Titre: Title:	Building Damage Assessment After a Natural Disaster in Emergency Contexts: A Deep Learning Approach
Auteur: Author:	Isabelle Bouchard
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Bouchard, I. (2021). Building Damage Assessment After a Natural Disaster in Emergency Contexts: A Deep Learning Approach [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/9470/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/9470/
Directeurs de recherche: Advisors:	Daniel Aloise, & Marie-Ève Rancourt
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Building Damage Assessment After A Natural Disaster In Emergency Contexts: A Deep Learning Approach

ISABELLE BOUCHARD
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Octobre 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Building Damage Assessment After A Natural Disaster In Emergency Contexts: A Deep Learning Approach

présenté par **Isabelle BOUCHARD**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Guillaume-Alexandre BILODEAU, président
Daniel ALOISE, membre et directeur de recherche
Marie-Ève RANCOURT, membre et codirectrice de recherche
Sarath Chandar ANBIL PARTHIPAN, membre

ACKNOWLEDGEMENTS

Research can be a lonely process and having support is essential. I consider myself lucky for working along with such great collaborators and supervisors.

Marco and Thierry, thank you for onboarding on this project with me, and steadily meet every two weeks to provide your honest and important feedback. Being connected to an important humanitarian organization such as the World Food Program has given me the motivation to really ground the project in the emergency context.

Daniel, thank you for your kind and reassuring words the many times I lost focus to help me get back on track.

Marie-Ève, thank you for your trust, honesty and dedication. I am grateful you embarked on this project with me, from seeking collaborators to defining the project.

Freddie, thank you for being so interested and interesting. Your deep technical knowledge was the missing piece in this project, and I am indefinitely thankful you so closely supervised my work. Thank you for your time, and your patience, during this whole year working together.

Finally, I will never forget the past year I worked on this thesis project as the COVID19 pandemic added a unique dimension to it. More than ever, the support of close friends and family was crucial.

Philou, thank you for being the most amazing confinement partner, but foremost the best life partner. Thank you for challenging me, and for listening to my excitements and my disappointments.

Friends and family, I love you.

RÉSUMÉ

Les catastrophes naturelles peuvent être extrêmement dévastatrices, autant sur le plan humain que matériel. Trop souvent, ces événements laissent derrière eux des communautés en situation précaires. Les organisations humanitaires agissent en première ligne: elles doivent intervenir le plus rapidement possible. Toutefois, opérer dans des zones dévastées peut est assurément difficile à coordoner et peut d'autant plus s'avérer dangereux pour les travailleurs de terrain. L'imagerie satellite offre une alternative à faible risque pour évaluer la situation sur le terrain avant d'y déployer des ressources. Cependant, les images satellites couvrent souvent de larges superficies et leur traitement manuel pour identifier les zones affectées tendent à générer de longs délais. Ainsi, dans ce projet, nous proponsons une approche basée sur l'apprentissage machine pour accélérer l'évaluation des dommages aux suites d'une catastrophe naturelle. Plus spécifiquement, nous avons conçu un système de réseaux convolutifs profonds pour la détection des bâtiments endommagés à partir d'images satellites qui utilisent des techniques de transfert d'apprentissage pour réduire le temps d'exécution. En résulte un système adapté pour les situation d'urgence.

ABSTRACT

Natural disasters can be devastating: they may cause the loss of life and major damages to properties and infrastructures. Too often, they lead the way to precarity in the affected communities. Humanitarian organizations are in the frontline and, as such, they must intervene without delay. However, operating in devastated areas can be hazardous for field workers and is assuredly chaotic. Remote sensing imagery enables for a low-risk ground assessment that can be done prior to deploying resources on the field. Yet, satellite images often cover large areas and their manual processing to identify affected regions result lengthy delays. Thereby, in this work, we propose a Machine Learning (ML) approach to support and speed up the damage assessment workflow in the aftermath of a natural disaster. More specifically, we design a system based on Convolutional Neural Networks (CNN) to detect damaged buildings from satellite imagery and experiment with transfer learning techniques to shorten the runtime. The result is an end-to-end machine learning workflow for emergency context for humanitarians to detect damaged buildings in the shortest delays after a natural disaster.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

