yield a wealth of information about the surface of the planet at both local and global scales. This information can be used to gain new knowledge about the environment, develop solutions to existing problems and make data-driven policies and decisions.

Change detection (CD) is a sub-field of remote sensing concerned with identifying and localizing differences in surface objects using RS images taken at different times [5]. CD has been successfully applied in many different fields such as forest monitoring [6, 7, 8], surface water monitoring [9, 10], sea ice monitoring [11, 12], landslide monitoring [13], tsunamis [14, 15], fires [16], urban development, LULC (land use land cover) [17, 18, 19] and planetary surface monitoring [20].

There are many success stories for CD being used for humanitarian assistance and disaster response. During the World Trade Center attacks satellite images were used to map the damaged area, detect locations with risk of ignition and detect rubble to help with planning of site cleanup and restoration. CD was equally used to plan disaster response and relief for hurricane Katrina [21]. In addition to humanitarian rescue and relief, ground surveys are carried in the wake of disasters to perform thorough inspections of the damage. These in person inspections are necessary to evaluate buildings' structural integrity and to estimate the necessary repair costs. These surveys require a large number of participants and can last for a long duration. In the case of the 2020 Beirut Port explosion, ground inspections took about a month to be carried [22]. While the damage maps obtained from satellite image CD are not detailed and granular enough to make these assessments, they can guide the efforts of field surveyors and help them implement plans for more effective and faster surveys. In all of the aforementioned scenarios, the most notable objects of interest were buildings [3]. In

this work, we focus our CD methodology on disaster-induced building damage detection.

Change detection is a fairly mature field. Initially, CD was performed using classical image processing techniques such as algebraic operations and image transformations [23]. With the increasing availability of huge amounts of data and with the AI revolution, CD techniques based on machine learning have emerged [24]. Most importantly deep learning methods, because of their breakthrough success in computer vision [25]. CD has been addressed using different deep architectures under both supervised and unsupervised learning settings. Notably, convolutional neural network (CNN) models [26] (U-Net [27], Siamese CNN [28], combining CNN with recurrent neural networks (RNN) [29]...) have been predominant across the literature [25]. More specifically, CD for building damage assessment has been addressed using similar approaches [30, 31, 32, 33, 34, 35, 15].

However, CNNs can only extract spatial information within the defined neighborhood of the convolution kernel. Graph convolutional networks (GCN) on the other hand are an emerging architecture capable of learning from data that is modeled as graphs [36]. Images have been cast into graphs using various formulations where GCNs can explore non-euclidean relationships between pixels that go beyond the pixel neighborhood defined by a CNN kernel [37]. These networks have outperformed CNN-based models in many computer vision applications [38, 37] including urban change detection [39].

Moreover, when a disaster has recently occurred, RS images are usually readily available within a number of hours [21]. However, first responders cannot wait until the new data is labeled and a model trained to plan their rescue and relief actions. Therefore, an important hallmark in any disaster CD solution is the ability to quickly obtain new predictions. This is important to consider given the fact that machine learning models can take days if not weeks to be properly developed and trained. In general, there are two broad directions for mitigating this problem: 1) semi-supervised learning which requires few labeled samples [15, 40, 41]; 2) pre-trained models that are used to infer on the new disaster without training [31, 42, 43, 44, 45].

Additionally, many cities include a mixture of buildings of various shapes, heights, time periods, building material and architectural styles in close proximity to each other. These architectural and structural differences can cause buildings to interact differently with the disaster and therefore sustain damage to varying levels. For instance, a short building could be shielded by a taller building and a heritage building would sustain more damage than surrounding modern buildings. While current approaches rely strictly on satellite or aerial imagery, it is not possible to capture the aforementioned architectural differences based on imagery alone. Therefore, the resulting interactions are not modeled well.

In summary, we identify three main gaps and problems in the field of building damage change detection. The first is that relationships between neighboring buildings which would have interacted with the disaster similarly based on prox-

imity are ignored in object-based CNN models. Secondly, we emphasize that the ability to quickly obtain damage assessments is critical in this application. Finally, we highlight that satellite images alone are not sufficient to capture subtle properties and building-level interactions which could lead to different levels of sustained damage.

Based on the above, we cast building damage CD into a node classification problem which we solve using a novel architecture combining a Siamese CNN and a GCN. This formulation leverages both the local and neighborhood features and patterns in the data. Also, it is well equipped to satisfy the two aforementioned solutions for obtaining predictions in a quick manner. When it comes to the semi-supervised approach, semi-supervised GCNs have been shown to be effective at propagating labels from labeled graph nodes to unlabeled nodes. As for the transfer of pre-trained models, different GCN architectures and graph aggregation methods have been proposed to learn on mini batches of data in a supervised way and infer on unseen data such as the sample and aggregate (SAGE) operator [46]. Furthermore, we demonstrate how architectural and contextual building features which we henceforth call meta-features can be incorporated into our graph to allow the model capture these subtle properties and interactions and improve the prediction results.

To summarize, the contributions of this work are the following:

- We present a novel formulation of the problem of building damage CD as a graph node classification which allows learning representations based on local features as well as relationships with neighboring samples.

- We propose a novel architecture based on a Siamese CNN combined with a GCN which we trained in a semi-supervised manner to reduce the number of labeled samples needed to obtain new predictions.

- We showcase our methodology on the Beirut Port explosion and we demonstrate how we augment the data with domain knowledge based on building meta-features to improve results.

- We investigate the transferability of the supervised SAGE variant of our model between different types of disasters.

To the best of our knowledge, there is no prior work which fuses both images and meta-features for building damage change detection. Also, there is no prior work which adopts a meta-feature framework with graph convolutional networks.

The rest of this document is organized as follows. In Chapter 2 we provide some background about remote sensing in general and specifically change detection. We review the literature related to different methodologies addressing CD. We also provide background on GCNs and review some works using GCNs for applications similar to CD. Finally, we highlight the previous works that are

considered state-of-the-art and show what they achieved and contrast their approaches to the solution presented here. In Chapter 3, we explain our proposed methodology which includes the graph formulation and the model architectures used. In Chapter 4, we provide details about the different datasets used and the experiments performed. We show and analyse our results and highlight our findings. In chapter 5, we summarize our targeted problem and the solution proposed to address this problem. We reemphasize key takeaways from this work and propose potential directions for improvement and future research.

Chapter 2

Literature Review

In this chapter, we provide an essential background around the central concepts of this thesis work such as remote sensing, change detection and graph convolutional networks. We also review the works related to CD and GCNs in general and focus on their applications in building damage detection.

2.1 Physical Principles of Remote Sensing

Remote sensing data consists of capturing the electromagnetic radiation emitted or reflected by surface objects. Electromagnetic radiation comprises of waves having different wavelengths. The electromagnetic spectrum (distribution of radiation by wavelength) is subdivided into different categories grouping radiation into different wavelength ranges. Figure 2.1 shows the full electromagnetic spectrum and the various uses for each range of wavelengths.

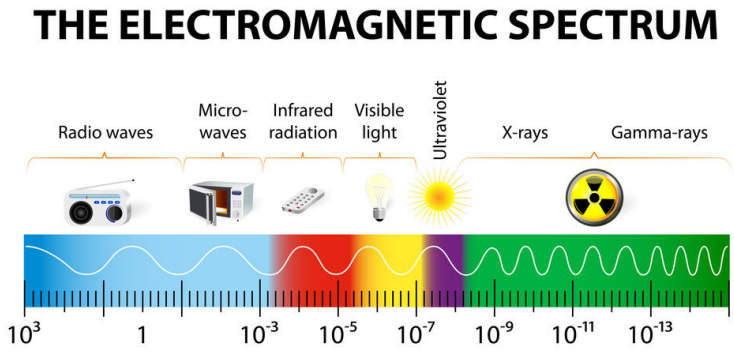

Figure 2.1: Electromagnetic spectrum (Image credit: `https://www.nist.gov/image/designuassspectrum154169990hrjpg`).

The Sun provides the Earth with radiation spanning the full electromagnetic spectrum. This radiation is partially reflected by surface objects or absorbed to

be emitted as thermal radiation and is at the basis of remote sensing. The most commonly sensed bands are the red, green and blue (RGB) bands which produce visible colors. These bands are essential for vision and photography and are also widely sought in remote sensing. RS sensors are also designed to capture light outside the visible spectrum. When the imaging product is composed of different color bands, it is referred to as a multispectral (MS) image. RGB is a type of MS imagery having only visible bands. It is common to additionally acquire the Red Edge and Near Infrared (NIR) bands. When bands are sampled continuously over a narrow range (such as 10 - 20 nm), we obtain hyperspectral (HS) images. These images have hundreds or thousands of spectral bands and therefore offer much more spectral information than MS images but at the cost of signal-to-noise ratio (SNR) which results in lower spatial resolution. Another type of RS imagery is the panchromatic image which is acquired using a sensor that is highly sensitive to a large number of bands often including the visible spectrum. These images have a high SNR and therefore a very high spatial resolution. This sensor however, does not register different bands separately like MS sensors, it records the overall energy reflected by each point and therefore each pixel value represents the overall brightness of that point. As such, panchromatic images are grayscale [47]. Pan-sharpening is an image fusion technique where a panchromatic image is fused with a multispectral image in order to obtain a very high resolution colored image. The resulting image is named a pan-sharpened image [48].

Some sensors emit their own electromagnetic energy and capture the portion that is reflected back. These systems can provide better observations as they can be tailored towards a specific application (such as atmospheric aerosol content) and because the radiation used has a high atmospheric penetration ability (clouds or fog) [21]. They can also be used to obtain 3D data [49]. Some of the most notable active sensor systems are radio detection and ranging (Radar), light detection and ranging (LiDAR) and synthetic-aperture radars (SAR) [49].

Different objects reflect light differently. In the case of RGB, the difference is observed as a difference in color. Similarly, objects reflect non-visible light differently. Thus, augmenting remote sensing images with more spectral bands allows to detect more discriminatory properties and characteristics of the targeted surface objects. For instance, thermal infrared radiation (TIR) can be used to measure land surface temperature [50], the near infrared band is widely used in vegetation monitoring applications as it can be used to calculate the normalized difference vegetation index (NDVI) which indicates the health of a plant [51].

2.2 Technology of Remote Sensing

Mankind has always been interested in obtaining a bird's-eye view of the environment and the planet in general. The earliest attempts at remote sensing were carried using hot air balloons, cameras attached to pigeons or simply by stand-

ing on a natural or artificial high altitude spot (hilltop, tower) [52]. Nowadays, geospatial sensors are installed on various types of airborne vehicles, most notably orbital satellites. There is a variety of satellites orbiting the Earth and collecting large amounts of images. These satellites differ by the on-board sensors (spectral bands, active/passive), the ground sampling distance (GSD), swath width and revisit period [49]. Also, the data from some of these satellites is freely available such as Landsat, Sentinel and MODIS, however this comes at the cost of having long revisit periods and/or comparatively low spatial resolution. Other satellites are operated by commercial firms and have a higher image resolution than free satellites. These satellites, such as WorldView, do not have a fixed periodic trajectory, but are commissioned to survey a given region at a given time. Other emerging airborne platforms include unmanned aerial vehicles (UAV), helicopters and other aircrafts. The main advantage of these aircrafts and most notably UAVs are lower operational cost, high spatial resolution, more frequent revisit times and flying at lower altitudes thus avoiding atmospheric interference [53]. However, they still cannot cover areas as large as the areas covered by satellites.

2.3 Applications of Remote Sensing

In order to extract useful information from remote sensing data, a number of RS image processing and analysis procedures are applied. These techniques include image fusion, image registration, change detection, scene classification, object detection, land use land cover (LULC) classification and segmentation. Some of these procedures are generally used as preprocessing steps to prepare the data for other applications (fusion, registration, segmentation) while the others are directly used to extract information for a specific application [54]. The use cases of RS are quite numerous and relate to different fields such as urban monitoring [55], climate monitoring [56], disaster response [21], precision agriculture [51], archaeology [57] and criminology [21].

2.4 General Background on Change Detection

Change detection is the process of using two or more remotely sensed images of a certain region taken at different times to identify spatial changes that occurred in this region. It has wide applications such as environmental, urban and disaster monitoring. The information extracted from CD can be used to learn about the changes in an environment such as updating urban inventories, damage assessment [5] and disaster response [21].

2.4.1 Image Modalities in Change Detection.

CD has been carried using a variety of RS image types. These include passively sensed multispectral (including RGB), hyperspectral and panchromatic images as well as actively sensed SAR images. HS images offer the richest spectral information as they have a large number of spectral bands. However, this leads to an increased dimensionality. Furthermore, HS images suffer from low SNR and therefore low spatial resolution (blurred objects or vague boundaries). MS images seem to compromise on the shortcomings of HS images by sacrificing spectral information, they are more easily acquired and are more readily accessible (Landsat, Sentinel) and they generally have a higher spatial resolution than HS images. Panchromatic images have the highest SNR and therefore a very high spatial resolution. Their only imitation is that they are monochromatic and lacking in spectral information. On the other hand, SAR images offer a distinct advantage of not being affected by atmospheric interference. However, they are plagued by speckle noise which can influence the output of the CD model.

In some cases, images from different types of sensors have been used in aggregation for CD. This is referred to as heterogeneous CD. Using images from different modalities can have the advantage of leading to a better differentiation. However, it also introduces some complications due to the discrepancies in the data between the different sensors (different resolutions and bands). It also requires the inclusion of an image registration step in the preprocessing pipeline to make sure that image pairs are correctly matched. Shi et al. [24] have compiled a comprehensive background on change detection in their survey.

2.4.2 Classical Methods for Change Detection

We identify classical CD methods as approaches that do not employ machine learning. The most common classical CD methods are algebra based and transform based methods [23, 24].

Algebra-based CD methods consist of performing mathematical operations on images to obtain the change map. Xiong et al. [58] have proposed a change thresholding method following an image differencing approach applied on SAR images. They computed the likelihood ratio using the joint probability distribution functions (PDF) of the two (pre and post) images. Assuming the PDFs follow a gamma distribution, the single peak of the distribution was used to select an appropriate threshold. This method however only works if the data follows a gamma distribution. He et al. [59] proposed a dynamic difference image (DI) thresholding method. The DI was first produced using change vector analysis (CVA) (a method which represents the change between two bi-temporal pixels as a vector with a magnitude and an angle [60]). Then the statistical parameters of the DI were computed using Expectation Maximum algorithm assuming a Gaussian Mixture model. The threshold was initialized using Bayesian decision

and the the dynamic threshold computed using Fuzzy c-means. Liu et al. [61] proposed an efficient method to detect small variations in HS images by using an iterative approach to CVA. Ke et al. [62] proposed a probabilistic significance test to threshold a DI taking into consideration the pixel neighborhood.

Transform-based methods are mainly concerned with reducing the dimensionality of images by removing redundant and highly correlated data while keeping only the data with the most variance and information. These transforms include principal component analysis (PCA) [63], tasseled cap transform [64], Gramm-Shmidt transform and Chi-square transform [65] and discrete wavelet transform [66]. Homogeneous pixel transformation is another transform that was used to transfer both pre and post images to the same feature space to make it easier to use them for CD [67].

There are many other classical CD methods such as reflectance models, spectral mixture models and GIS-based models [24]. A comprehensive review of classical CD methods was done by Asokan and Anitha [23]. While classical CD methods have proven to be successful on multiple occasions, they are generally sensitive to the threshold selection strategy and can be affected by noise in the images (from difference in sensors or from natural sources like fog, clouds and sun angle) [23, 25].

2.4.3 AI for Change Detection

In recent years, the change detection community has adopted AI-based solutions in order to implement efficient methodologies that do not suffer from the drawbacks of the classical methods. More so, deep learning methods have been the most predominant among these owing to their success in computer vision. We therefore distinguish deep learning techniques from shallow learning techniques. For a more comprehensive review of CD and AI, the reader is referred to the surveys carried by Shi et al. [24] and Khelifi and Mignotte [25].

Among the many shallow learning approaches, decision trees and support vector machines (SVM) have been used for change detection. SVMs and tree-based models are the most popular shallow models in RS applications [68, 69]. Im and Jensen [70] computed a neighborhood correlation image by defining pixel neighborhoods of fixed size (they experimented with different sizes) and calculating the piecewise correlation between the pre and post images. They used these then with a decision tree classifier to obtain the change map. Nemmour and Chibani [71] used SVMs in the context of land cover CD to detect urban expansion and they obtained better results than those obtained using a neural network.

CNNs have lead to the revolution of many computer vision applications. Aptly, CNNs are currently the most used architecture for CD [25]. Khan et al. [6] cast the CD problem as an object detection problem. Since their dataset spanned 29 years, some samples had missing reflectance information so they devised an inpainting method that takes into account surrounding pixels from nearby tem-

poral samples. They generated bounding box proposals using EdgeBoxes [72] and used a CNN to predict whether each patch belongs to the changed class or not. Cao et al. [18] used transfer learning for land use change analysis. They experimented with several pre-trained Imagenet [73] CNNs and fine tuned them on an open LULC dataset. Then they used the best performing model, a GoogleNet [74] with an SVM classifier, to detect the expansion of residential areas in the study area. Peng et al. [27] approached CD as a semantic segmentation problem. They used the encoder-decoder network UNet++ [75] with deep supervision. As an input to the network, they also concatenated the bi-temporal images. They tested their formulation on an open dataset of an urban setting and outperformed other related works, especially concerning small and obscure changes.

Zhan et al. [28] used a Siamese CNN for CD. They used a weighted contrastive loss function to train the network which resulted in having feature vectors distant from each other if there is a change. Therefore, the CM was constructed based on a simple distance measure and thresholding. They also added a K-nearest neighbors (KNN) classifier to further refine the results. Daudt et al. [76] compiled and annotated a dataset for urban growth change detection using MS images from Sentinel 2. They trained two deep models on their dataset. The first was a single stream CNN that concatenates the two images on input and produces a Softmax prediction at the output. The second was a double stream Siamese CNN, the output of the two streams was concatenated and classified by a fully-connected network. They showed that the first architecture tends to perform better and that adding more MS color channels improved the detection.