CNN	Convolutional Neural Network
DL	Deep Learning
GIS	Geospatial Information Systems
ML	Machine Learning
NN	Neural Networks
SiameseNet	Siamese Neural Network
UN	United Nations
UAV	Unmanned Aerial Vehicle
VHR	Very High Resolution
WFP	World Food Program

CHAPTER 1 INTRODUCTION

Humanitarian impacts of natural disasters can be catastrophic. Not only the events directly occasion casualties, but material damages also open onto precariousness for the affected communities. Marginalized populations tend to feel stronger and more long-term impacts of the event, which accentuates pre-existing inequalities. Moreover, climate changes are likely to influence the intensity and the frequency of extreme natural events.

Technologies are leveraged to mitigate the impact of environmental phenomenons. Satellite imagery can help solve the scarcity of information on the ground required for an effective disaster relief response. It can be used to visualize the devastated area and identify ravages and affected populations. For instance, damaged buildings can be identified from satellite images. Yet, in emergency context, processing the immense amount of data hinders the deployment of resources on the field. The goal of this thesis is to develop tools to facilitate the processing of satellite images in an emergency context and help minimize the devastating impact of natural disasters on affected populations.

Machine learning algorithms have the ability to treat and analyze large datasets to support human decisions. More specifically, convolutional neural networks (CNN) can process images to extract relevant information. In this work, we study the use of CNN to identify damaged buildings from satellite images after a natural disaster. Damaged building identification can be used to locate the most severely affected populations and inform the emergency relief response. For the approach to be used in emergency contexts, it is crucial for the solution to take both data scarcity and time constraints into account. Post-disaster data can only be accessed with some delays ranging from hours to days after the event and its distribution cannot be known prior to that. Therefore, we develop a complete emergency building damage assessment workflow based on machine learning.

CNN learns by examples: they are trained to recognize specific patterns on large amounts of images to then infer predictions on a new set of images sampled from the same distribution. Ideally, a model trained on a collection of historical natural disaster satellite images could be used to identify damages on a forthcoming event. However, in practice, given the unpredictability of natural disasters, images from forthcoming events are not guaranteed to be in-distribution. Hence, in this work, we also investigate the ability of CNN to generalize to images from an unseen disaster event. We train models to detect damages to building on historical data and test on a new disaster event; this represents our baseline. Then, assuming a poor generalizability, we study the use of standard transfer learning techniques to

adapt the model on a new disaster data distribution. The idea is to first train the model on historical data, and then fine-tune the model on disaster-specific images. These techniques allow for the bulk of processing and training to be done prior to the event so that only rapid adjustments need to be made after the event. In fact, because they depend on post-disaster data, adjustments can only be made after the event.

As such, we experiment with two model fine-tuning approaches: supervised and unsupervised. In the former approach, manual annotation is required to annotate samples from the new disaster event. As data annotation is time-costly, only a limited number of images is annotated. Conversely, in the latter approach, no annotation is required. It is expected to be faster than supervised fine-tuning, but it assumes that sufficient signal can be captured from the baseline model.

1.1 Collaboration

This project has been done with the support and collaboration of the World Food Program (WFP) geospatial information system (GIS) team. They were initially involved in the definition of the project scoped, and provided consistent feedback on the results and progress made to ensure an alignment of the solution with the humanitarian practice. This collaboration fostered a strong focus on the emergency context.

1.2 Research Objective

The research objective is to develop a model to automatically detect damaged buildings after a natural disaster that can be deployed in an emergency context. That goes by three research sub-objectives:

- Assessing the feasibility of damaged building detection using deep learning techniques
- Designing a workflow (including data gathering, annotation, model training and evaluation) applicable in crisis relief situations
- Implementing such a model and evaluating its performance.

1.3 Thesis Overview

The thesis is organized as follows. First, we introduce the machine learning fundamentals in Chapter 2. Then, we present the manuscript submitted to Expert System with Applications

in Chapter 3. It contains a presentation of the problematic from a humanitarian point of view (Section 3.2) and related work (Section 3.3), an overview of the dataset (Section 3.5) and a series of experiments (Section 3.6 and 3.7). The manuscript experiments focus on the model baselines trained on previous disaster events, as well as supervised transfer learning experiments. Then, in Chapter 4, we present follow up unsupervised transfer learning experiments. Finally, in Chapter 5, we discuss the approach and avenues for future work.