Lyu et al. [77], used a long short-term memory (LSTM) network as opposed to a CNN. In this framework, the pixel vector for the pre image was fed to the input layer which computes the hidden state. The hidden layer then receives the pixel vector of the post image along with the hidden state and outputs the change decision. This approach was shown to transfer well to multi-class scenarios and for inference on new images. Mou et al. [29] took this further by combining RNN with CNN. At each time step, a CNN was used to extract feature maps representing the images which were fed to a hidden RNN layer. They experimented with vanilla RNN, LSTM and gated recurrent networks (GRU) and obtained competitive results with the LSTM-based model. Chen et al. [78] used a multi-layer LSTM coupled with a Siamese CNN. First, the Siamese network was used to extract features from the bi-temporal images. These were then inputted sequentially to the multi-layer LSTM. This step replaced stacking or subtracting the feature maps by a learnable model. Finally, a fully connected network was used to classify the output. This method was shown to excel in comparison with SOTA especially with heterogeneous data. Jing et al. [79] introduced a Tri-Siamese network where each stream of the Siamese architectures receives three inputs: one image (pre or post) but at three different scales. They showed that this methods outperforms other SOTA and that the introduction of the Tri-Siamese aspect was their most impactful contribution.

2.4.4 AI for Building Damage Change Detection

Building damage detection is a specific case of change detection where the objective is to localize damaged buildings and assess the damage severity in the wake of a disaster. This information is important to help plan rescue and relief in the short term and repair operations in the longer term.

Kalantar et al. [33] experimented with three CNN models for post-earthquake building damage detection. The first is a single stream CNN for which the pre and post RGB images are stacked. The second is a Siamese CNN whose two outputs are concatenated. The third is similar to the Siamese CNN however the two streams do not share parameters. Their experiments showed that the Siamese model outperforms the others. Jiang et al. [32] introduced a pyramid feature-based attention guided Siamese network for building damage detection. The network consists of a Siamese VGG16 [80] feature extractor. They proposed a novel co-attention model that takes the feature maps of the two input images and outputs a correlation map. Areas with high temporal correlation are likely to be unchanged. This was followed by a pyramid-based decoder which iteratively merges the features across the layers while up-sampling the output at each layer until it recovers the original image size at the output. Their approach achieved SOTA results on the WHU dataset [81] and on their own dataset. Ji et al. [30] proposed a framework for detecting building damage using automatically generated positive samples (since they are usually scarce). They first extracted building footprints using a segmenter, then the bi-temporal segmented images were fed into the change detection network. To generate more changed samples, they simulated changes in the binary building maps by reduction. Also, they simulated co-registration errors by shifting building footprints between the bi-temporal images. This had a regularizing effect and made the model more robust to errors or noise in the data. They obtained adequate scores on the WHU Building dataset [81]. Most importantly they fared well even when training on only simulated data. The performance improved as they added more real data. Wang et al. [82] addressed the CD as an object detection task rather than a segmentation task. Khan et al. [6] argued that this approach has some advantages over the segmentation-based approach. They used Faster Region-based CNN (Faster R-CNN) [83] for change detection in high resolution images in a from-to scheme rather than binary (from vegetation to building). They tried two different image fusion methods: concatenation and differencing. They found that the differencing technique worked better outperforming other baselines. Wheeler and Karimi [35] used a CNN-based model to classify disaster damage level in the xBD dataset [3]. They used a pre-trained UNet model to generate building footprint polygons, these polygons were then used to extract sub-images that were passed to a CNN classifier. The models they tested were based on transfer learning from several Imagenet [73] pre-trained models. They also incorporated a class weighing scheme in their loss function because the "no damage" class greatly outnumbered

the other classes. Weber and Kané [84] experimented with different data processing and model training approaches for building damage assessment on the xBD dataset. They used a shared weight dual stream ResNet50 [85] backbone with a feature pyramid network for feature extraction. The features were then fed to segmentation and damage assessment model. Dividing each input image into four crops, using the same network for both building localization and damage assessment and weighing the loss function inversely to the occurrence of the classes provided a significant performance improvement. Su et al. [86] provided a review of the techniques used for building damage detection and used the xBD dataset to perform an in depth assessment of technical and operational problems that hinder progress in building damage detection. They identified an inconsistency and non-uniformity among evaluation metrics used because of the various model types used (object level and pixel level). They proposed a novel metric and a way to convert between the different metrics. They suggested training models relying solely on post images is case pre images were not available. Another suggestion was generating pre-images using generative adversarial networks. They also identified class imbalance as being a major issue in this field and proposed various re-sampling techniques. Weight sharing in Siamese architectures, network pruning and knowledge distillation were discussed as approaches for reducing the computational needs while still maintaining a good performance.

2.4.5 Methodologies for Fast Change Detection

We have highlighted the importance of obtaining predictions as quickly as possible in the aftermath of a disaster. There is a large and growing body of work that aims to address this problem. We have identified two main strategies used to fulfill this requirement: pre-trained models that transfer from one disaster/region to another allowing them to be used to infer on new disasters without training and semi-supervised learning which allows a model to be trained with few labeled samples thus reducing the amount of time needed for data labeling.

Cross-Domain Transfer for Disaster Damage Detection

Nex et al. [31] acknowledged the effectiveness of CNN-based solutions for CD and they tested what they can deliver from an operational and first responder perspective for post-disaster damage detection. They assessed model transferability and time cost. They also proposed their own CNN model which uses dense connections and dilated convolutions. They found that the transferability is highly dependent on the original training data being similar to the target data. They also proposed that time cost can be reduced by eliminating some tiles with irrelevant content (vegetation, water...). Miura et al. [34] also used a CNN with a similar framework. They tested it on a dataset of typhoon-induced damage. Benson and Ecker [43] focused on the ability of change detection models

to generalize into different disaster types. Since the xBD dataset is by default an IID (independent and identically distributed) set, it was not suitable for assessing the cross-domain generalization ability of a model. Therefore, they created their own split which provided ample data for training while testing on entirely different disasters. They experimented with two different models on both the IID set and their own set and noted the generalization gap. They proposed multi-domain adaptive batch normalization and stochastic weight averaging to mitigate this gap in model generalization. Bai et al. [87] studied the effectiveness of a Semi-Siamese model with a pyramid poling module and an attention mechanism. They showed good performance on the xBD dataset. To further demonstrate the robustness of their model, they showed its effectiveness in predicting on an additional dataset. Gupta and Shah [88] introduced RescueNet, a unified framework that localizes buildings and provides damage classification. They outperformed the xBD baseline [3]. To demonstrate the domain-shift generalization, they trained the model on the Tier 1 data and tested on the Tier 3 data. The model transferred well in terms of localization but less so in terms of damage classification. Yang et al. [44] assessed the generalization ability of four CNN models. They first trained these models on data from the xBD dataset following a variety of training and fine-tuning schemes. They then tested their on earthquake satellite imagery taken in a different geographical regions and on aerial images. Their findings highlighted that the models transferred poorly when tested on aerial images due to the difference in image sensors. However, this performance was improved when a small number of images from the aerial dataset were included with the training data. Zheng et al. [45] aimed to address the issues with object-based image analysis with patch-based CNNs and fully convolutional Siamese networks. The former has good semantic consistency but weak feature representation while the latter is the opposite. Therefore, they proposed a deep object-based semantic change detection framework integrating the two aforementioned frameworks to mitigate their weaknesses. They trained their model on data from the xBD dataset and tested its performance on the Beirut Port and Bata barracks explosions to demonstrate its performance and generalization ability. They highlighted that being an end-to-end trainable model was the most important improvement they introduced.

Semi-supervised Learning for Change Detection

Pati et al. [89] developed a hybrid approach combining supervised and unsupervised learning. They proposed a feature extraction algorithm that takes into consideration neighboring pixels. For labeling the data they used a hybrid unsupervised model based on clustering and a fuzzy logic membership function. Then they used these labels to train an SVM and tune its hyperparameters using genetic optimization. Sublime and Kalinicheva [15] adopted an unsupervised approach to detect damage caused by a tsunami. They used a deep autoencoder

(AE) with the assumption that the AE will easily learn trivial changes (caused by weather, illumination...) while the disaster-induced changes (which are of interest) will be considered as outliers and having high reconstruction error. They used this information to construct a CM of the non-trivial changes. Finally, they used clustering to separate the non-trivial changes into different types (flood, damaged building...). This approach achieved comparable results to supervised methods. Since the majority of samples in building damage CD are negative samples (no damage), Tilon et al. [40] cast this problem into an anomaly detection problem with the damaged cases considered as anomalies. They used an anomaly detecting generative adversarial network. The main advantage of this approach is that it was only trained on pre-disaster imagery and therefore can be prepared before a disaster occurs. They tested their method on both the xBD dataset and their own UAV dataset. They also evaluated the generalization of their model by testing on different disaster types and regions and they noted that it decreased when the test set is contextually different than the training set. The main disadvantage of this approach is that it can only provide binary damage classification. Peng et al. [41] proposed a semi-supervised framework based on a UNet++ segmentation network and generative adversarial training. The full set of both labeled and unlabeled images was fed into the segmentation network which produces binary change maps. The labeled subset is used to train the segmenter in a supervised manner. The outputs of unlabeled images along with the ground truth of the labeled images were used to train a discriminator intended to differentiate between output from unlabeled images or ground truth. This discriminator was trained in an adversarial manner with the segmenter thus making the unlabeled outputs more similar to ground truth. A second discriminator was similarly trained on the entropy maps of labeled and unlabeled outputs. They tested their approach on both a public dataset as well as their own dataset and showed that they achieve SOTA performance.

In this paragraph, we have reviewed several approaches that can accelerate the process of obtaining predictions for new disasters. We note that the majority of the work is focused around providing pre-trained models that can be used for direct inference. However, the generalization of these models has been reported to be effective to varying degrees and oftentimes hinged on how similar the test data is to the training data and could be improved by retraining on some samples of the new data. It is also subject to degradation due to data heterogeneity. This motivates the use of semi-supervised learning since it is built for each disaster individually. Also, it is arguably easier to develop since pre-trained models need to be trained on large and diverse data to maximize their effectiveness which could be time and resource consuming.

2.5 Graph Convolutional Networks

2.5.1 Concepts Behind Graph Convolutions

Graphs are a non-linear data structure used to represent complex relationships among entities of the dataset. We formally define a graph $G = (\nu, \varepsilon)$ by a set of N nodes (vertices) ν connected by a set of edges ε with each node characterised by a feature vector of size F. Node edges can also be weighted or can simply represent a binary connection. Figure 2.2 shows an example of a classical social circle problem represented by a graph.

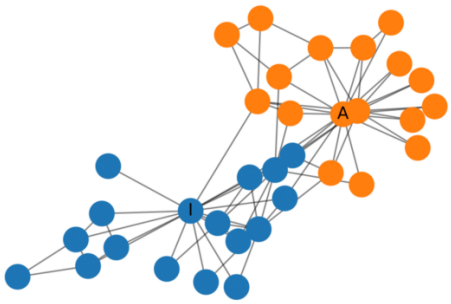

Figure 2.2: Zachary's Karate Club [1], a common social relation problem represented by a graph. Each node is a student and the edges represent their social relations. Different colors represent how the students split into two factions after a conflict between the club owner and their coach.

Most machine learning models can only deal with euclidean data which is data that is highly structured and can be represented in a flat space. For example, CNNs are extremely efficient at exploiting the stationary structure of images (a two-dimensional array). However, many real world problems naturally fall into a graph representation such as social relations, molecular chemistry, recommender systems and traffic prediction. Also, many types of data that are naturally euclidean can be cast as graphs which enables a more global and wholesome representation of the data. Graph neural networks have thus emerged to fill this gap [90, 2].

Graph convolutional networks are currently the most popular type of graph neural networks, they generalize the concept of convolution into non-euclidean space by extracting features using signals coming from neighboring nodes. Figure 2.3 shows a high level illustration of conventional and graph convolutions.

In general, the purpose of graph convolutions is to produce node embeddings which are subsequently used for different downstream tasks such as graph classification, node classification, and graph clustering. For a given graph, the node

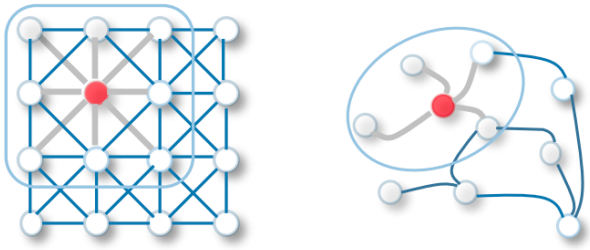

Figure 2.3: High level depiction of 2D convolution (left) and graph convolution (right) (Image credit: [2]).

embeddings Z are of shape $N \times C$ where C is the number of node output features. GCNs are divided into two categories: spectral GCNs and spatial GCNs. Spectral GCNs rely on the graph Fourier transform (similar to the Fourier transform in signals). Graph convolutions are obtained by multiplying two graph signals in the spectral domain and doing the inverse Fourier transform (a principal well established in the field of signal processing) [37]. However, the computation of the graph Fourier transform is relatively expensive. Therefore, a first order approximation of this process is proposed in spatial GCNs [36] which makes the process more analogous to the forward pass in a traditional neural network.

In spatial graph convolutions, the graph G is represented by a feature matrix X of size $N \times F$ where each row i is the F-dimensional feature vector of node i and an adjacency matrix A of size $N \times N$. The adjacency matrix denotes the edges in the graph. For each pair of nodes i and j, the value of A_{ij} represents the connection between these two nodes. A value of 0 denotes the absence of an edge and a value of 1 denotes its presence. In the case of weighted edges, the value of 1 is replaced by the edge weight value. In general, a graph convolutional layer (l) of a graph neural network of L layers can be defined according to Equation 2.1:

$$H^{(l+1)} = f(H^{(l)}, A) \tag{2.1}$$

For the input layer, $H^{(0)} = X$ and for the output layer $H^{(L)} = Z$. the mapping function f is the graph convolution. Most graph convolutional algorithms differ only by the definition of f. Initially, this function is defined by Kipf and Welling [36] according to equation 2.2 where $W^{(l)}$ is the matrix of learnable weights at layer (l) and σ is a non-linear activation function such as ReLu (rectified linear unit).

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)}) \tag{2.2}$$

This propagation rule implies that for each node, we compute the elementwise sum of the features of the nodes that are connected to it which is then propagated to the next layer $(l + 1)$ through the weight matrix $W^{(l)}$. However, this does

not take into account the features of the node itself unless the graph has self-loops. This is corrected by adding the identity matrix I to A thus enforcing self-loops. Another proposed modification is the symmetric normalization of A using its degree matrix D. The outcome is the following updated propagation rule (Equation 2.3).

$$f(H^{(l)}, A) = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{2.3}$$

Where $\hat{A} = A + I$ and $\hat{D}$ is the degree matrix of $\hat{A}$. This proposed formulation has mostly been applied in transductive settings such as semi-supervised learning. The major shortcomings of this model is that for transductive learning, the model needs to have access to the full graph which can be very memory consuming in certain applications. Also, this trained model cannot be used to generalize on unseen graphs or nodes.

To address these issues, a different graph convolution operator named SAGE was proposed by Hamilton et al. capable of learning in an inductive manner and inferring on new unseen data [46]. This method is based on a cycle of neighborhood sampling and aggregations to compute the new node embeddings. This cycle is run for K iterations. For a node ν, the embeddings are noted by h_ν^k. Initially, the embeddings h_ν^0 are equal to the node features.Then for each node $v \in \nu$, the features of its neighborhood $N_{(v)}$ are aggregated according to the notation in Equation 2.4.

$$h_{N_{(v)}}^k = AGGREGATE_k(h_u^{k-1}, \forall u \in N_{(v)}) \tag{2.4}$$

Where $AGGREGATE$ is a differentiable permutation invariant aggregation function. The authors [46] experiment with three types of aggregators: mean, max pooling and LSTM. In most cases, the max pooling aggregation is found to be effective. In this approach, the feature vector of each neighbor $u \in N_{(v)}$ is fed through a fully connected neural network layer and then the resulting vectors are aggregated into a single vector using an elementwise max operation. This process is show in Equation 2.5.

$$AGGREGATE_k^{pool} = max(\sigma(W^{pool}h_u^k + b), \forall u \in N_{(v)}) \tag{2.5}$$

The embedding of node v is then updated using the aggregated features of its neighbors following Equation 2.6.

$$h_v^k = \sigma(W^k.CONCAT(h_v^{k-1}, h_{N_{(v)}}^k)) \tag{2.6}$$

Where σ is a nonlinearity, W^k is a matrix of learnable parameters and $CONCAT$ is a concatenation operation. This process is repeated K times to obtain the final node embeddings. Effectively, K determines the number of neighborhood hops used to aggregate features for each node. Using this sampling and aggregation strategy, node embeddings can be learned using small subgraphs rather than

the entire graph. The size of the subgraphs is dictated by the number of edge traversals K.

2.5.2 Graph Convolutional Networks in Computer Vision

The representation of images as graph structures has enabled the use of GCNs to solve a number of computer vision problems such as image classification and object detection. Chauduri et al. [38] used a Siamese GCN for remote sensing image retrieval. First, the two images were segmented into different regions. Then a feature vector for each segment was constructed using handcrafted shape, color and texture features. In their graph representation, each segment is a node and nodes are connected if the segments are adjacent. The edges were also weighted according to the pairwise centroid pixel and segment orientation difference. They trained the Siamese GCN using the contrastive loss and used the Euclidean distance on its output to get image retrievals. Chen et al. [91] used GCNs to improve the process of multi-label classification in images. While the co-occurring classes were treated independently, they built a directed graph that maps the logical relationships between the different classes. This graph was used to train a stack of GCNs to learn a set of classifiers which were then combined with the output of a CNN (feature map) to obtain the final classification. By inspecting the learned weights of the graph, they showed that these weights represented logical conditional probabilities between the objects. Khan et al. [92] extended their previous work [38] for multi-label RS image classification. They used a multi-label output GCN instead of a Siamese GCN. Mou et al. [93] exploited the semi-supervised nature of GCNs and the possibility of learning from image-wide relationships using non-local graph representation for LULC classification in hyperspectral imagery. In their graph formulation, each pixel was considered to be a node and the similarity between every pair of nodes was measured to build the edges indicating the likelihood of the two nodes belonging to the same class. They trained a semi-supervised GCN and showed its effectiveness on different datasets. They noted a major shortcoming of this method being the high computational resource demand. Hong et al. [37] developed a GCN variant for HS image classification. Their variant, called miniGCN, can be trained using mini-batches of data (like traditional neural networks) and can be used to infer on new data. In their graph model, each pixel was a node and edges were a measure of similarity or dissimilarity between pixels. They experimented with miniGCN as well as miniGCN combined with a GCN throught different output fusion schemes. They outperformed other works using their concatenation-based fusion network.

2.6 State-of-the-Art Work

In this chapter, we have reviewed works related to change detection and GCNs as these are all related to the problem we are targeting and/or to our formulated solution. In this paragraph, we regard the works which we identified as being state-of-the-art in relation to this work.

In general, it was difficult to select a prior work that met all the criteria to be considered SOTA mainly for two reasons: 1) the lack of a unified benchmark dataset as most works tend to use their own data; 2) the non-uniformity of evaluation metrics across different works.

The first SOTA work considered here it that of Saha et al. [39] which was the first to use a semi-supervised GCN for change detection, largely motivated by the difficulty of obtaining a large number of labeled samples. They used pre and post VHR satellite images of the area of interest where a few samples were labeled as changed or not changed. A superpixel segmentation algorithm was used to segment the image into multitemporal parcels at different resolution levels. The graph was constructed using the parcels at the base resolution level as nodes. The parcel segmentations at higher resolution levels were only used to compute node features (mean, maximum and minimum spectral values and the area at all the segmentation resolution levels)r. The graph connectivity was represented by the adjacency matrix. Any two parcels that did not share an adjacent pixel were not connected. Otherwise, the edge weight was calculated according to a node feature similarity measure. For the model architecture they used the GCN operator as defined by Kipf and Welling [36] to construct a multi-layer network. They trained and tested their approach for a generic binary urban change detection for an area in Trento, Italy. They experimented with different hyperparameters and outperformed other semi-supervised and unsupervised approaches.

Concerning the domain transfer of building damage change detection models, Xu et al. [42] worked on different earthquake cases. They collected three earthquake datasets (Haiti, Mexico City and Indonesia). The only image preprocessing done was histogram equalization. The labels were obtained from the UNOSAT program on a five level damage scale similarly to the xBD annotations. However, to avoid ambiguity, the authors merged the "Severe" and "Destroyed" classes. Since the UNOSAT data only provided annotations for damaged buildings, the authors used a Faster RCNN building footprint detector to add negative (non damaged) samples to the dataset. Finally, they cropped the images into smaller patches for both the pre and the post images. They compared four different CNN models. The four architectures only differed in how the input is provided: concatenating the pre and post images, using only the post image, a Siamese architecture with concatenation of the feature maps and a Siamese architecture with the difference of the feature maps instead of their concatenation. They first compared the four models on the Haiti data and found that the Siamese subtraction model outperformed the others even though theoretically the Siamese

concatenation model should have fared better since it is more general. They then proceeded to test the generalization of the Siamese subtraction model by training it on one set of regions and testing on another. The best generalization was achieved when a small percentage of samples was leaked from the test set into the training data.

In summary, Saha et al. [39] introduced GCNs to the field of change detection and showed how this problem can be cast into a semi-supervised graph node classification. One important shortcoming of this approach is that for GCNs to be trained under a transductive setting, the entire graph must be loaded into memory. In the case of satellite imagery, this can reach unscalable levels of memory consumption. It is worth noting that the authors of the above work avoided this bottleneck by using handcrafted summary features of the node segments. However, this eliminates the opportunity of using the full image pixels to learn important features as is done in traditional CNN based frameworks. On the other hand, Xu et al. [42] proposed a systematic framework for assessing the generalization and domain shift robustness of a model which is a necessary test to perform on any disaster related change detection solution. One notable gap in their analysis is that the studied domain shift is exclusively related to a change in the disaster location and not in the disaster type.

In this work, we present a graph node classification building disaster damage detection framework. Our formulation is inspired by that of Saha et al. [39] but we take it further by modifying the way the graph is built by learning deep pixel feature maps as node features instead of calculating summary features as it is done in traditional CNN frameworks. We also use a geometric triangulation to define the edges to ensure that every building is connected to all of its surrounding neighbors instead of only connecting adjacent patches. We adopt both an inductive learning setting and a transductive setting. The former being suitable for semi-supervised learning which needs much fewer labeled samples to converge and the latter being more suitable when seeking a solution that can transfer to other data domains. We test the transferability by adopting a similar experimental setup to that of Xu et al. [42] but using the xBD dataset but our assessment also takes into account transferring to different disaster types and not just different regions with the same disaster. Benson and Ecker [43] also provided a robust experimental setup for testing model domain shift effectiveness using the xBD dataset. However, we adopted that of Xu et al. [42] because it allows for more flexibility regarding which disasters to choose as training and target subsets and regarding the size of each subset. Furthermore, we embed contextual building meta-features which introduce additional data for learning based on domain knowledge. None of the previous works rely on anything other than image data. Table 2.1 summarizes how the work presented here is different from the SOTA literature.

Table 2.1: Qualitative Comparison with State-of-the-art Work.

Work	Building Damage	Uses GCN	Semi-supervised	Domain Transfer	Uses Meta-features
Xu et al. [42]	Yes	No	No	Yes	No
Saha et al. [39]	No	Yes	Yes	No	No
This Work	Yes	Yes	Yes	Yes	Yes

2.7 General Research Trends

Based on the literature reading we have done in this work, we draw several conclusions and note patterns and trends in research concerning the field of change detection. For this analysis, we discard works that were simply cited to define a term or to credit resources. As a result, we obtain a corpus of 83 publications. We divide these works into several subcategories based on the general topic (Figure 2.4). The majority of the works considered are naturally related to CD. We divide the different CD works by application. We focus on damage detection and disaster related applications since this is the case considered in this work. Urban CD works are those concerned with tracking urban expansion or LULC change while Environmental CD are works related to monitoring natural phenomena and environmental problems such as lakes and sea ice. Remote sensing subtopics were collected to provide the background on remote sensing and its applications at the beginning of this chapter. Works under the category Algorithms were the works that introduced an algorithm or a theory that was useful for this work such as the Semi-supervised GCN [36].

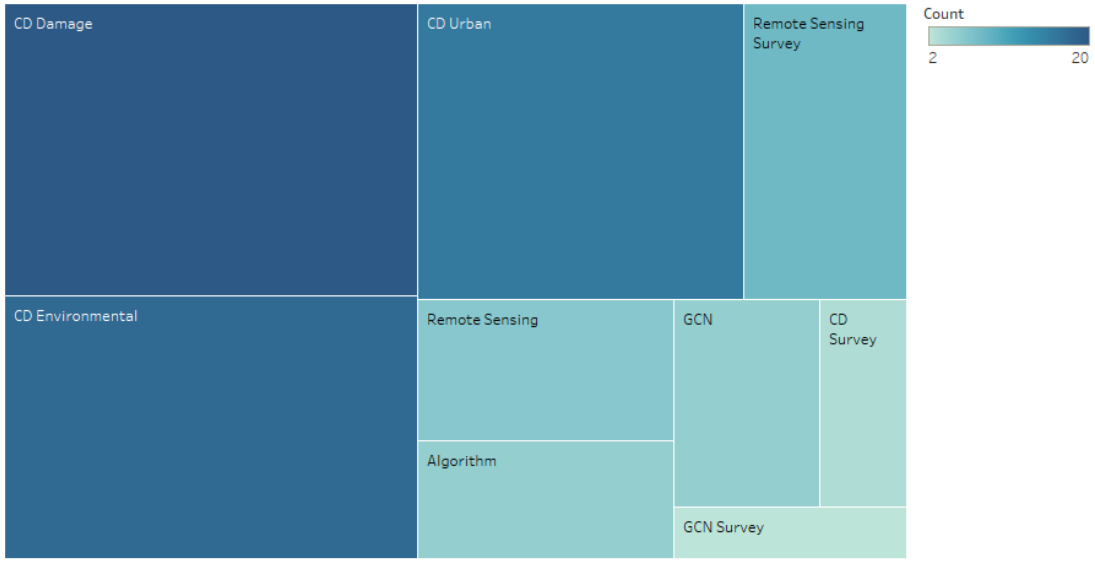

Figure 2.4: Different subcategories of the collected publications.

With only a few exceptions, the majority of the publications are journal articles which indicates the maturity of the research being done in this field. We

also note that most of them were concentrated in the last few years and have been increasing every year (Figure 2.5). This means that although CD is a long-standing field, interest in new solutions has not waned and the field is still being continuously improved.

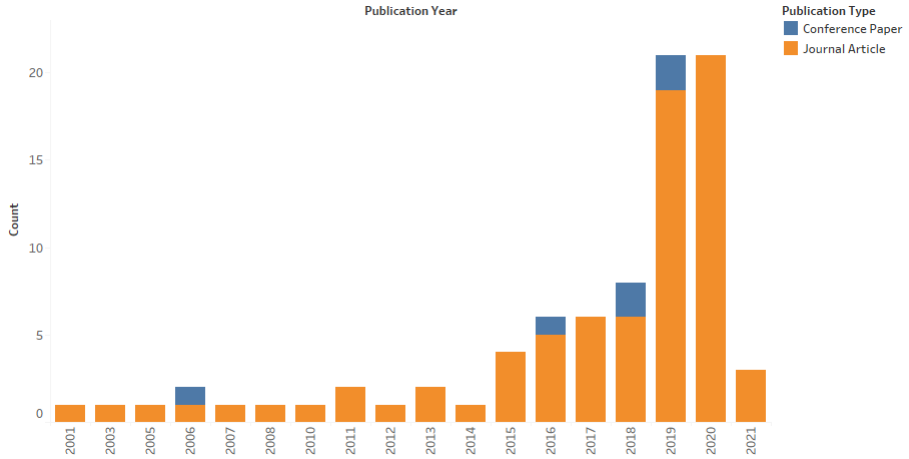

Figure 2.5: Publication year and venue type of the collected works.

In terms of publication venues, the majority of works were published in remote sensing themed journals instead of being published in machine learning venues (Figure 2.6) despite the solutions and experiments carried being heavily dominated by machine learning development and testing protocols.

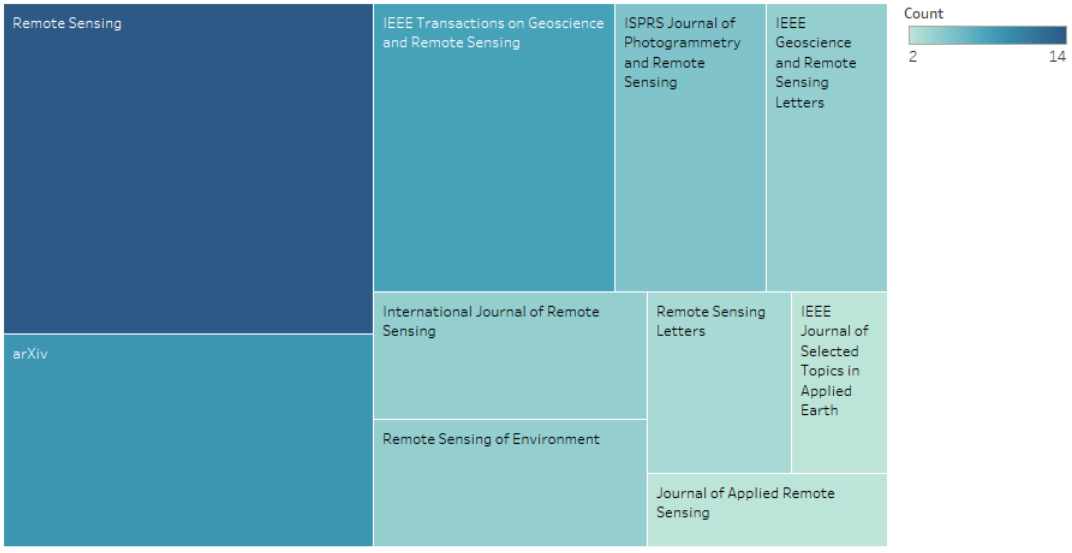

Figure 2.6: Top 10 publication venues of the reviewed literature.

Finally, we show in Figure 2.7 how the adopted CD approaches have changed over the years. We first note that traditional approaches such as classical CD

and shallow learning have seen a decrease in use and have all but disappeared in 2019. This was accompanied by the adoption of deep learning based approaches. Particularly, there was a sharp increase in CNN adoption in 2019 and it has been since the predominant family of models. Moreover, new architectures have appeared in 2020 and 2021 such as hybrid GAN-based models as well as GCN.

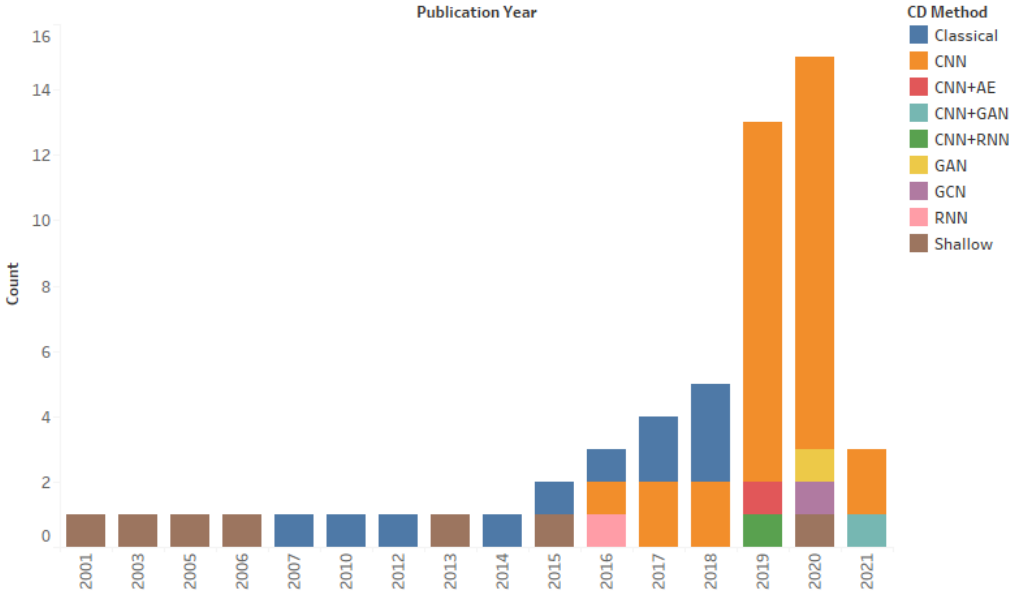

Figure 2.7: Evolution of CD methods over the years.

In summary, our review of related works shows that the field of change detection has been rapidly evolving in recent years. The amount of research interest in this field has risen sharply and new and sophisticated solutions are being developed. In this work, we proceed along this path by proposing our own method based on a GCN combined with a CNN that still retains the powerful capabilities of the CNN while overcoming the limitation of its locality. We also propose how the limitations of satellite-based solutions can be mitigated using contextual urban features.

Chapter 3

Methodology

The main contribution of this work is the introduction of a novel graph-based approach for building damage change detection. Below we present details about the graph-based data structure and the model architectures used and training configurations.

3.1 Graph Data Formulation

We propose a graph-based structure for presenting the data which allows our models to exploit both the local image features as well as their relationships with neighboring samples. This formulation assumes the existence of building footprint polygons which are available in the xBD dataset and were procured for our Beirut Port Case study from the Beirut Recovery Map project [1]. In case these footprints were not available, they can be obtained using a building footprint detector [35].

Our formulation builds an undirected acyclic graph using the building polygons. Each node in the graph represents one building. The node features are the concatenation of the pre and post image crops defined by the rectangular envelope of the building polygons. The crops are resized to a width and height of 128×128 in order to unify their size and reduce the memory footprint of the graph. The pixel values are normalized to a range between 0 and 1. The resulting feature vector is of size $N = 128 \times 128 \times 3 \times 2 = 98304$. To exploit domain knowledge information, we embed the meta-features at the node level which increases the dimension of the vector to $N = 98324$. The edges are constructed using a Delaunay triangulation [94] based on the building envelope centroid coordinates. Each edge is given a weight measuring the similarity between the connected nodes. This weight is calculated according to Equation 3.1 which was used by Saha et al. [39] to build their adjacency matrix. In Equation 3.1 below, $E_{i,j}$ is the weight of the edge connecting node i to node j and N is the number of elements in the

[1] https://openmaplebanon.org/beirut-recovery-map

feature vector F_i of node i which is equal to the number of pixels per RGB image multiplied by two due to the presence of both a pre and post image.

$$E_{i,j} = 1 - \frac{1}{N} \sum_{n=1}^{N} \frac{|F_i^n - F_j^n|}{|F_i^n| + |F_j^n|} \tag{3.1}$$

Figure 3.1 shows a conceptual sketch of the proposed graph formulation while Figure 3.2 shows a realized implementation on an example from the xBD Joplin Tornado set.

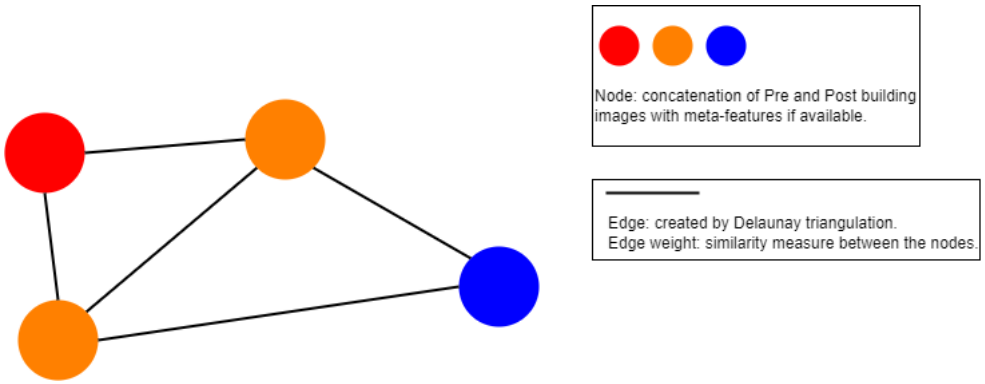

Figure 3.1: Concept of the Graph Formulation. Different node colors indicate nodes' different classes.

It is important to note the differences in constructing the graph when using the transductive semi-supervised approach as opposed to the inductive supervised approach. For the transductive case, the entirety of the data needs to be built into a single graph because the semi-supervised GCN, as is the case with semi-supervised models, needs to observe the entirety of the data during training. Only a relatively small number of samples is labeled during training and the model learns to propagate labels along the graph edges to the rest of nodes. However, this requires that the forward pass be made on the full dataset during training which requires heavy computational resources to hold the large graph in memory. In the inductive setting, training is done on batches of data which mitigates the memory issue of transductive graph learning. In the case of graphs, each batch is created by sampling a neighborhood of nodes from the full graph [46]. Based on the above, we exploit the natural organization of the xBD dataset to define suitable graph construction methods for both cases. For the transductive setting, we construct one large graph out of each region. In this case, the coordinates are the UTM (Universal Transverse Mercator) coordinates of the polygon envelope centroids. What we do differently is in regards to the inductive case. Normally, we would have built a large graph out of each region as we did for the transductive case and sampled subgraphs. However, the size of such a graph proved too large to hold even within CPU memory. The xBD dataset is already divided into

Figure 3.2: Subgraph implemented on a sample from the Joplin Tornado set.

image chips with each chip covering a certain geographical area and having a non uniform number of buildings. Therefore, we use this predefined organization to build a single graph out of each image chip. In this case, the coordinates are based on the pixel positions. Figure 3.2 is an example of an image chip that was used to create a subgraph.