CHAPTER 2 LITERATURE REVIEW

This chapter presents a brief overview of ML fundamentals and explains the main theoretical concepts behind this work. This chapter is complemented by the section Machine Learning Fundamentals of the article introduced in section 3. This chapter is inspired by the textbook Deep Learning [2].

2.1 Basic Concepts

2.1.1 Machine Learning

Machine Learning algorithms parse data and learn from it to make predictions without relying on rule-based programming. We get machine to learn a function f by training them on a defined set of observations D. Such observations may be feature vectors, images, text inputs, etc. There exist different machine learning paradigms: supervised, unsupervised, semi-supervised, and more.

Supervised Learning

In supervised learning, the dataset D is composed of N data points $x(t)$ and targets $y(t)$, i.e. that $D = \{(x(t), y(t))|t = 1 \dots N\}$. The algorithm learns a function f that maps the distribution of data points X to the target Y.

$$Y = f(X)$$

The targets Y help the algorithm extract patterns from X. For regression problems, targets are real numbers; for classification, a class index (from 1 to the number of classes). For instance, to classify cats' and dogs' photos, $x(t)$ would be an image and $y(t)$ the corresponding category, either cat or dog.

Unsupervised Learning

Unsupervised learning is a family of algorithms that look for patterns in the data points $x(t)$, in the absence of an explicit target to predict, i.e. that $D = \{x(t)|t = 1 \dots N\}$. Such algorithms include clustering methods to partition a dataset D into groups of similar attributes or dimensionality reduction methods to extract a lower-dimension representation of data.

Semi-supervised Learning

As the name says, semi-supervised techniques are employed when the annotation is only available for a portion of the whole dataset. Semi-supervised algorithms leverage both annotated and non-annotated data.

2.1.2 Parametric Models

In this work, we only consider parametric models. Such models have a fixed number of parameters and do not increase twitch training data. For parametric models, learning means adjusting the parameters to fit the training data. The learned parameters a model contains are defined by θ; a parametric model is thus described by $f(x(t), \theta)$.

The simplest parametric model is linear regression.

$$y'(t) = f(x(t); \theta) = ax(t) + b, \qquad \theta = [a, b]$$

Neural networks, too, are parametric models. The number of parameters of a model defines its capacity. To fit large-scale training data, neural networks are constituted of millions of parameters.

Empirical Risk Minimization

Loss functions measure the error between the outcome predicted by the model and the actual target. Depending upon the task and the specificity of the problem, the loss function can be designed to influence the result. For instance, simple regression problems can be solved using the least square error as the loss function.

$$l[f, (x(t), y(t))] = [y(t) - f(x(t), \theta)]^2$$

If f can describe X well, the output will be close to Y, and the loss will be low, and vice versa.

Parametric models are solved by minimizing the empirical risk. The true risk is defined as the expectation of the loss function over samples drawn from a distribution P. In practice, we measure the empirical risk, given the finite number of samples from a dataset D.

$$R(f, D) = \underset{(x(t),y(t))\in D}{E}[l(f, (x(t), y(t)))] = \frac{1}{T}\sum_{i=1}^{N} l[f, (x(i), y(i))]$$

For parametric models, minimizing the empirical risk goes by finding the parameters theta that minimize the loss function on our data. is the optimal set of parameters for a given dataset D and loss function l.

$$\hat{\theta} = argmin_{\theta} \frac{1}{T} \sum_{i=1}^{N} l[f, (x(i), y(i))]$$

For simple models such as linear models, the analytical solution is tractable; hence finding $\hat{\theta}$ is straightforward. However, as the number of parameters grows, the analytical solutions can't be found, and we use the gradient descent procedure.

Gradient descent

Gradient descent is an iterative procedure to find the optimal model parameters. It assumes a loss function that is differentiable with respect to the model's parameters.

The iterative procedure is two-fold. First, it consists of computing the gradient of the loss function with respect to the parameters and, then, taking a step in the gradient's opposite direction (see Figure 2.1 and Algorithm 1). The step size depends on the learning rate α. Effectively, it adjusts the parameters in the direction that reduce the error.

For neural networks with large numbers of parameters, gradient descent is computationally expensive. Therefore, in practice, parameters are updated after computing the gradient on mini-batches of randomly sampled examples. When the mini-batch is composed of only one sample, the procedure is called stochastic gradient descent.

Initialize θ

for N iterations **do**

for mini-batch in D_{train} **do** $\Delta = -\frac{1}{T} \nabla_{\theta} \, l[f, (x(t), y(t))]$

$\theta \leftarrow \theta + \alpha\Delta$

Algorithm 1 Mini-batch gradient descent algorithm

Typically, a model f is trained on a training set D_{train} to be used on other inputs from the same distribution. It is assumed that samples are independent and identically distributed.

$D_{validation}$ is the validation set; it represents a subset of samples drawn for the same distribution used to fine-tune hyper-parameters. Hyper-parameters are any parameters associated with a model that is not learned, for instance, the mini-batch size or the gradient descent procedure's learning rate.

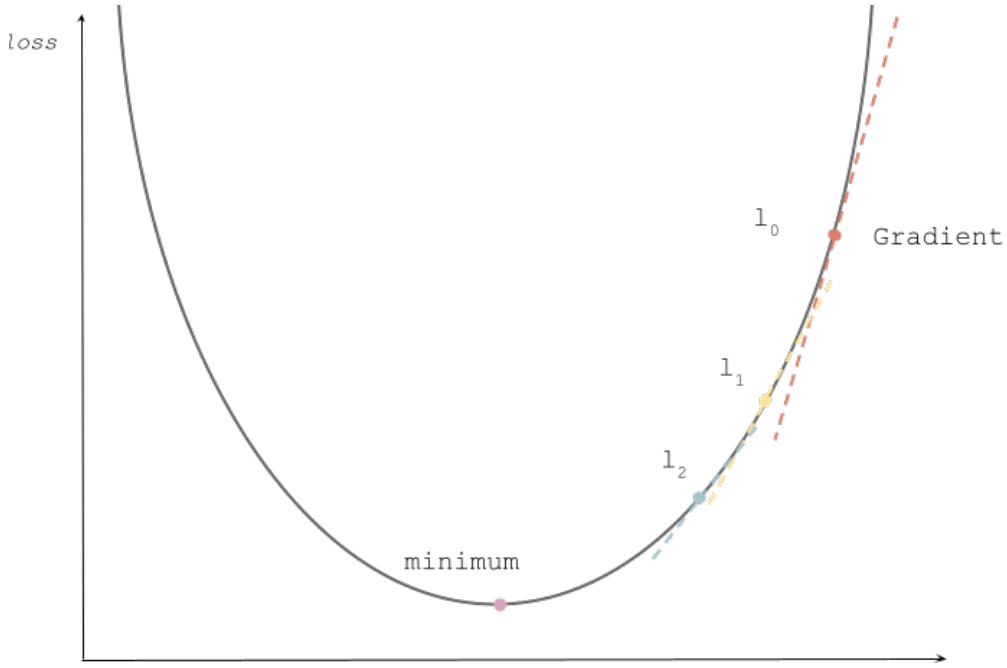

Figure 2.1 Gradient descent procedure where l_0, l_1 and l_2 show three steps of the iterative process.

Finally, D_{test} is the testing set used to measure the generalization error, i.e. how well the learned function f behaves on unseen data points. Ultimately, with enough parameters, a model could memorize all training samples from D_{train}; however, this would result in a substantial generalization error on D_{test}. This phenomenon is called overfitting and can be avoided by reducing the model capacity of a model. Conversely, underfitting is when the generalization error could be improved with more capacity to describe the inputs.