3.2 Model Architecture

Using the proposed graph formulation, we use graph convolutions to aggregate signals coming from neighboring nodes. However, our preliminary experiments have shown that it is important to have a reliable way of extracting local features. Therefore, we present an architecture combining both a CNN and a GCN. The CNN backbone is used to extract local image features and the GCN operator is used to aggregate them with CNN features of connected nodes. The CNN backbone is a Siamese ResNet34 network with the classification layers removed [85].

The output of the two-stream network is the difference between the individual stream outputs which is piped into a graph convolutional network whose output is the classification. The ReLu activation function is used for all layers except the last layer which uses a Softmax activation. The ResNet weights are initialized with the ImageNet weights but are trained along with the entire model.

The same architecture is used for both the transductive and inductive cases, the only difference being the graph convolution operator used. For the transductive case, we use the semi-supervised graph convolutional operator by Kipf and Welling [36]. For the inductive case, we use the SAGE operator by Hamilton et al. [46] which was conceived to allow GCNs to be trained in an inductive manner using batches of data.

When introducing domain knowledge data, the images are fed through the ResNet backbone and the output is concatenated with the meta-features. The resulting vector is then inputted through the GCN layers. A similar meta-feature injection method was adopted in different applications [95, 96]. The diagram in Figure 3.3 shows the overall layout of the proposed architecture with or without meta-features.

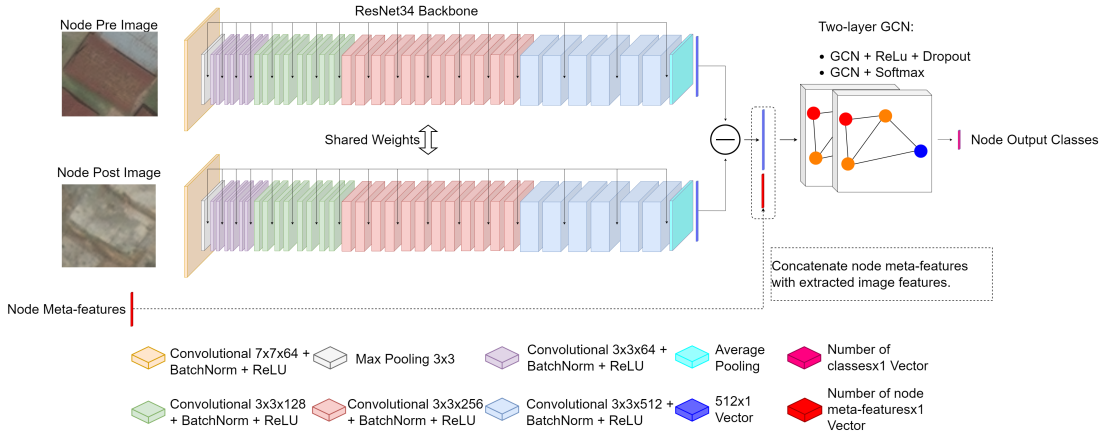

Figure 3.3: Architecture of the proposed model. Objects in dotted lines indicate meta-feature injection if these features are available.

Chapter 4

Experiments, Results and Discussion

In this chapter, we present the data that was used in this work along with the data preprocessing steps needed. We describe the experiments and procedures done to implement and evaluate our approach. We also show and analyse our results and discuss their significance and implications.

4.1 Datasets and Evaluation Metrics

The proposed approach is tested on both a public dataset to position this methodology with respect to the literature as well as the Beirut Port Explosion dataset which serves as a demonstration of the effectiveness of meta-feature injection for building disaster damage change detection.

4.1.1 xBD Dataset

The xBD dataset [3] has different cases of city damage for different disaster types (earthquake, flood, wildfire...). It contains a collection of manually annotated pre and post Worldview 2 images acquired from Maxar's open data program. The dataset is organized into two big sets. The Tier 1 set contains a collection of disasters divided into train, test and hold subsets. The Tier 3 set introduces additional disasters. This dataset was used for the xView2[1] competition.

At the time of writing, this dataset is the most diverse in terms of regions, disaster types, buildings and urban density. It contains image chips of earthquake, tsunami, flood, volcano, wildfire and tornado/hurricane disasters across sixteen regions as shown in Figure 4.1. The landscape in these different regions varies from an urban scene such as Mexico to largely rural regions such as Guatemala.

[1] `https://xview2.org/`

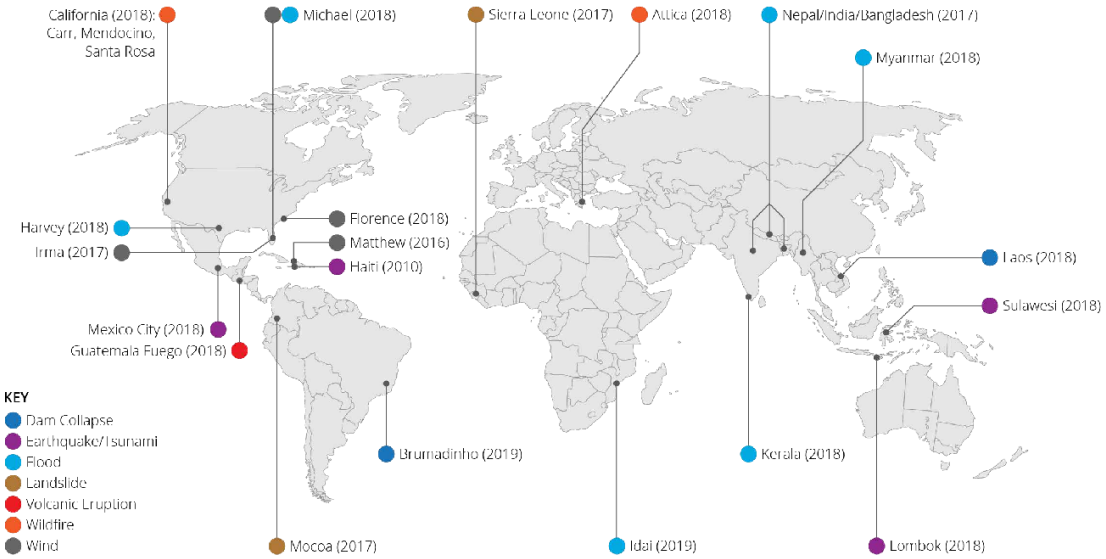

Figure 4.1: xBD disaster distribution across the regions (Image credit: [3])

The dataset is annotated according to a four-class joint damage scale: no damage, minor damage, major damage and destroyed. As with most disaster datasets, this dataset suffers from severe class imbalance as the majority of buildings are non-damaged. This was indicated by the authors and is further highlighted by exploring the number of individual buildings belonging to each class per region. This is shown in Figure for a subset (Tier 3) of the regions. On average, 80.4% of individual buildings across the Tier 3 regions are non-damaged (Fig. 4.2). Also, since each region is divided into equal image chips, we investigated the number of chips with at least one damaged instance and found that about 50% on average do not contain any damaged buildings (Fig. 4.3). This was important for formulating our data sampling strategy. Especially considering that many image chips had no damage at all and these could be undersampled. Dataset class balance visualizations for Tier 1 regions are provided in Appendix A.

4.1.2 Beirut Explosion Dataset

Beirut is an architectural mosaic with irregular and heterogeneous building patterns with many different layers of architectural styles added on top and next to each other over the decades. The architecture in this city spans many centuries and civilizations as it is home to Roman heritage sites and civil war (1975-1990) ruins. However, these structure are not located far from luxurious commercial and financial districts and modern residential high-rises. The architectural styles that dot the city are also varied and include Ottoman, European and Gothic patterns [97].

The port explosion occurred on August 4, 2020 resulting in the death of nearly

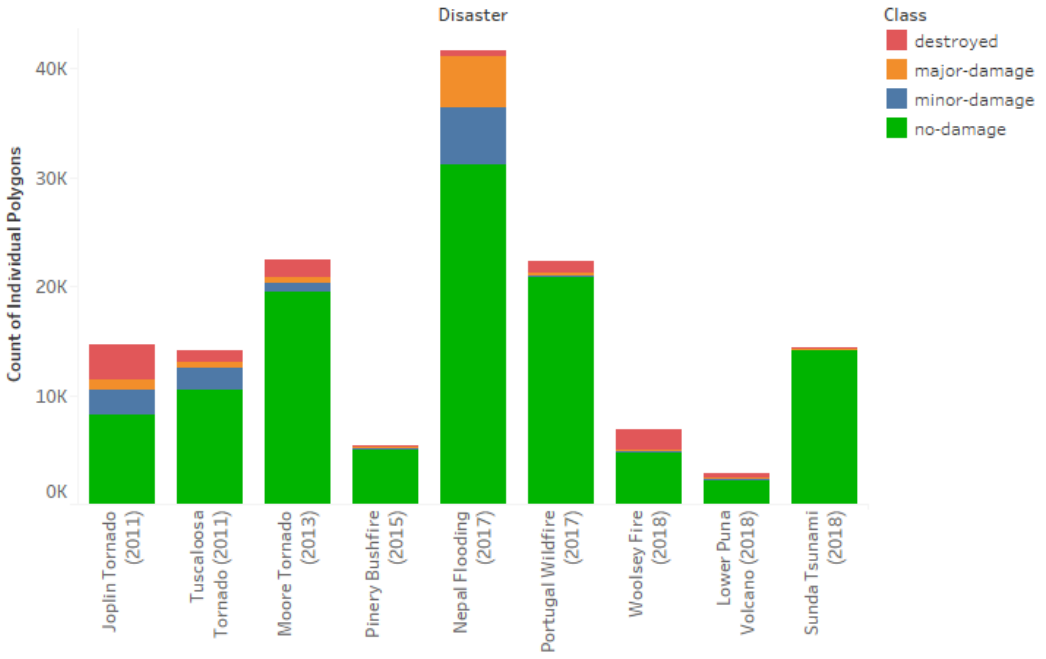

Figure 4.2: xBD Tier 3 number of buildings per class.

135 people, around 5000 injuries and causing severe structural damage to the surrounding area leaving more than 300000 people homeless [98]. The strength of the blast was estimated to be equivalent to the twentieth of the Hiroshima bomb and as the strongest non-nuclear explosion in the 21st century [99]. Worldview 2 satellite products covering a large portion of Beirut were donated by Maxar Technologies[2]. The first was taken on July 31 2020 and the second was taken on August 5 2020. The damage classification was provided by the Open Map Lebanon team who curated these annotations for their Beirut Recovery Map project[3]. These annotations use a four class damage scale: minor, moderate, major and severe damage. Furthermore, the heterogeneous and irregular urban and architectural landscape of Beirut motivated us to use of contextual meta-features to enable our model to learn the patterns and interactions resulting from this building non-uniformity. This dataset were obtained from Krayem et al. [100].

The images needed to be preprocessed before being used. All of the following steps were carried using ArcGIS Pro 2.8.0. First, the two image products (pre and post) were used to produce pansharpened color images (using the RGB and panchromatic images). Then the shapefiles containing the image polygons were overlaid on top of the images. A subset of the provided polygons was selected for memory constraints. These shapefiles contain the polygon geometries as well

[2]https://www.maxar.com/

[3]https://openmaplebanon.org/beirut-recovery-map

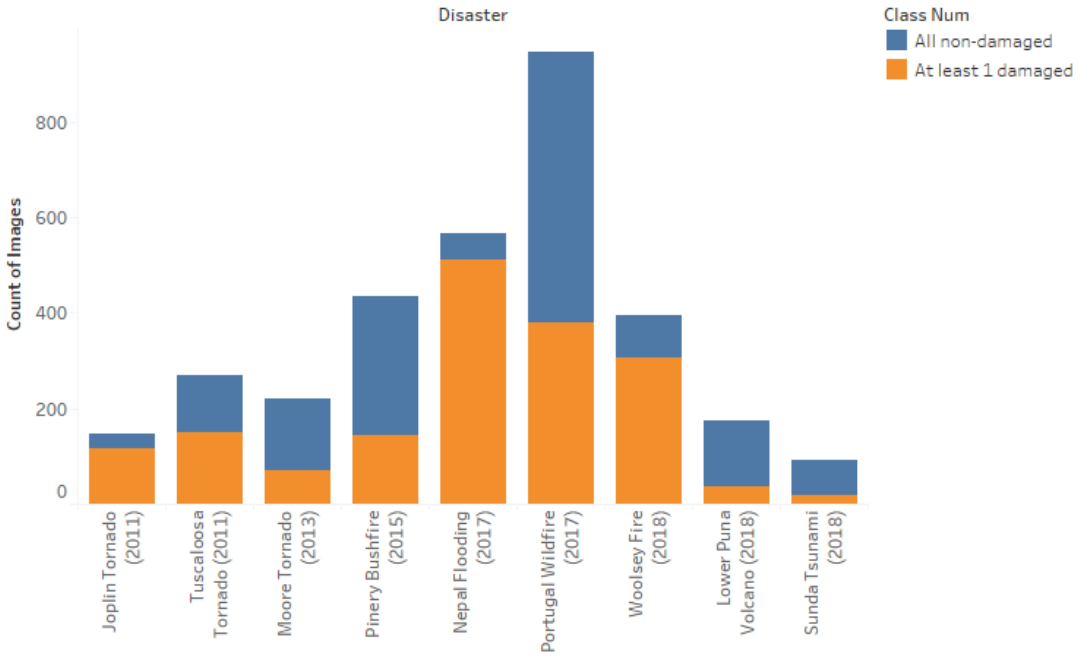

Figure 4.3: xBD Tier 3 chips with at least one damaged building.

as some additional features including the damage class which was converted to integer encoding. A view of this area as well as the class distribution of the included polygons are shown in Figures 4.4 and 4.5 respectively.

The first and problem with the data is that the building polygons are not aligned with the buildings in the actual images because the polygons were created[4] on a different image. Different satellite images are taken at different angles which produces this alignment error. Therefore, each one of the two images was georeferenced to the polygons. First, the entire image was moved in order to align with as many polygons as possible. However, it is not possible to achieve a satisfactory general alignment using this method since the misalignment is not uniform across the entire map. The next step was to manually add ground control points (GCP) linking each building in the image to its corresponding polygon. Once all the GCPs were added, a third order transformation was used to fit the image into the GCPs thus reducing the misalignment error. However, this does not achieve a perfect fit. An example of a corrected polygon misalignment is shown in Figure 4.6.

The second problem was that the polygons were drawn to the shape of the building rooftops. This assumes that the satellite image was taken at nadir and therefore only the rooftops of buildings would be visible and this is often not the case. For most buildings this would not cause an issue. However for tall

[4]`https://www.beiruturbanlab.com/en/Details/666/municipal-beirut-basemap-available-for-download`

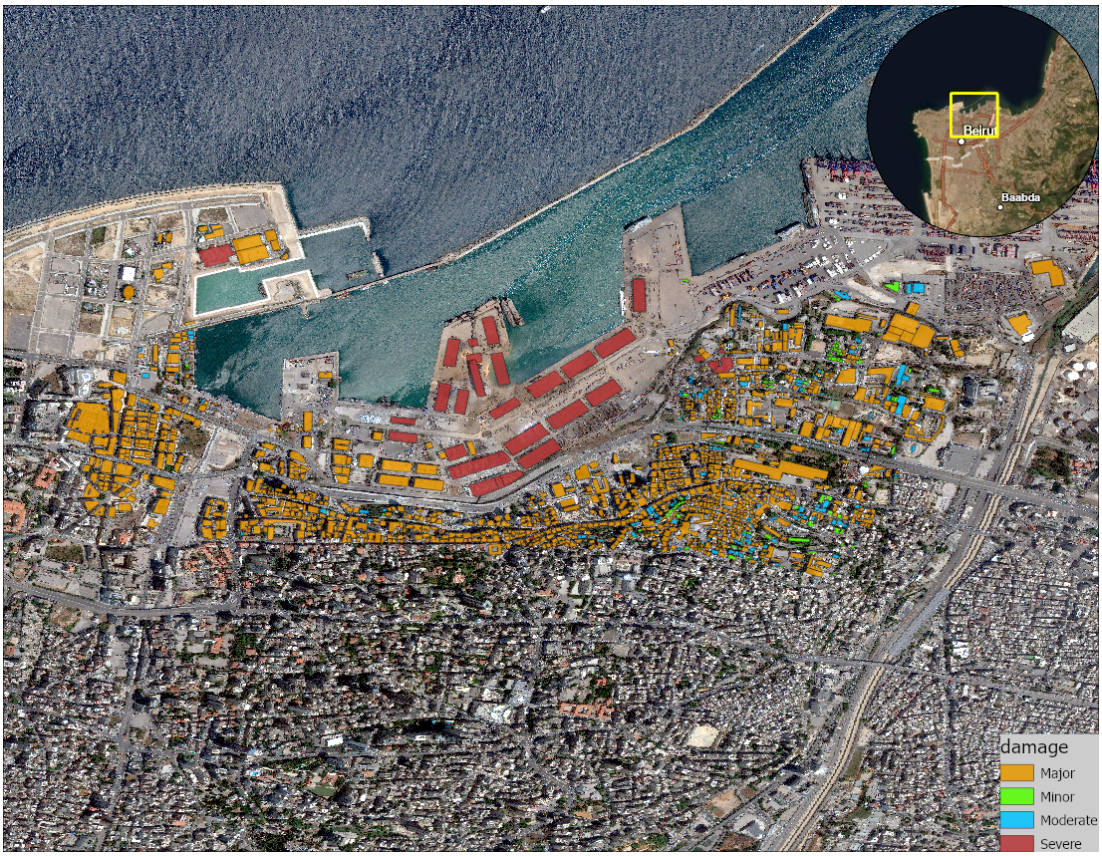

Figure 4.4: Overview of the selected area.

buildings, this would have resulted in the cropped image showing only a small portion of the visible part of the building. A similar difficulty was also reported by Zheng et al. [45]. An example of this scenario is shown in Figure 4.7.