2.1.3 Neural Networks

Artificial Neurons

Neural networks base unit is the artificial neuron, composed of a set of weights w_i and a bias b (Figure 2.2). These are learnable parameters. The artificial neuron applies a linear combination of the input x with the weights and bias, followed by a non-linear activation function g, such as *sigmoid*, *tanh* or *ReLU*. It allows for the artificial neuron to solve linearly separable problems. The following is the mathematical representation of the artificial neuron:

$$h(x) = g(a(x)) = g(b + \sum_{i=0}^{d} w_i x_i)$$

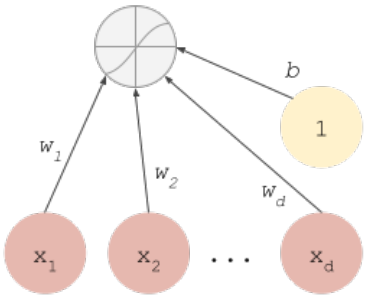

Figure 2.2 Single artificial neuron.

Multilayer Neural Networks

The artificial neuron is assembled in layers, which are stacked to form neural networks. Typically, each artificial units in a layer is connected to those of the following layer (Figure 2.3), an arrangement called fully-connected layer. W(t) refers to the weights matrix of a given layer and b(t) its bias vector.

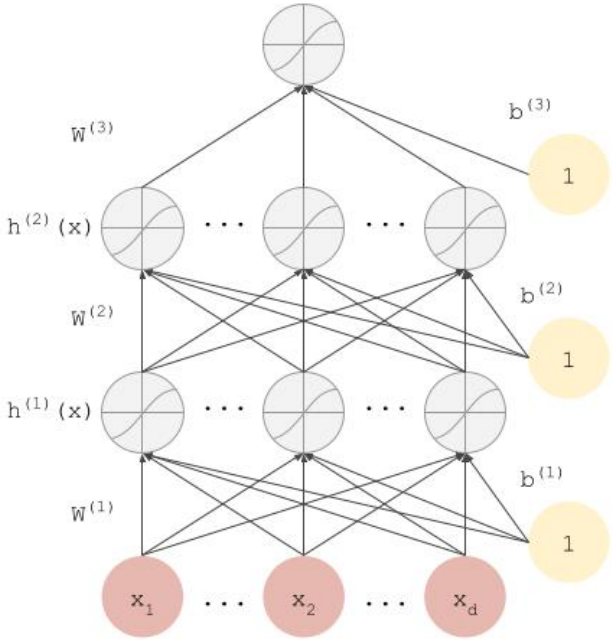

Figure 2.3 Multilayers neural network. W(t) represents the ensemble of weights of a given layer, b(t) the ensemble of bias.

The output activation function is specific to the task to solve. For instance, for binary classification, *sigmoid* activation bounds the output between 0 and 1; for multi-classes classification, *softmax* normalizes the output to a probability distribution over predicted classes.

Neural networks are optimized through empirical risk minimization and stochastic gradient descent. The gradient of the loss with respect to the parameters can be computed thanks to the chain rule during a so-called forward pass. Gradients are memorized, and the backward pass consists of back-propagating the error through the network's parameters. Thus, neural networks become computationally expensive as the number of parameters grows.

In theory, the Universal Approximation Theorem says that neural networks can approximate any distribution: a single hidden layer neural network with a linear output unit can approximate any continuous function arbitrarily well, given enough units. However, in practice, they can be tricky to optimize.

Convolutional Neural Network

Convolutional Neural Networks (CNN) are neural network specialized in the ingestion of data with a known topology such as images (2D topology) or time series (1D topology). They are specifically designed to make use of that topology. In this work, we will focus on CNN as applied to image inputs.

CNN are based on the convolution operation, which can be seen as the application of a filter described by a kernel k to an input x producing a feature map.

A convolution is the simple application of a filter to an input that results in an activation. The following is the result of a discrete convolution on an image x at position ij with a kernel k. Here, p and q represent the kernel size.

$$(x * k)_{ij} = \sum_{p} \sum_{q} w_{i+p,j+q} \times k_{r-p,r-q}$$

Convolution can be applied with predefined weights, think Sobel filter for instance. Conversely, in convolution neural network, the kernel values are initialized randomly and gradually adjusted during training to better fit the input data: the values are learned. In practice, the discrete convolutions are computed as sparse matrix multiplications.

Convolutional layers leverage three concepts: local connectivity, parameter sharing and equivariant representations.

In traditional neural networks, each unit of a layer is connected to every subsequent layer unit. The idea of local connectivity for CNN is to have sparse connectivity such that each unit is connected to a small region of the input only, known as a neuron's receptive field. This effectively means that the kernels are smaller than the image.

Local connectivity allows for the detection of small features in the image. It also limits the

number of parameters that would quickly explode with traditional fully connected layers. Parameter sharing refers to the concept of applying the same kernel at every position in the image. The same feature is extracted at every position, and the output is called feature maps (Figure 2.4).

Ultimately, with enough training, convolutional kernels become experts at extracting some features from images, like edges or some specific textures. The deeper the network, the more high-level the features the network can extract by assembling low-level features.

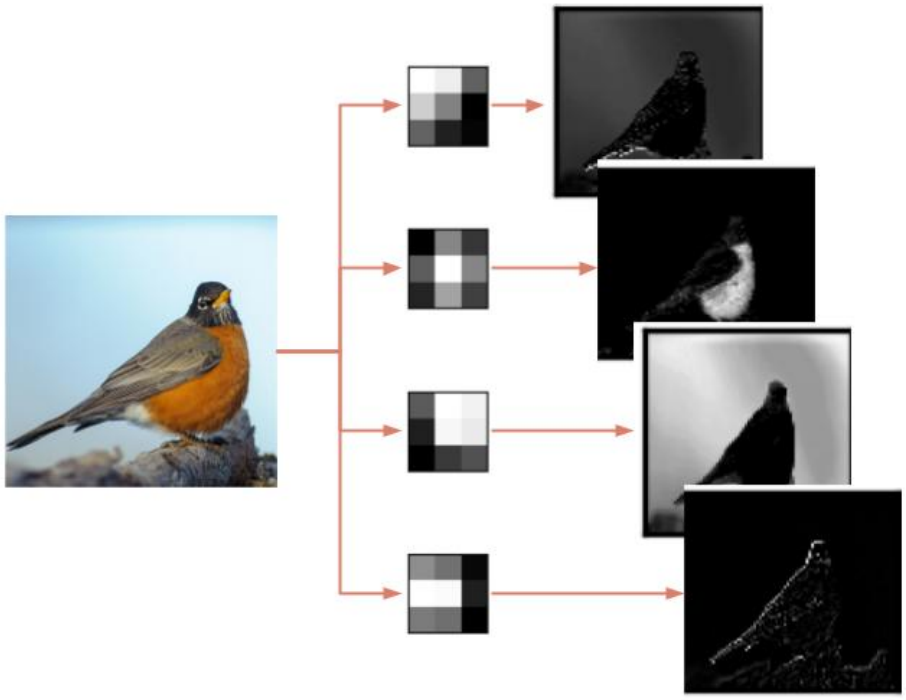

Figure 2.4 Parameters sharing. The same kernel is applied at every position in the image to produce feature maps.

Parameter sharing significantly reduces the number of parameters. It also causes CNN to be equivariant to translation, i.e. that the output does not change under a translation of the input. CNNs are generally constituted of a stack of convolutional layers, followed by activation functions and pooling layers.

Pooling

Pooling layers downsample or upsample input image or feature maps. Statistics over input neighbours define the output values. For instance, Figure 2.5 shows a 2x2 maximum pooling layer, i.e. that only the maximum value out of a 2x2 neighbourhood is kept. Pooling layers

also help small translation invariance. Pooling also allows for dealing with the input of varying sizes.

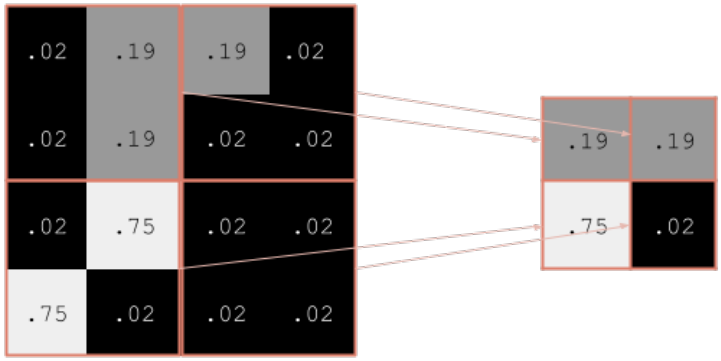

Figure 2.5 Max pooling example.