To mitigate this, we create a rectangular envelope around each polygon buffered by five meters in each direction. This also helps with the georeferencing error. A similar procedure was proposed by Miura et al. [34]. An example of a buffered mask is shown in Figure 4.8.

The centroid latitude and longitude for each of the buffered masks was calculated for building the graph. Finally, the image chips defined by the buffered masks were extracted.

The meta-features dataset contains multiple building features collected from multiple sources. To join the explosion damage database with this database, we first convert this dataset's coordinate reference system (CRS) from Stereographic Levant to World Geodetic System (WGS) 84 using a geodetic translation with (-186.99, -20.42, 272.936) XYZ values. Due to the absence of a common sample identification key between the two datasets, we perform a spatial join operation. As feature preprocessing, we first drop the features related to electricity consump-

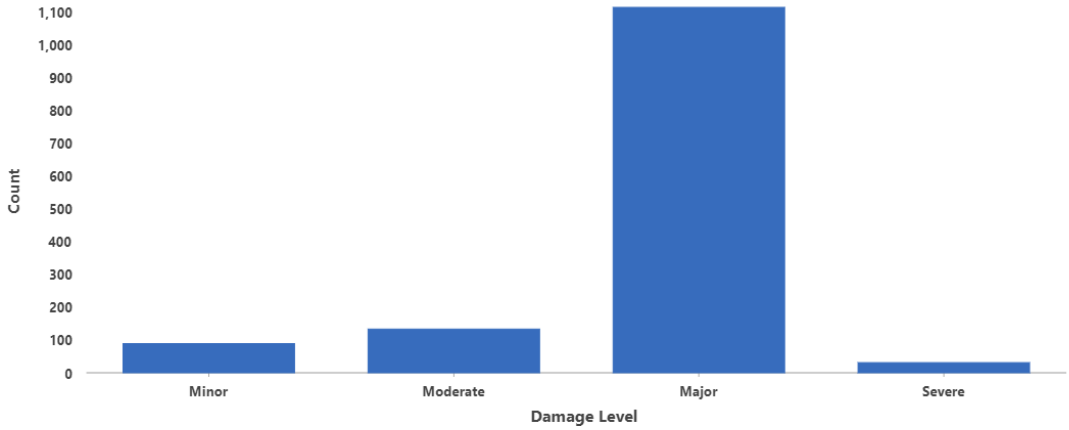

Figure 4.5: Class distribution of the selected samples.

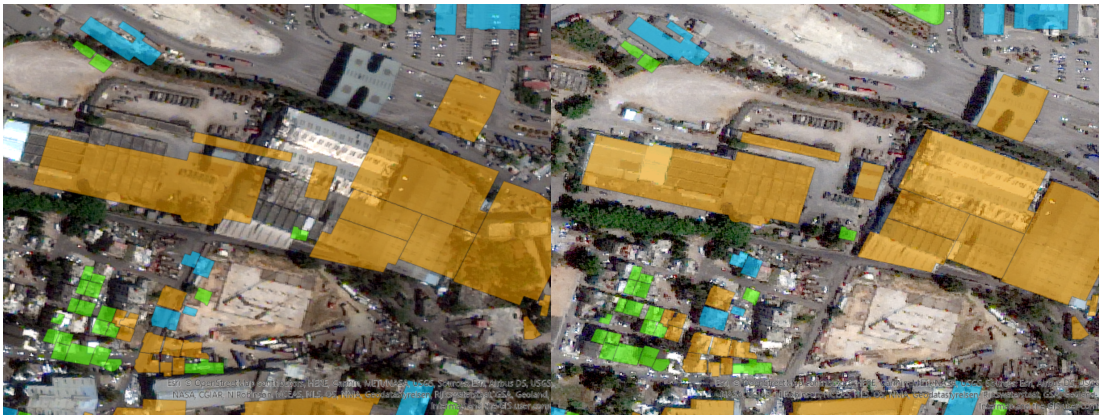

Figure 4.6: Before (left) and after (right) georeferencing.

tion as we deem them irrelevant. We retain features related to the number of apartments, area, perimeter, number of floors, digital surface model (DSM) value, building height, construction year, construction era, heritage status and building function. Then we consolidate the different columns containing the same feature coming from different sources into single columns to minimize missing values. However, the data still had a lot of missing values. Categorical features such as building function are converted to one-hot encoding format. Figure 4.9 shows the distributions of the Heritage and Building Function attributes. For the case of the Function attribute, the value "Other" is equivalent to a missing value. Finally, all columns were normalized to a range between zero and one with zero representing missing values. Table 4.1 shows some statistics regarding the meta-features including the number of missing values (nulls) for each attribute.

Figure 4.7: A polygon covering only part of the building.

Table 4.1: Summary Statistics of the Beirut Meta-features.

	Number of apartments	Mean DSM	Mean building height	Area	Perimeter	Era	Built year	Floors
Count	535	1023	1023	1257	1257	711	789	870
Number of nulls	831	343	343	109	109	655	577	496
Mean	6.72	41.24	18.47	568.03	86.95	2.74	1954.41	4.17
Std	6.95	18.39	10.88	1484.34	76.44	0.97	36.93	2.63
Min	1	3.78	1.66	17.58	17.81	1	1219	1
Max	60	113.70	95.65	16574.86	1062.72	5	2021	26

4.2 Model Evaluation

4.2.1 Performance Metrics

For model evaluation, we refer to SOTA literature to decide which metrics to use. For the transductive experiments, we compute the accuracy (Equation 4.1), precision (Equation 4.2), recall (Equation 4.3) and specificity (Equation 4.4) as was done by Saha et al. [39] along with the F1 score (Equation 4.5). For the inductive case, the accuracy and ROC (receiver operating characteristic) AUC (area under the curve) score are used as was done by Xu et al. [42]. The ROC curve is a plot of the true positive rate (recall) with respect to the false positive rate ($1 - specificity$). This curve represents the ability of a model to distinguish between positive and negative samples. The area under this curve (AUC) is used as a summary metric of this curve. The closer the AUC is to 1 the more

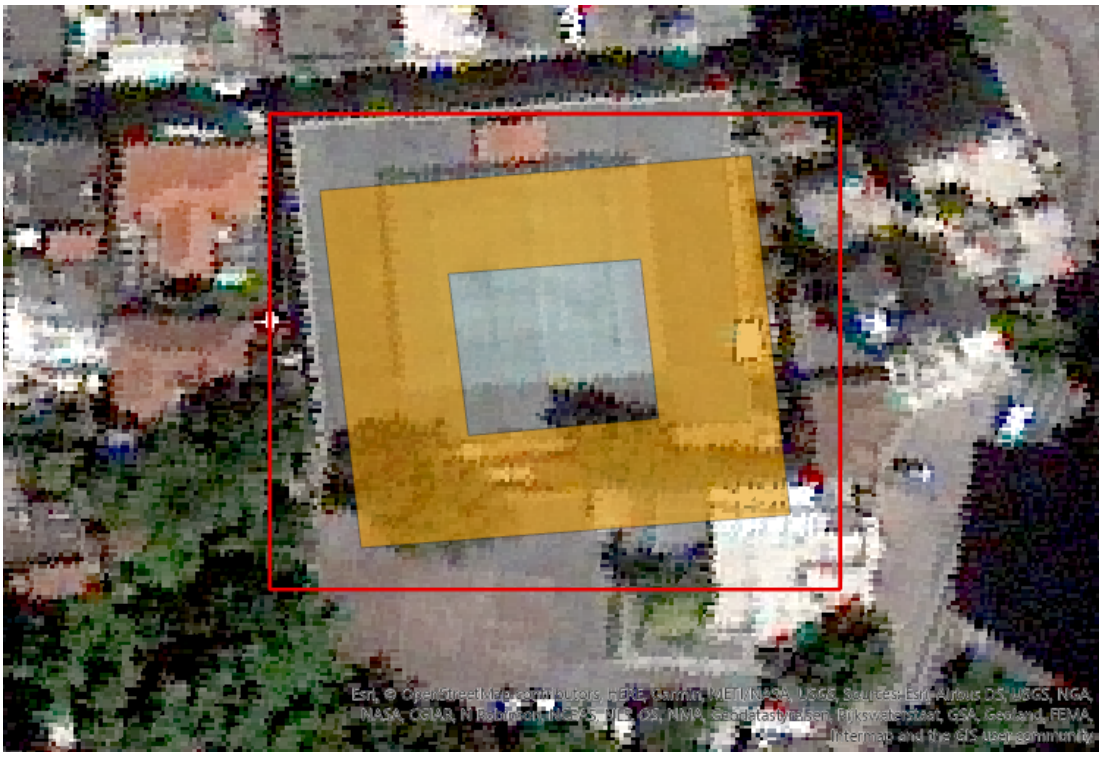

Figure 4.8: Buffered mask (red) around the original polygon (orange).

capable the model is of correctly distinguishing between the classes. We also supplement it with the F1 score which is, along with The AUC score, sensitive to class imbalance which is prevalent in this dataset. The aforementioned metrics are shown in the Equations below where:

- TP (true positive): number of positive samples predicted as positive.
- TN (true negative): number of negative samples predicted as negative.
- FP (false positive): number of negative samples predicted as positive.
- FN (false negative): number of positive samples predicted as negative.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$recall = \frac{TP}{TP + FN} \tag{4.3}$$

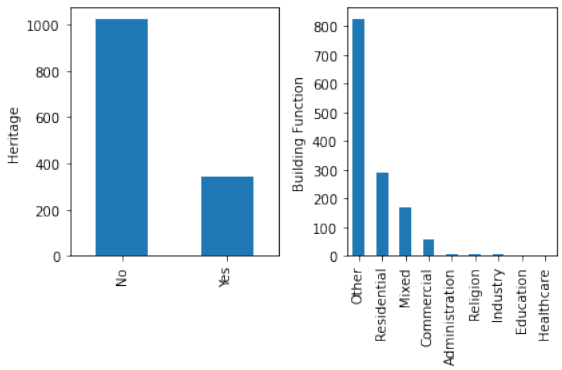

Figure 4.9: Counts of Beirut heritage buildings and Beirut buildings by function.

$$specificity = \frac{TN}{TN + FP} \tag{4.4}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{4.5}$$

4.2.2 Model Benchmarking

Since none of the works in the literature report results on the same data sampling scheme followed here, we rely on training benchmark models and comparing their performance to our method.

For evaluating the semi-supervised GCN, we train and report the performance of the Semi-supervised Multiresolution Autoencoder by Ienco and Pensa [4]. This model was also used for benchmarking in the recent work by Saha et al. [39]. The key concept behind this autoencoder is that it solves a classification problem in addition to the traditional reconstruction problem. The architecture (Figure 4.10) consists of a traditional encoder-decoder with a bottleneck layer in the middle which is also connected to a prediction layer. The encoder-decoder is trained to reproduce the input at the decoder output by optimizing the mean squared error reconstruction loss calculated on the entire data. This is important for learning robust data embeddings at the bottleneck layer which are subsequently used for prediction. The classification categorical crossentropy loss is calculated only on a small number of training samples. This tunes the bottleneck embeddings to be useful for classification and not just for reconstruction. This method also adopts an ensemble strategy where multiple autoencoders are trained with a randomly sampled number of neurons for the bottleneck and intermediate layers. The output of the model is the concatenation of the bottleneck embeddings of all autoencoders. The samples are then partitioned into different groups by applying k-means clustering on the embeddings.

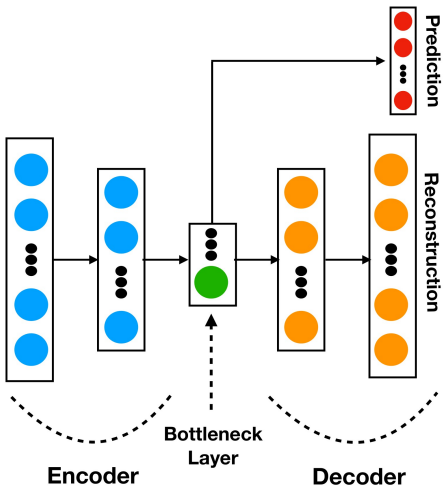

Figure 4.10: Semi-supervised autoencoder architecture (Image credit: [4])

To calculate the desired metrics on the output of the autoencoder, we needed to establish which cluster corresponds to which class. This was done based on the majority class per cluster. Each cluster is considered to represent the class of the majority of the samples that were placed in it. In practice, this was done by building the confusion matrix which would be initially disordered and then rearranging either the rows or the columns to maximize the sum of the diagonal [101]. The right permutation is found using the Hungarian algorithm [102].

For evaluating the supervised Graph SAGE, we adopt our own benchmark model, a Siamese CNN, which is nearly identical to the Graph SAGE architecture except that it does not perform graph convolutions. This was done by simply replacing the SAGE layers with fully connected layers. The reason behind adopting such a benchmark is that we want the models to be relatively on the same complexity level with only the SAGE graph convolution being the difference. This allows us to clearly highlight the contribution of the graph convolutions when comparing the two models.

4.2.3 Null Hypothesis Testing

In the previous paragraph, we have described how we compare our methodology to other benchmark models in order to assess its merit. Such a claim can be made more certain by eliminating the possibility that the difference in performance is situational and proving that it is statistically significant. This is done by performing a hypothesis test in order to reject the null hypothesis which assumes the difference is due to chance.

For this purpose, we adopt the paired Student's t-test [103] and the Wilcoxon signed-rank test [104]. The latter is similar to the t-test however it is not parametric, meaning it does not assume any distribution of the data (the t-test assumes

a t distribution). Each of the two models being compared is trained thirty times on a differently sampled training set and its performance is reported on a hold set. To ensure that the independence condition is not violated and that there is no overlap between the re-sampled training sets and the hold set, the latter is separately partitioned and the remainder of the data is randomly sampled into a different training set every run. The variable considered for the test is the difference between the metrics of the two models. A separate test is carried for each of the calculated metrics with a significance level of 5%.

4.3 Experimental Setup

All of the below experiments and procedures were implemented using Python 3.8. GCN Models were built and trained using Pytorch 1.7.1 and Pytorch Geometric 1.7.0 [105]. The siamese CNN was built using Pytorch and the multiresolution semi-supervised autoencoder was built using Tensorflow 2.3. Additional computational libraries used include Numpy, Scikit-learn, PIL (Python Image Library), Scipy and Pandas. Code was executed on a virtual machine running on 8 cores of an AMD EPYC 7551 32-Core Processor with an Nvidia V100 32GB GPU. The code base that was used to perform these experiments is available[5][6]. The Adam optimizer [106] was used with the categorical crossentropy loss function for all models (except for the autoencoder which was trained as specified by the authors). The loss function included class weights in order to mitigate the impact of class imbalance. The class weights were calculated based on Equation 4.6 which was inspired by King and Zen [107]. Additionally, the "major damage" and "destroyed" classes in the xBD dataset were merged into a single class since they are very nuanced and are hard to distinguish which would degrade the performance of our models. This was also done by Xu et al. [42] on their dataset. For similar reasons, we merge the "minor" and "moderate" classes in the Beirut dataset.

$$class\_weight = \frac{n\_samples}{n\_classes \times class\_occurrence} \tag{4.6}$$

For all models (except for the autoencoder), the number of neurons, number of layers, dropout rate, learning rate and batch size were tuned in a way to reduce overfitting and improve performance on the test set. The final adopted configuration is two graph convolutional (fully connected in the case of the Siamese CNN) layers with 32 neurons and a dropout rate of 0.5. The batch size was set to 256 (only for the supervised models) and the learning rate to 0.0003.

[5] `https://gitlab.com/awadailab/gcn-remote-sensing`
[6] `https://gitlab.com/awadailab/sage-project`

4.4 Semi-supervised GCN with the xBD dataset

The first set of experiments aims to evaluate the merit of our semi-supervised GCN by training and testing it on data from xBD. We use the Pinery Bushfire, Joplin Tornado and Nepal Flooding events from the Tier 3 set. One important aspect of this method is that it needs to observe all of the training data during training because it is semi-supervised. Moreover, in the case of GCN, all of the data needs to be loaded at once since it is necessary to have all the samples to build the full graph.

In order to accommodate this requirement with the memory resources available, we devised a strategy to prune the data. First of all, all unlabeled samples were discarded. As previously mentioned, each disaster event is divided into equal image chips. Figure 4.11 shows the image chips for Joplin Tornado. We have also shown (Figure 4.3) that many chips do not have any damaged samples at all. This is because the damage is more concentrated around the location of the damaging force. Figure 4.12 shows how the damage is mostly concentrated along the trajectory of the tornado.

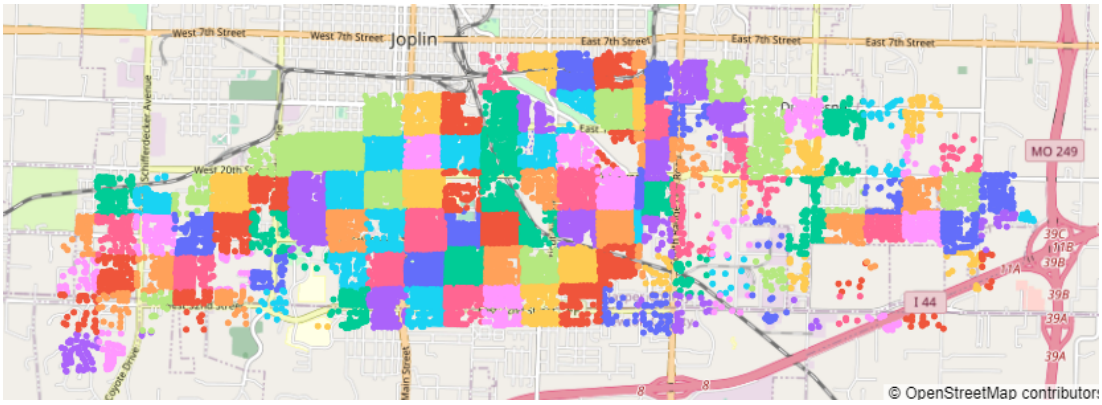

Figure 4.11: Image chips of Joplin Tornado as squares of different color.

Therefore, We discard image chips that do not have any damaged samples. Finally, if the data still needed to be reduced, a random subset is sampled while preserving the class distribution of the original data.

The obtained samples are used to build the graph. Afterwards, each node in the graph is marked as being either training, testing and hold. This assignment is done randomly while maintaining the same class distribution. The training nodes are a small number of nodes for which the loss function will be computed during training and used to optimize the model. Testing samples will be used to compute performance metrics after every epoch for model selection. After the training is done the metrics are computed on the hold set as well as the entire graph. Metrics are also computed on the training set after each epoch.

We experimented with different sizes for the training set and found that us-

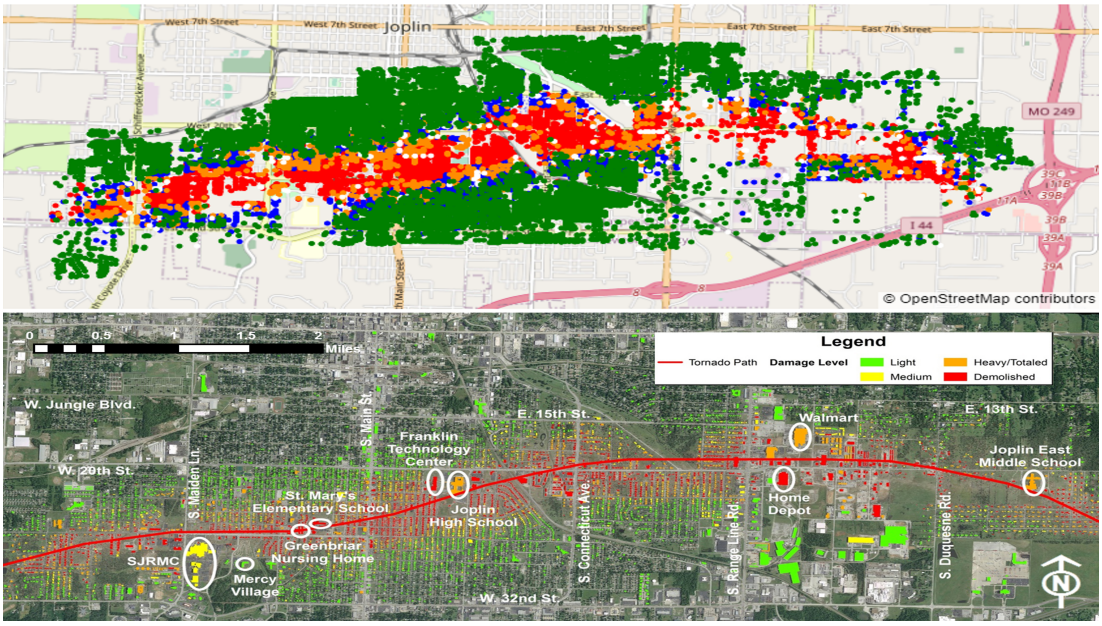

Figure 4.12: Above: Joplin Tornado damage as provided in the xBD dataset. Below: Joplin Tornado trajectory (Image credit: `https://www.nist.gov/image/aerialimageofjoplinmissouritornadodamagejpg`).

ing 20% of the data to be the best compromise in terms of performance and maintaining a relatively small number of training samples (Figure 4.13).

4.4.1 Benchmarking with Multiresolution Autoencoder

The multiresolution autoencoder was trained on the same data that was used to train the GCN with exactly the same train, test and hold splits. The hyperparameters such as the number of layers, batch size and learning rate were used as proposed by the authors [4]. However, the number of intermediate neurons was specified as being proportional to the number of features. We had to scale that down while maintaining the proportional relationship because due to the large number of features we have (pixels) with the original scaling that number became too large and the memory footprint of the model exploded. Table 4.2 show the results for both the GCN and the multiresolution autoencoder.

In general, we find that the GCN systematically outperforms the autoencoder. Fig. 4.14 shows the difference between the BLDNet and autoencoder hold scores for all three disasters. We achieved an average increase of 16.3%, 14.05%, 16.51%, 8.62% and 15.98% for accuracy, precision, recall, specificity and F1 score respectively on the hold set across disasters. Recall saw the highest average improvement which signifies a notable amelioration in the ability to detect the positive class (damage). We also see a trend where specificity tends to

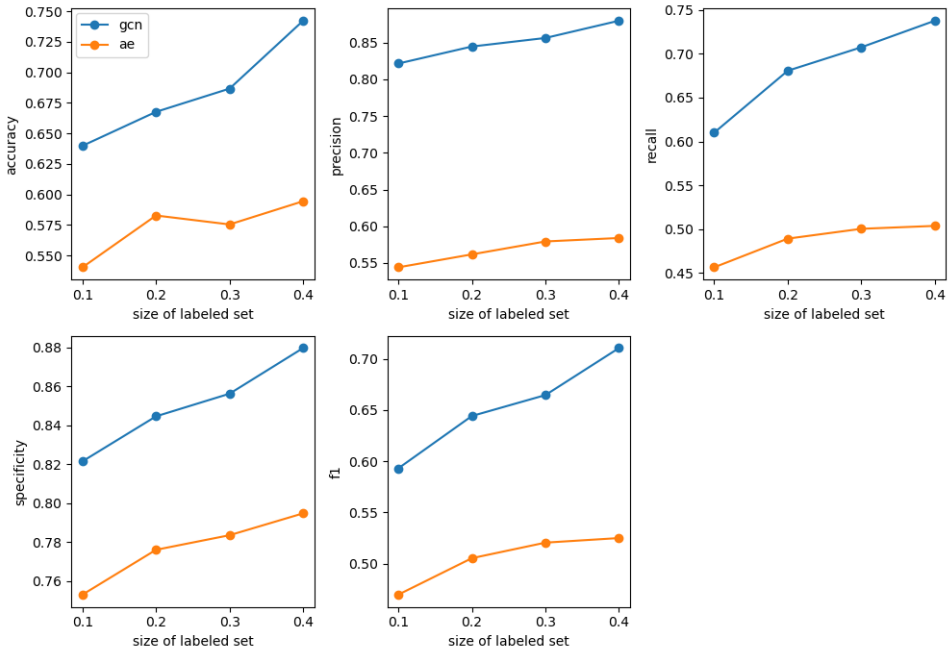

Figure 4.13: Model performance with respect to the training data size.

be higher than other metrics. This measure represents the model performance on the negative class as opposed to recall. This difference is more pronounced the more unbalanced the dataset is. To show this, we compute the Shannon equitability index (range 0-1 with 1 being balanced and 0 unbalanced) [108] for each of the three sets we used in both their original and pruned versions. We also calculate the difference between the specificity and recall scores on the full set for each disaster (Table 4.3). First of all, we find that all pruned sets have a higher index than their original counterparts which means that our data pruning had a positive impact on the unbalanced state of the dataset. This is due to our prioritization of non-damaged samples for removal. Also, the difference between specificity and recall is consistently reduced with lower class imbalance (higher Shannon index). We therefore conclude that decreasing class imbalance and the prevalence of non-damaged samples increased the model's ability to accurately detect damage. The disasters in Table 4.3 are sorted to reflect this pattern.

Additionally, we investigate the class separation ability of this model by producing a TSNE (t-distributed stochastic neighbor embedding) [109] visualization of the node embeddings produced by the GCN (Figure 4.15). We notice that each class forms a cluster. However, we also note that the "minor damage" class lies between the two other classes and that it does not form a cluster as separated and condensed as the other two since we see points belonging to the "minor damage" class within the other clusters. This gives credit to our and Xu et al.'s [42] assumption that intermediate damage classes can be ambiguous and harder

Table 4.2: Comparison between our Semi-supervised GCN and the Multiresolution Autoencoder (AE) with Bold Italics Indicating Better Performance.

Set		AE						GCN				
		Acc	**Precision**	**Recall**	**Specificity**	**F1**		**Acc**	**Precision**	**Recall**	**Specificity**	**F1**
Pinery Bushfire	**Train**	0.6044	0.4935	0.4811	0.7386	0.4495	**Train**	***0.9524***	***0.8444***	***0.9813***	***0.9822***	***0.9023***
	Test	0.5000	0.3338	0.3288	0.6357	0.3013	**Test**	***0.7146***	***0.4515***	***0.5057***	***0.7791***	***0.4660***
	Hold	0.4773	0.3439	0.3090	0.6651	0.3044	**Hold**	***0.7145***	***0.4517***	***0.4941***	***0.7743***	***0.4646***
	Full	0.3909	0.2993	0.3226	0.6028	0.2588	**Full**	***0.7247***	***0.4671***	***0.5182***	***0.7918***	***0.4826***
		Acc	**Precision**	**Recall**	**Specificity**	**F1**		**Acc**	**Precision**	**Recall**	**Specificity**	**F1**
Joplin Tornado	**Train**	0.6337	0.5450	0.4925	0.8033	0.5067	**Train**	***0.9634***	***0.9462***	***0.9696***	***0.9839***	***0.6562***
	Test	0.5171	0.5188	0.4274	0.7484	0.4457	**Test**	***0.7610***	***0.6968***	***0.6968***	***0.8802***	***0.6900***
	Hold	0.6179	0.5382	0.4967	0.7837	0.4967	**Hold**	***0.7540***	***0.6987***	***0.6992***	***0.8778***	***0.6914***
	Full	0.5988	0.5400	0.4893	0.7809	0.4961	**Full**	***0.7606***	***0.7049***	***0.7027***	***0.8810***	***0.6954***
		Acc	**Precision**	**Recall**	**Specificity**	**F1**		**Acc**	**Precision**	**Recall**	**Specificity**	**F1**
Nepal Flooding	**Train**	0.6484	0.3976	0.6864	0.6836	0.3905	**Train**	***0.9707***	***0.9310***	***0.9871***	***0.9889***	***0.9566***
	Test	0.5463	0.3664	0.3701	0.6958	0.3643	**Test**	***0.6780***	***0.4907***	***0.4535***	***0.7344***	***0.4623***
	Hold	0.5930	0.4056	0.4121	0.7059	0.4082	**Hold**	***0.7086***	***0.5589***	***0.5199***	***0.7613***	***0.5328***
	Full	0.5754	0.3712	0.3733	0.6907	0.3720	**Full**	***0.7152***	***0.5585***	***0.5244***	***0.7710***	***0.5363***

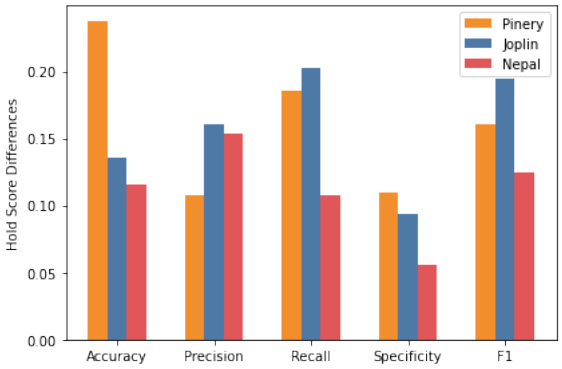

Figure 4.14: Performance difference with respect to the autoencoder for all three disasters.

to distinguish from other classes. These results are achieved without using an ordinal crossentropy loss function and therefore the model had no incentive to treat the classes as being ordered.

To further ensure the validity of these results, we ran the Student t-test and Wilcoxon signed rank test as previously described. The null hypothesis was rejected for every metric meaning that the difference in performance is not due to chance. Looking at the distributions of the metric population (Figure 4.16), it can be argued that the difference is clearly significant without the need to run a statistical test.

4.5 Semi-supervised GCN with the Beirut Explosion dataset

The second set of experiments demonstrates the effectiveness of our method on a different scenario. The Beirut Port explosion is a man-made disaster resulting from artificial explosives as opposed to the natural disasters we have dealt with

Table 4.3: Shannon Equitability Index for the Chosen xBD Sets and the Difference between Specificity and Recall.

	Joplin Tornado	Nepal Flooding	Pinery Bushfire
Shannon equitability original	0.813	0.558	0.247
Shannon equitability pruned	0.878	0.597	0.495
Specificity - Recall	0.1783	0.2466	0.2736

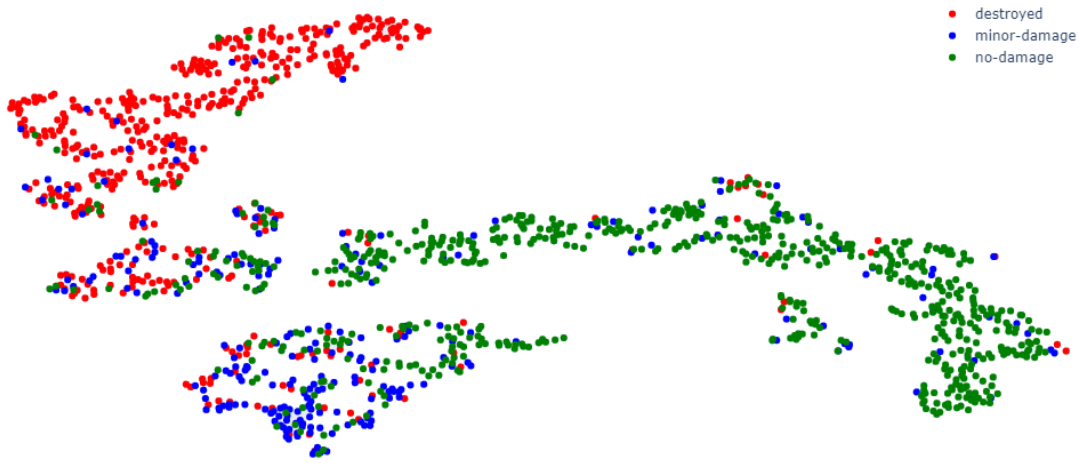

Figure 4.15: GCN node embeddings for Joplin tornado.

previously. Also, Beirut presents the case of a city with largely irregular and non-homogeneous urban and architectural layouts which entails that buildings within close proximity of each other would interact differently with the damaging force.

We run these experiments with the same model and training configurations that were adopted for the xBD dataset. No data pruning strategy was needed here since the area of interest was selected during preprocessing (Figure 4.4). The first experiment was performed without any meta-features and the second experiment was performed with meta-feature injection. Table 4.4 shows the results of both experiments. Bold and italic numbers indicate superior numbers.

In most aspects, we find that the experiment with the meta features yielded better results. On the hold set, we achieve an average increment of 5.65% over the five metrics. The least improvement is 3.09% for specificity and the highest is 8.79% for accuracy. This means that augmenting the images with these additional contextual features helps the model better estimate how each building was impacted by the disaster. This improvement was achieved despite 32.18% of the data being missing on average across the meta-features.

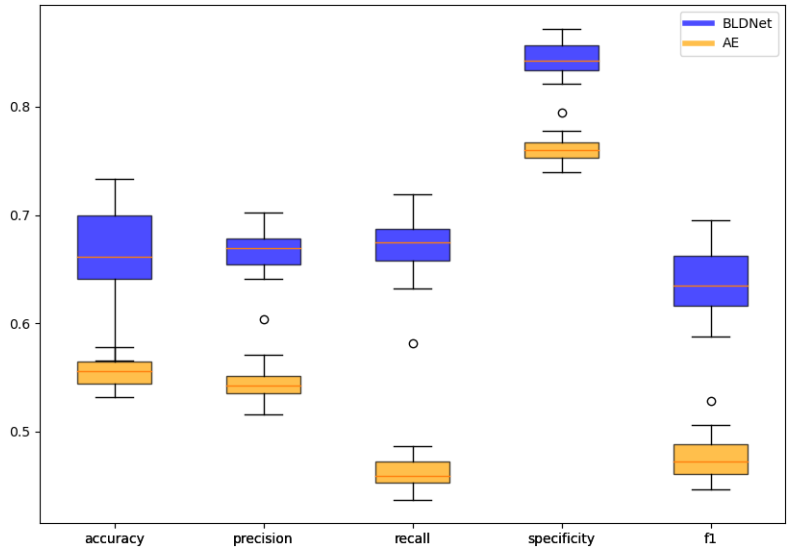

Figure 4.16: Distribution of the performance metrics obtained after 30 runs with different data samples.

Table 4.4: Comparison between the GCN with or without Meta-features on the Beirut Data with Bold Italics for Best Metrics.

		Accuracy	Precision	Recall	Specificity	F1
Beirut	**Train**	0.9534	0.8474	0.9804	0.9817	0.9033
	Test	0.7415	***0.5930***	***0.6744***	0.7810	***0.6189***
	Hold	0.7291	0.5993	0.6826	0.7815	0.6274
	Full	0.7460	0.6241	0.7276	0.8031	0.6595
		Accuracy	**Precision**	**Recall**	**Specificity**	**F1**
Beirut Meta-features	**Train**	***0.9963***	***0.9926***	***0.9985***	***0.9985***	***0.9955***
	Test	***0.7878***	0.5779	0.6571	***0.7827***	0.6100
	Hold	***0.8170***	***0.6533***	***0.7329***	***0.8124***	***0.6866***
	Full	***0.8133***	***0.6409***	***0.7407***	***0.8178***	***0.6808***

4.6 Supervised change detection and domain shift with Graph SAGE

We recall that in this work, a desired feature of our solution is ability to obtain predicitions in a quick manner because this is necessary for planning emergency disaster response for rescue and relief. The previous experiments achieve this by relying only on a small number of annotated samples. This experiment aims at addressing this aspect using a different approach: having a pre-trained model that can used on new disasters without training.

For this purpose, we adopt the SAGE graph operator since it can be trained on mini batches in a supervised manner and used to infer on unseen data. The experiment designed to test generalization was inspired by Xu et al. [42]. They

started by training and testing on the same disaster and then experimented with training on a certain disaster set and testing on a different disaster. We follow a similar strategy using disasters from the xBD dataset. Additionally, our experiments not only test training and testing sets belonging to different regions, but we chose different types of disasters. The last experiment aims to indicate how much leaking some of the test data into the training data improves performance. This simulates the case where a small number of labeled samples for the new disaster can be obtained (similarly to the semi-supervised case). Image chips with only one building and image chips with only one annotated building or with no labeled buildings were discarded since these would not be suitable for creating a graph. As a benchmark, the Siamese CNN model described previously was also trained on the same data. The data used for these experiments as well as the results for both models are shown in Table 4.5.

Table 4.5: Comparison between the Graph SAGE and the Siamese CNN.