2.2 Related Works

In this section, we present reviewed works in damage assessment using machine learning techniques. Cooner and al. [3] were the first to apply Machine Learning techniques (namely Feed Forward Neural Networks and Random Forests) to the task of building damage assessment. They trained and evaluated their approach on satellite images of urban areas after the 2010 Haiti earthquake. In 2017, Fujita and al. [4] instead used Convolutional Neural Networks, specialized in image data. They trained and evaluated their method on pre- and post-disaster satellite images of the 2011 Japan earthquake and tsunami, which they released as the *ABCD* dataset. Sublime et al. [5] studied the same event using change detection techniques. Similarly, Doshi and Al. [6] developed a damage detection model inspired by change detection methods to produce segmentation maps of devastated areas. Instead of focusing on solely one event, they leveraged two newly released datasets for building and road detection: SpaceNet [7] and DeepGlobe [8], to develop a building damage detection model. More recently, Xu et al [9] studied the transferability of the learned models to new disaster events. They trained their model on images from 2010 Haiti, 2017 Mexico City, and 2018 Indonesia earthquakes and evaluated using unseen disaster images.

In 2019, Gupta et al. [1] released the xBD dataset for building damage assessment after natural disasters. That large database contains image pairs of before and after 18 different natural disaster events, covering a variety of locations around the globe, diverse disaster types as well as different constructions and climates. Each image pair comes along with buildings location and annotation on a scale of four ordinal classes: `No damage`, `Minor damage`, `Major`

`damage` and `Destroyed`. Durnov [10] developed a two-step modeling approach composed of a building detector and a damage classifier - he won the competition organized by the dataset creators.

Since then, Shao et al. [11] evaluated different loss functions to achieve better performance on the xBD dataset. Gupta et al. [12] and Weber and al. [13] both presented end-to-end per-pixel classification models with multi-temporal fusion to solve the task. Moreover, Hao and al. [14] studied the use of self-attention to capture long-range information while Shen et al. [15] investigated different methods to fuse pre- and post-disaster feature maps. Finally, a study conducted by Boin et al. [16] reported imbalance issues within the xBD dataset and proposed to upsample challenging classes to help resolve them.

Valentin et al. [17] studied CNN performance to detect damaged buildings on individual disaster events from the xBD dataset. They discussed in their paper the importance of having a model that generalizes to new disaster events under operational conditions. Along the same line, Benson et al. [18] framed the damage assessment task as an out-of-domain distribution problem and applied two techniques to help solve it: multi-domain AdaBN [19] and Stochastic Weight Averaging [20]. In order to leverage unlabeled data, Lee et al. [21] successfully applied semi-supervised techniques, namely FixMatch [22], to the damage assessment task. They studied three natural disasters, the 2010 Haiti earthquake, and the 2018 Santa Rosa wildfire, as well as the war-destroyed region of Aleppo in 2016. Finally, Next et al. [23] studied the operational requirements of damage assessment models in terms of computational runtime and transferability for both satellite and drone images.

2.3 Discussion

Like most Deep Learning applications, the development of models for building damage assessment has been limited by data. That is why most works before 2019 were conducted as case studies on isolated events. As such, the scope of these studies [3–6, 9] is restricted and the conclusion remain specific and hardly transferable.

The release of the xBD dataset [1] undoubtedly encouraged research in the field and many studies were published not long after its creation [10–16]. If these studies can more easily be compared thanks to the common train and test sets, we argue that the dataset split does not reflect the operational context. Images from all 18 disaster events are randomly split into train and or test set, resulting in sets with identical distributions. However, in an emergency context, the test set is a completely unseen disaster event, with a possibly different distribution. Therefore, we argue that the standard dataset splits are inadequate to

measure the ability of a model to generalize to a new disaster event in an emergency context.

Some pieces of work rearranged the xBD dataset to adequately measure the expected performance of machine learning models in emergency contexts [17, 18, 21, 23]. To guarantee that the train and test set are completely independent, the evaluation is done on images from disaster events that were not seen by the model during training.