Train	Test/Hold		Siamese CNN					Graph SAGE			
			Acc	Macro F1	Weighted F1	AUC		Acc	Macro F1	Weighted F1	AUC
Socal + Portugal fire	Socal	Train	0.8746	***0.5994***	0.9036	***0.9648***	Train	***0.9110***	0.5642	***0.9205***	0.9347
		Test	0.8439	***0.5267***	0.8696	***0.8606***	Test	***0.8672***	0.5042	***0.8768***	0.8567
		Hold	0.8581	***0.5449***	0.8787	***0.8609***	Hold	***0.8870***	0.5315	***0.8902***	0.8374
			Acc	**Macro F1**	**Weighted F1**	**AUC**		**Acc**	**Macro F1**	**Weighted F1**	**AUC**
Nepal flooding	Socal	Train	***0.6921***	***0.6342***	***0.7275***	***0.8764***	Train	0.5092	0.3421	0.5487	0.5314
		Test	0.4609	0.3223	0.5911	***0.6229***	Test	***0.5697***	***0.3245***	***0.6525***	0.6121
		Hold	0.4356	0.315	0.5708	***0.6601***	Hold	***0.6075***	***0.3275***	***0.6727***	0.4999
			Acc	**Macro F1**	**Weighted F1**	**AUC**		**Acc**	**Macro F1**	**Weighted F1**	**AUC**
Nepal flooding + Joplin tornado + Puna volcano	Socal	Train	***0.9283***	***0.9008***	***0.9306***	***0.9877***	Train	0.7042	0.2824	0.5870	0.5199
		Test	0.599	***0.3832***	0.6928	***0.7265***	Test	***0.8744***	0.3110	***0.8157***	0.6319
		Hold	0.6082	***0.3913***	0.6966	***0.6597***	Hold	***0.8542***	0.3072	***0.7873***	0.6445
			Acc	**Macro F1**	**Weighted F1**	**AUC**		**Acc**	**Macro F1**	**Weighted F1**	**AUC**
Nepal flooding + Joplin tornado + Puna volcano + 10% Socal Fire	Socal	Train	***0.9666***	***0.9522***	***0.9671***	***0.9963***	Train	0.7094	0.2873	0.5951	0.5394
		Test	0.6659	***0.4205***	0.7418	***0.7173***	Test	***0.8591***	0.3172	***0.7959***	0.6765
		Hold	0.6668	***0.4308***	0.7405	***0.6853***	Hold	***0.8552***	0.3131	***0.7899***	0.6665

The results shown in Table 4.5 do not demonstrate a clear advantage for the Graph SAGE which outperforms the Siamese CNN on 20 out of 48 total occasions in terms of raw performance. By looking at the results we can deduce that the likely reason is that the Graph SAGE is being more severely degraded by the class imbalance than the Siamese CNN. The Graph SAGE seems to outperform the Siamese CNN in terms of accuracy and weighted F1 score with an average increase of 14.74% and 6.34% on the hold set respectively. These two metrics ignore the impact of class imbalance. On the other hand, it falls behind most of the time in terms of macro averaged F1 and underperforms systematically in terms of AUC with an average difference of 5.07% and 5.44% on the hold set respectively. These last two metrics are both sensitive to class imbalance.

In terms of domain shift generalization, the Graph SAGE presents a more satisfactory behavior. We demonstrate this by computing the difference between the performance on the training set and the hold set for every experiment (Figure 4.17). For the first experiment, the Siamese CNN presents a lesser drop in performance compared to Graph SAGE. However, for subsequent experiments which are cross-domain, the difference for Graph SAGE is negative in most cases which means that performance on the hold set was higher than performance on the training set. Furthermore, the negative difference increases in magnitude as

we move from left to right. Meaning that generalization improved when augmenting the training data with more diverse disasters. This implies that having larger training data improves generalization to other domains even if none of the subsets in the training data are related to the test data. When the training data is supplemented with a small number of test samples, we achieve the best generalization. On the other hand, the Siamese CNN shows a consistent positive generalization gap which is likely due to overfitting the training data. We note that it is not suitable to compare the test set results between the last experiment and the others since for the last experiment, the leak data was taken from the test set.

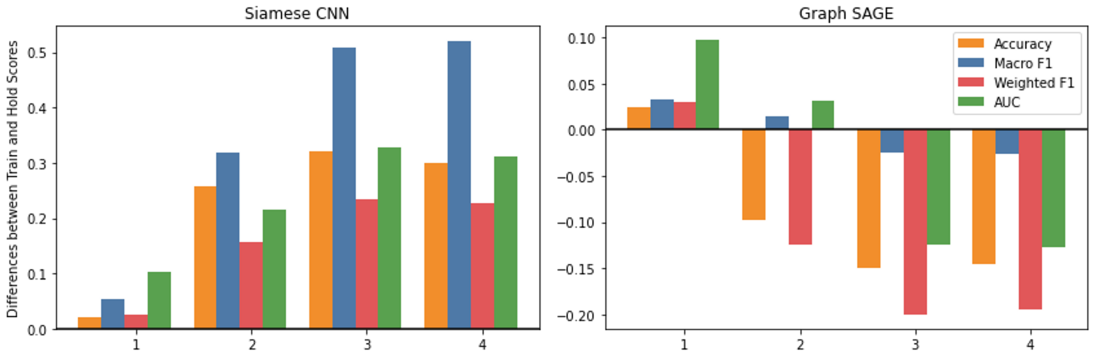

Figure 4.17: Difference between training and hold scores for the Siamese CNN and Graph SAGE. The x-axis indexes indicate the different experiments with different train - test configurations: 1) Fire - Fire, 2) Flooding - Fire, 3) Flooding + Tornado + Volcano - Fire and 4) Flooding + Tornado + Volcano + 10% Fire - Fire.

4.7 Computational and performance considerations for best practices

In this work we have expressed our aim to provide a solution that allows predictions for new disasters to be obtained in relatively fast manner. Here we compare the GCN and Graph SAGE approaches with respect to this requirement and we highlight the computational constraints and considerations we've had to address while working with these models.

For the GCN, we reiterate that it needs to observe all the data during training. This is typical for most if not all semi-supervised models. However, one additional constraint when it comes the GCN is that the entirety of the data is used to build the graph. This graph is loaded into working memory during training and inference which means that it would occupy a sizeable memory footprint. This was noted by Mou et al. [93] who also used a semi-supervised GCN with satellite

images. This is not the case for the multiresolution autoencoder for instance, since the entirety of the data is observed however it is observed in batches. Therefore, to be able to use this model to its best potential, a large GPU is needed that can hold all the data as well as the model itself. However, once these resources are provided, the GCN runs relatively fast. All our GCN experiments that included preprocessing the data, building the graph, training and producing predictions took around 10 to 15 minutes to finish on a virtual machine running on 8 cores of an AMD EPYC 7551 32-Core Processor with an Nvidia V100 32GB GPU.

As for the Graph SAGE, it addresses the aforementioned constraint by working with subgraphs which are effectively batches of the original graph. The major downside of this method is that it takes very long to train. Each of our experiments with the Graph SAGE took on average four days to finish. Also, building the subgraphs and storing them on disk (since they cannot all be built and stored in memory during runtime) takes a considerable amount of time (two to three days). These problems would be alleviated in case this long development and training procedure is done once and the model can be used for inference without training. However, despite a promising domain shift generalization ability, our results are not conclusive in terms of raw performance. Additionally, the best domain shift performance was achieved when a small number of labeled test samples was added to the training data. This effectively leads to a framework similar to the semi-supervised approach where few samples are annotated. Furthermore, even if domain shift performance was demonstrated for the data that we have, it would not guarantee that this would continue to be the case as new and different disasters occur across different regions. It might be that this pretrained model would have to continuously be refined in a lifelong learning manner which is still a time consuming process.

Chapter 5

Conclusion and Future Work

Every year, countless human lives are jeopardized by devastating events of various sources, magnitude and extent. In such cases, awareness of the location and level of destruction is essential for planning an effective response. By knowing which sites need to be prioritized, response teams can maximize the amount of lives that can be saved. More importantly, this information needs to be obtained as quickly as possible as delays can cause more lives to be lost due to injuries or entrapment. These maps are not only beneficial for rescue and relief efforts as they are also instrumental for planning ground survey missions. By knowing the location and the rough estimation of the damage severity, field inspectors can select where to concentrate their efforts to produce detailed reports about structural hazards and estimated repair costs in a relatively quick manner. Satellite imagery remains one of the most reliable data sources that can be used to gain such information. The main advantage of satellites is that they can cover very large areas with their images. With the availability of these images, artificial intelligence and machine learning have proven to be valuable tools for detecting disaster damage. Like most vision applications, convolutional neural network models are currently at the core of this field.

When it comes to disaster building damage, the majority of the destruction is usually clustered around the path or the epicenter of the disaster such as the epicenter of an earthquake or the trajectory of a tornado. Therefore, when it comes to assessing damage, it can be expected that neighboring buildings would have a similar damage signature. In many cases, the urban layout of the devastated city is non-uniform and therefore it is very common to observe neighboring buildings of different age, height and function. This adds an additional layer of interactions that makes these buildings sustain damage differently. All of these aspects are not addressed by traditional models since they do not take into account the similarities between neighboring buildings nor do they incorporate additional domain knowledge related to building properties.

In this work, we have presented a graph formulation that allows representing building damage data in a way where buildings are connected to their neigh-

bors and therefore allows exploiting these relationships for assessing damage. Additionally, we have shown how our formulation can augment satellite image data with additional building features. Our work also takes into account the necessity of providing damage mappings with the least time and effort possible. We proposed a novel architecture comprising a graph convolutional network with a Siamese convolutional backbone for extracting local features and aggregating them with neighboring features. We demonstrated the effectiveness of this method on the xBD dataset by comparing it to the multiresolution autoencoder benchmark. We also showcased how it can be effectively applied on the Beirut Port explosion data and obtained improved performance when augmenting the image data with building meta-features. Through semi-supervised learning, few images need to be labeled which enables our model to obtain predictions in an efficient manner and alleviates the need to label a large number of images. Another approach we explored is the possibility of having a pretrained model with good generalization ability that would be used to infer on new disasters with little to no training. For this we adopted a different graph aggregation operator (SAGE) based on sampling mini batches of graphs (subgraphs) and trained it in a supervised setting on different subsets of the xBD dataset to assess its cross domain generalization. Our results do not allow us to recommend this as much as we recommend the semi-supervised GCN especially given that it has very long training times and its performance was severely impaired by class imbalance. However, we have found that Graph SAGE can close the generalization gap and that increasing the number of disasters in the training data improves performance on the unseen disaster. This leads us to speculate that given large enough training data and addressing the class imbalance issue can lead to more favorable results from the SAGE model.

Based on the above and the insights we gathered throughout this work, we recommend the following as potential new directions for future research:

- In this work we have only considered RGB satellite images while other satellite bands are known to be good indicators for specific types of surfaces such as vegetation or built surfaces. The donated Beirut Port data included a multitude of multispectral products. However, the potential improvement from adding these bands was not explored since we needed to evaluate our novel approach on public data which is strictly RGB.

- In many cases when a disaster occurs the pre and post images are not from the same satellite. This data heterogeneity introduces additional noise that makes the task more challenging. In this work, we encountered some data heterogeneity with the Beirut data since the pre and post images were taken at different angles and under different illumination conditions. However, we have not tested the robustness of our method against data coming from two different satellites having different image resolutions. In many other works, heterogeneity was shown to degrade model performance [44].

- After the Beirut Port explosion, there have been several attempts at documenting the event at a street view level by collecting street view images of the damaged buildings[1]. Since these images are geotagged, our graph formulation would be suitable for using this data for assessing the damage from the street view images which would replace or be combined with the satellite imagery. Street view images are valuable since they offer a point of view that cannot be seen from above.

- In terms of meta-feature injection, we highlight that the flexibility of our graph formulation can be further exploited by injecting these meta-features at the edge level and not the node level only. In this work, we had intended to add the difference between the building heights as the edge weight. This is more suitable for capture the impact of varying building heights where a tall building may shield a short building next to it. However, due to the prevalence of missing values in our dataset, we decided that this is more likely to hurt performance.

[1] `https://openmaplebanon.org/beirut-recovery-map`

Appendix A

xBD Dataset Tier 1 Class Distribution

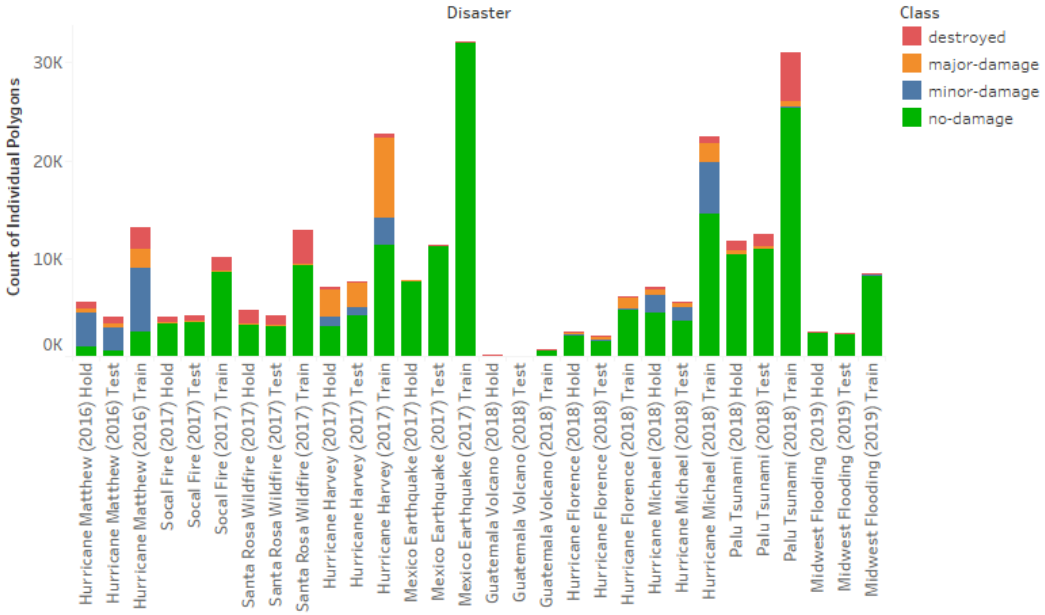

Figure A.1: xBD Tier 1 building per class.

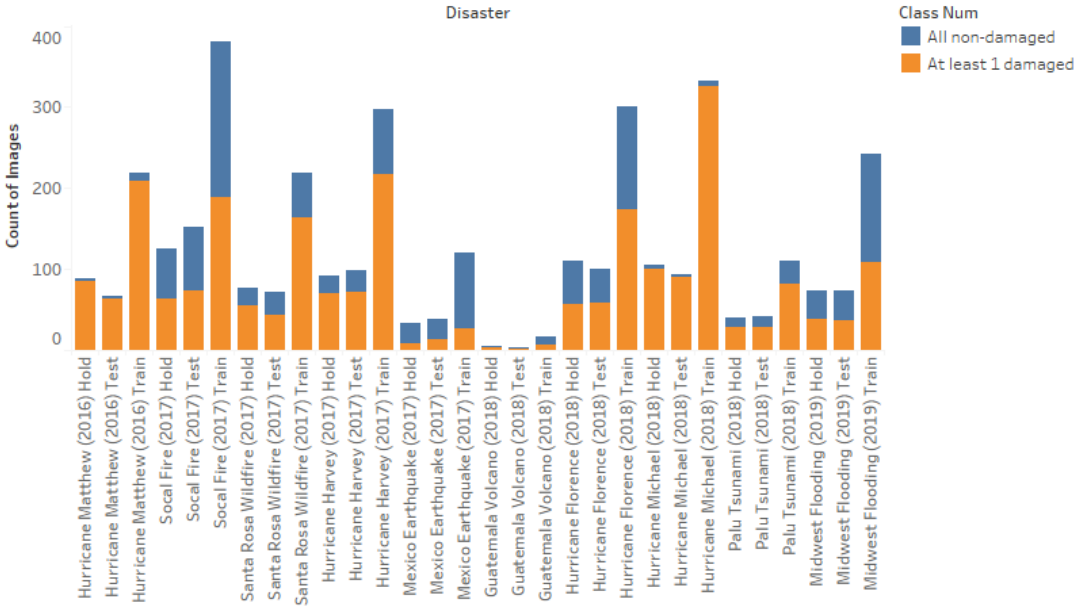

Figure A.2: xBD Tier 1 chips with at least one damaged building.

Appendix B

Abbreviations

AE	Autoencoder
AI	Artificial Intelligence
AUC	Area Under the Curve
CD	Change Detection
CM	Change Map
CNN	Convolutional Neural Network
CVA	Change Vector Analysis
DI	Difference Image
GCN	Graph Convolutional Network
HS	Hyperspectral
LULC	Land Use Land Cover
MS	Multispectral
ReLu	Rectified Linear Unit
RGB	Red Green Blue
RNN	Recurrent Neural Network
RS	Remote Sensing
SAGE	Sample and Aggregate
SAR	Synthetic-aperture Radar
SNR	Signal-to-noise Ratio
SOTA	State-of-the-art
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle

Bibliography

[1] W. W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.

[2] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020.

[3] R. Gupta, R. Hosfelt, S. Sajeev, N. Patel, B. Goodman, J. Doshi, E. Heim, H. Choset, and M. Gaston, "xBD: A Dataset for Assessing Building Damage from Satellite Imagery," *arXiv:1911.09296 [cs]*, Nov. 2019.

[4] D. Ienco and R. G. Pensa, "Semi-Supervised Clustering With Multiresolution Autoencoders," in *2018 International Joint Conference on Neural Networks (IJCNN)*, (Rio de Janeiro), pp. 1–8, IEEE, July 2018.

[5] A. Singh, "Review Article Digital change detection techniques using remotely-sensed data," *International Journal of Remote Sensing*, vol. 10, pp. 989–1003, June 1989.

[6] S. H. Khan, X. He, F. Porikli, and M. Bennamoun, "Forest Change Detection in Incomplete Satellite Images With Deep Neural Networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, pp. 5407–5423, Sept. 2017.

[7] B. Tan, J. G. Masek, R. Wolfe, F. Gao, C. Huang, E. F. Vermote, J. O. Sexton, and G. Ederer, "Improved forest change detection with terrain illumination corrected Landsat images," *Remote Sensing of Environment*, vol. 136, pp. 469–483, Sept. 2013.

[8] C. E. Woodcock, S. A. Macomber, M. Pax-Lenney, and W. B. Cohen, "Monitoring large areas for forest change using Landsat: Generalization across space, time and Landsat sensors," *Remote Sensing of Environment*, vol. 78, pp. 194–203, Oct. 2001.

[9] K. Rokni, A. Ahmad, K. Solaimani, and S. Hazini, "A new approach for surface water change detection: Integration of pixel level image fusion and image classification techniques," *International Journal of Applied Earth Observation and Geoinformation*, vol. 34, pp. 226–234, Feb. 2015.

[10] A. Song, Y. Kim, and Y. Kim, "Change Detection of Surface Water in Remote Sensing Images Based on Fully Convolutional Network," *Journal of Coastal Research*, vol. 91, p. 426, Aug. 2019.

[11] F. Gao, X. Wang, Y. Gao, J. Dong, and S. Wang, "Sea Ice Change Detection in SAR Images Based on Convolutional-Wavelet Neural Networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, pp. 1240–1244, Aug. 2019.

[12] Y. Gao, F. Gao, J. Dong, and S. Wang, "Transferred Deep Learning for Sea Ice Change Detection From Synthetic-Aperture Radar Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, pp. 1655–1659, Oct. 2019.

[13] A. Ding, Q. Zhang, X. Zhou, and B. Dai, "Automatic recognition of landslide based on CNN and texture change detection," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, (Wuhan, Hubei Province, China), pp. 444–448, IEEE, Nov. 2016.

[14] F. Bovolo and L. Bruzzone, "A Split-Based Approach to Unsupervised Change Detection in Large-Size Multitemporal Images: Application to Tsunami-Damage Assessment," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, pp. 1658–1670, June 2007.

[15] J. Sublime and E. Kalinicheva, "Automatic Post-Disaster Damage Mapping Using Deep-Learning Techniques for Change Detection: Case Study of the Tohoku Tsunami," *Remote Sensing*, vol. 11, p. 1123, May 2019.

[16] R. H. Fraser, R. Fernandes, and R. Latifovic, "Multi-temporal Mapping of Burned Forest over Canada Using Satellite-based Change Metrics," *Geocarto International*, vol. 18, pp. 37–47, June 2003.

[17] C. Zhang, S. Wei, S. Ji, and M. Lu, "Detecting Large-Scale Urban Land Cover Changes from Very High Resolution Remote Sensing Images Using CNN-Based Classification," *ISPRS International Journal of Geo-Information*, vol. 8, p. 189, Apr. 2019.

[18] C. Cao, S. Dragićević, and S. Li, "Land-Use Change Detection with Convolutional Neural Network Methods," *Environments*, vol. 6, p. 25, Feb. 2019.

[19] D. Lu, "Detection of urban expansion in an urban-rural landscape with multitemporal QuickBird images," *Journal of Applied Remote Sensing*, vol. 4, p. 041880, Sept. 2010.

[20] H. R. Kerner, K. L. Wagstaff, B. D. Bue, P. C. Gray, J. F. Bell, and H. Ben Amor, "Toward Generalized Change Detection on Planetary Surfaces With Convolutional Autoencoders and Transfer Learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, pp. 3900–3918, Oct. 2019.

[21] S. A. Robila, "Use of Remote Sensing Applications and its Implications to the Society," in *2006 IEEE International Symposium on Technology and Society*, (Queens, NY, USA), pp. 1–6, IEEE, June 2006.

[22] S. Sadek, "Engineering impacts of the august 4, 2020 port of beirut, lebanon explosion," *GEER Association (Report - 70)*, 2021.

[23] A. Asokan and J. Anitha, "Change detection techniques for remote sensing applications: a survey," *Earth Science Informatics*, vol. 12, pp. 143–160, June 2019.

[24] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, "Change Detection Based on Artificial Intelligence: State-of-the-Art and Challenges," *Remote Sensing*, vol. 12, p. 1688, May 2020.

[25] L. Khelifi and M. Mignotte, "Deep Learning for Change Detection in Remote Sensing Images: Comprehensive Review and Meta-Analysis," *arXiv:2006.05612 [cs]*, June 2020.

[26] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object Recognition with Gradient-Based Learning," in *Shape, Contour and Grouping in Computer Vision*, vol. 1681, pp. 319–345, Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.

[27] D. Peng, Y. Zhang, and H. Guan, "End-to-End Change Detection for High Resolution Satellite Images Using Improved UNet++," *Remote Sensing*, vol. 11, p. 1382, June 2019.

[28] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change Detection Based on Deep Siamese Convolutional Network for Optical Aerial Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, pp. 1845–1849, Oct. 2017.

[29] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning Spectral-Spatial-Temporal Features via a Recurrent Convolutional Neural Network for Change Detection in Multispectral Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, pp. 924–935, Feb. 2019.

[30] S. Ji, Y. Shen, M. Lu, and Y. Zhang, "Building Instance Change Detection from Large-Scale Aerial Images using Convolutional Neural Networks and Simulated Samples," *Remote Sensing*, vol. 11, p. 1343, June 2019.

[31] F. Nex, D. Duarte, F. G. Tonolo, and N. Kerle, "Structural Building Damage Detection with Deep Learning: Assessment of a State-of-the-Art CNN in Operational Conditions," *Remote Sensing*, vol. 11, p. 2765, Nov. 2019.

[32] H. Jiang, X. Hu, K. Li, J. Zhang, J. Gong, and M. Zhang, "PGA-SiamNet: Pyramid Feature-Based Attention-Guided Siamese Network for Remote Sensing Orthoimagery Building Change Detection," *Remote Sensing*, vol. 12, p. 484, Feb. 2020.

[33] B. Kalantar, N. Ueda, H. A. H. Al-Najjar, and A. A. Halin, "Assessment of Convolutional Neural Network Architectures for Earthquake-Induced Building Damage Detection based on Pre- and Post-Event Orthophoto Images," *Remote Sensing*, vol. 12, p. 3529, Oct. 2020.

[34] H. Miura, T. Aridome, and M. Matsuoka, "Deep Learning-Based Identification of Collapsed, Non-Collapsed and Blue Tarp-Covered Buildings from Post-Disaster Aerial Images," *Remote Sensing*, vol. 12, p. 1924, June 2020.

[35] B. J. Wheeler and H. A. Karimi, "Deep Learning-Enabled Semantic Inference of Individual Building Damage Magnitude from Satellite Images," *Algorithms*, vol. 13, p. 195, Aug. 2020.

[36] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *arXiv:1609.02907 [cs, stat]*, Feb. 2017.

[37] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph Convolutional Networks for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2020.

[38] U. Chaudhuri, B. Banerjee, and A. Bhattacharya, "Siamese graph convolutional network for content based remote sensing image retrieval," *Computer Vision and Image Understanding*, vol. 184, pp. 22–30, July 2019.

[39] S. Saha, F. Bovolo, and L. Bruzzone, "Semisupervised Change Detection Using Graph Convolutional Network," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 4, pp. 607 – 611, 2020.

[40] S. Tilon, F. Nex, N. Kerle, and G. Vosselman, "Post-Disaster Building Damage Detection from Earth Observation Imagery Using Unsupervised and Transferable Anomaly Detecting Generative Adversarial Networks," *Remote Sensing*, vol. 12, p. 4193, Dec. 2020.

[41] D. Peng, L. Bruzzone, Y. Zhang, H. Guan, H. Ding, and X. Huang, "SemiCDNet: A Semisupervised Convolutional Neural Network for Change Detection in High Resolution Remote-Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, pp. 5891–5906, July 2021.

[42] J. Z. Xu, W. Lu, Z. Li, P. Khaitan, and V. Zaytseva, "Building Damage Detection in Satellite Imagery Using Convolutional Neural Networks," *arXiv:1910.06444 [cs, eess, stat]*, Oct. 2019.

[43] V. Benson and A. Ecker, "Assessing out-of-domain generalization for robust building damage detection," *arXiv:2011.10328 [cs]*, Nov. 2020.

[44] W. Yang, X. Zhang, and P. Luo, "Transferability of Convolutional Neural Network Models for Identifying Damaged Buildings Due to Earthquake," *Remote Sensing*, vol. 13, p. 504, Jan. 2021.

[45] Z. Zheng, Y. Zhong, J. Wang, A. Ma, and L. Zhang, "Building damage assessment for rapid disaster response with a deep object-based semantic change detection framework: From natural disasters to man-made disasters," *Remote Sensing of Environment*, vol. 265, p. 112636, Nov. 2021.

[46] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," *arXiv:1706.02216 [cs, stat]*, Sept. 2018.

[47] "Panchromatic imagery and its band combination in remote sensing," Nov 2020.

[48] G. Vivone, L. Alparone, J. Chanussot, M. Dalla Mura, A. Garzelli, G. A. Licciardi, R. Restaino, and L. Wald, "A Critical Comparison Among Pansharpening Algorithms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 2565–2586, May 2015.

[49] C. Toth and G. Jóźków, "Remote sensing platforms and sensors: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 22–36, May 2016.

[50] J. Mallick, Y. Kant, and B. D. Bharath, "Estimation of land surface temperature over Delhi using Landsat-7 ETM+," *The Journal of Indian Geophysical Union*, p. 10, 2008.

[51] A. Lowe, N. Harrison, and A. P. French, "Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress," *Plant Methods*, vol. 13, p. 80, Dec. 2017.

[52] A. P. Cracknell, "The development of remote sensing in the last 40 years," *International Journal of Remote Sensing*, vol. 39, pp. 8387–8427, Dec. 2018.

[53] K. Anderson and K. J. Gaston, "Lightweight unmanned aerial vehicles will revolutionize spatial ecology," *Frontiers in Ecology and the Environment*, vol. 11, pp. 138–146, Apr. 2013.

[54] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 152, pp. 166–177, June 2019.

[55] Y. Jiang, P. Fu, and Q. Weng, "Assessing the Impacts of Urbanization-Associated Land Use/Cover Change on Land Surface Temperature and Surface Moisture: A Case Study in the Midwestern United States," *Remote Sensing*, vol. 7, pp. 4880–4898, Apr. 2015.

[56] J. Zhou, Y. Chen, J. Wang, and W. Zhan, "Maximum Nighttime Urban Heat Island (UHI) Intensity Simulation by Integrating Remotely Sensed Data and Meteorological Observations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, pp. 138–146, Mar. 2011.

[57] R. M. Murali, "Applications of remote sensing and Geographic Information System (GIS) in Archaeology," *Recent Researches on Indus Civilization & Maritime Archaeology in India*, p. 9, 2015.

[58] B. Xiong, J. M. Chen, and G. Kuang, "A change detection measure based on a likelihood ratio and statistical properties of SAR intensity images," *Remote Sensing Letters*, vol. 3, pp. 267–275, May 2012.

[59] P. He, W. Shi, H. Zhang, and M. Hao, "A novel dynamic threshold method for unsupervised change detection from remotely sensed images," *Remote Sensing Letters*, vol. 5, pp. 396–403, Apr. 2014.

[60] R. B. Lorena, J. Santos, Y. Shimabukuro, I. Brown, and H. Kux, "A change vector analysis technique to monitor land use/land cover in sw brazilian amazon: Acre state," in *International Society for Photogrammetry and Remote Sensing*, 2002.

[61] S. Liu, L. Bruzzone, F. Bovolo, M. Zanetti, and P. Du, "Sequential Spectral Change Vector Analysis for Iteratively Discovering and Detecting Multiple Changes in Hyperspectral Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 4363–4378, Aug. 2015.

[62] L. Ke, Y. Lin, Z. Zeng, L. Zhang, and L. Meng, "Adaptive Change Detection With Significance Test," *IEEE Access*, vol. 6, pp. 27442–27450, 2018.

[63] V. Sadeghi, F. Farnood Ahmadi, and H. Ebadi, "Design and implementation of an expert system for updating thematic maps using satellite imagery (case study: changes of Lake Urmia)," *Arabian Journal of Geosciences*, vol. 9, p. 257, Apr. 2016.

[64] C. Massarelli, "Fast detection of significantly transformed areas due to illegal waste burial with a procedure applicable to Landsat images," *International Journal of Remote Sensing*, vol. 39, pp. 754–769, Feb. 2018.

[65] R. Vázquez-Jiménez, R. Romero-Calcerrada, C. J. Novillo, R. N. Ramos-Bernal, and P. Arrogante-Funes, "Applying the chi-square transformation and automatic secant thresholding to Landsat imagery as unsupervised change detection methods," *Journal of Applied Remote Sensing*, vol. 11, p. 016016, Feb. 2017.

[66] H. Zhuang, K. Deng, Y. Yu, and H. Fan, "An approach based on discrete wavelet transform to unsupervised change detection in multispectral images," *International Journal of Remote Sensing*, vol. 38, pp. 4914–4930, Sept. 2017.

[67] Z. Liu, G. Li, G. Mercier, Y. He, and Q. Pan, "Change Detection in Heterogenous Remote Sensing Images via Homogeneous Pixel Transformation," *IEEE Transactions on Image Processing*, vol. 27, pp. 1822–1834, Apr. 2018.

[68] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, Apr. 2016.

[69] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, pp. 247–259, May 2011.

[70] J. Im and J. Jensen, "A change detection model based on neighborhood correlation image analysis and decision tree classification," *Remote Sensing of Environment*, vol. 99, pp. 326–340, Nov. 2005.

[71] H. Nemmour and Y. Chibani, "Multiple support vector machines for land cover change detection: An application for mapping urban extensions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 61, pp. 125–133, Nov. 2006.

[72] C. L. Zitnick and P. Dollár, "Edge Boxes: Locating Object Proposals from Edges," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), vol. 8693, pp. 391–405, Cham: Springer International Publishing, 2014.

[73] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *arXiv:1409.0575 [cs]*, Jan. 2015.

[74] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *arXiv:1409.4842 [cs]*, Sept. 2014.

[75] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A Nested U-Net Architecture for Medical Image Segmentation," *arXiv:1807.10165 [cs, eess, stat]*, July 2018.

[76] R. C. Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Urban Change Detection for Multispectral Earth Observation Using Convolutional Neural Networks," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, (Valencia), pp. 2115–2118, IEEE, July 2018.

[77] H. Lyu, H. Lu, and L. Mou, "Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection," *Remote Sensing*, vol. 8, p. 506, June 2016.

[78] H. Chen, C. Wu, B. Du, L. Zhang, and L. Wang, "Change Detection in Multisource VHR Images via Deep Siamese Convolutional Multiple-Layers Recurrent Neural Network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, pp. 2848–2864, Apr. 2020.

[79] R. Jing, S. Liu, Z. Gong, Z. Wang, H. Guan, A. Gautam, and W. Zhao, "Object-based change detection for VHR remote sensing images based on a Trisiamese-LSTM," *International Journal of Remote Sensing*, vol. 41, pp. 6209–6231, Aug. 2020.

[80] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015.

[81] S. Ji, S. Wei, and M. Lu, "Fully Convolutional Networks for Multisource Building Extraction From an Open Aerial and Satellite Imagery Data Set," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, pp. 574–586, Jan. 2019.

[82] Q. Wang, X. Zhang, G. Chen, F. Dai, Y. Gong, and K. Zhu, "Change detection based on Faster R-CNN for high-resolution remote sensing images," *Remote Sensing Letters*, vol. 9, pp. 923–932, Oct. 2018.

[83] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *arXiv:1506.01497 [cs]*, Jan. 2016.

[84] E. Weber and H. Kané, "Building Disaster Damage Assessment in Satellite Imagery with Multi-Temporal Fusion," *arXiv:2004.05525 [cs]*, Apr. 2020.

[85] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[86] J. Su, Y. Bai, X. Wang, D. Lu, B. Zhao, H. Yang, E. Mas, and S. Koshimura, "Technical Solution Discussion for Key Challenges of Operational Convolutional Neural Network-Based Building-Damage Assessment from Satellite Imagery: Perspective from Benchmark xBD Dataset," *Remote Sensing*, vol. 12, p. 3808, Nov. 2020.

[87] Y. Bai, J. Hu, J. Su, X. Liu, H. Liu, X. He, S. Meng, E. Mas, and S. Koshimura, "Pyramid Pooling Module-Based Semi-Siamese Network: A Benchmark Model for Assessing Building Damage from xBD Satellite Imagery Datasets," *Remote Sensing*, vol. 12, p. 4055, Dec. 2020.

[88] R. Gupta and M. Shah, "RescueNet: Joint Building Segmentation and Damage Assessment from Satellite Imagery," *arXiv:2004.07312 [cs, eess]*, Apr. 2020. arXiv: 2004.07312.

[89] C. Pati, A. K. Panda, A. K. Tripathy, S. K. Pradhan, and S. Patnaik, "A novel hybrid machine learning approach for change detection in remote sensing images," *Engineering Science and Technology, an International Journal*, vol. 23, pp. 973–981, Oct. 2020.

[90] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, p. 11, Dec. 2019.

[91] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-Label Image Recognition With Graph Convolutional Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Long Beach, CA, USA), pp. 5172–5181, IEEE, June 2019.

[92] N. Khan, U. Chaudhuri, B. Banerjee, and S. Chaudhuri, "Graph convolutional network for multi-label VHR remote sensing scene recognition," *Neurocomputing*, vol. 357, pp. 36–46, Sept. 2019.

[93] L. Mou, X. Lu, X. Li, and X. X. Zhu, "Nonlocal Graph Convolutional Networks for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–12, 2020.

[94] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," *International Journal of Computer & Information Sciences*, vol. 9, pp. 219–242, June 1980.

[95] M. Calderisi, G. Galatolo, I. Ceppa, T. Motta, and F. Vergentini, "Improve Image Classification Tasks Using Simple Convolutional Architectures with

Processed Metadata Injection," in *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, (Sardinia, Italy), pp. 223–230, IEEE, June 2019.

[96] J. S. Ellen, C. A. Graff, and M. D. Ohman, "Improving plankton image classification using context metadata," *Limnology and Oceanography: Methods*, vol. 17, pp. 439–461, Aug. 2019.

[97] A. Bucknell, "High contrast: We chart beirut's ever-changing architecture scene," Aug 2018.

[98] "Lebanon: Beirut port explosions - aug 2020."

[99] S. E. Rigby, T. J. Lodge, S. Alotaibi, A. D. Barr, S. D. Clarke, G. S. Langdon, and A. Tyas, "Preliminary yield estimation of the 2020 Beirut explosion using video footage from social media," *Shock Waves*, Sept. 2020.

[100] A. Krayem, A. Yeretzian, G. Faour, and S. Najem, "Machine learning for buildings' characterization and power-law recovery of urban metrics," *PLOS ONE*, vol. 16, p. e0246096, Jan. 2021.

[101] S. Morbieu, "Accuracy: From classification to clustering evaluation," Jun 2019.

[102] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[103] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, 1908.

[104] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[105] M. Fey and J. E. Lenssen, "Fast Graph Representation Learning with PyTorch Geometric," *arXiv:1903.02428 [cs, stat]*, Apr. 2019.

[106] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[107] G. King and L. Zeng, "Logistic regression in rare events data," *Political Analysis*, vol. 9, pp. 137–163, 2001.

[108] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[109] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.