In our work, we propose to split train and test sets as such: images from a single disaster event are used for testing and remaining images from the other 17 disaster events are used for training. We run an extensive ablation study overall on 18 disaster events for testing, resulting in 18 corresponding training sets. To our knowledge, this is the most extensive study to evaluate the model's ability to generalize to unseen disasters. It uncovers event-specific challenges and clearly shows the limitations of applying machine learning techniques for damage assessment in an emergency context.

Moreover, not only do we optimize for the model performance in terms of accuracy, but we also evaluate the applicability in the emergency context. We design a realistic operational workflow that accounts for data annotation, model training, and inference. We show that our approach can effectively shorten the turnaround time to deploy resources on the field after a natural disaster.

CHAPTER 3 ON TRANSFER LEARNING FOR BUILDING DAMAGE ASSESSMENT FROM SATELLITE IMAGERY IN EMERGENCY CONTEXTS

When a natural disaster occurs, humanitarian organizations need to be prompt, effective, and efficient to support people whose security is threatened. Satellite imagery offers rich and reliable information to support expert decision making, yet its annotation remains labour-intensive and tedious. In this work, we evaluate the applicability of convolutional neural networks (CNN) in supporting building damage assessment in an emergency context. Despite data scarcity, we develop a Deep Learning workflow to support humanitarians in time-constrained emergency situations. To expedite decision-making and take advantage of the inevitable delay to receive post-disaster satellite images, we decouple building localization and damage classification tasks into two isolated models. We show the complexity of the damage classification task and use established transfer learning techniques to fine-tune the model learnings and estimate the minimal number of annotated samples required for the model to be functional in operational situations.

3.1 Introduction

For decades, humanitarian agencies have been developing robust processes to respond effectively when natural disasters occur. As soon as the event happens, processes are triggered, and resources are deployed to assist and relieve the affected population. Nevertheless, from a hurricane in the Caribbean to a heavy flood in Africa, every catastrophe is different, thus requiring organizations to adapt within the shortest delay to support the affected population on the field. Hence, efficient yet flexible operations are essential to the success of humanitarian organizations.

Humanitarian agencies can leverage machine learning to automate traditionally labour-intensive tasks and speed up their crisis relief response. However, to assist decision making in an emergency context, humans and machine learning models can be no different; they both need to adjust quickly to the new disaster. Climate conditions, construction types, and types of damage caused by the event may differ from those encountered in the past. Nonetheless, the response must be sharp and attuned to the current situation. Hence, a model must learn from past disaster events to understand what damaged buildings resemble, but it should first and foremost adapt to the environment revealed by the new disaster.

Damage assessment is the preliminary evaluation of damage in the event of a natural disaster, intended to inform decision-makers on the impact of the incident [24]. This work focuses on the *building damage assessment*. Damaged buildings are strong indicators of the humanitarian consequences of the hazard: they mark *where* people need immediate assistance. In this work, we address building damage assessment using machine learning techniques and remote sensing imagery. We train a neural network to automatically locate buildings from satellite images and assess any damages.

Intuitively, the task might seem straightforward. Ideally, given the emergency context, a model trained on images of past disaster events should be able to generalize to images from the current one. But the complexity lies in the data distribution shift between these past disaster events and the current disaster. A distribution describes observation samples in a given space; here, it is influenced by many factors such as the location, the nature and the strength of the natural hazards.

Neural networks are known to perform well when the training and testing samples are drawn from the same distributions; however, they fail to generalize under important distribution shifts [25]. Therefore, we hypothesize that a model trained with supervision on past disaster event images is not sufficient to guarantee good performance on a new disaster event, given the problem's high variability. We thus suggest an approach where the model first learns generic features from many past disaster events to assimilate current disaster specific features. This technique is known as *transfer learning*.

In this work, we experiment with a transfer learning setup that tries to replicate the emergency context. To do so, samples from the current disaster event must be annotated manually in order to fine-tune the model with supervision. However, data annotation is time-consuming and resources costly, so it is crucial to limit the number of required annotated samples from the event's aftermath. Here, we aim to estimate the minimal required number of annotated samples to fine-tune a model to infer the new disaster damages. Developed in a partnership with the United Nations World Food Program (WFP), this work broadly intends to reduce the turnaround time to respond after a natural disaster.

This paper directly contributes to the use of Deep Learning techniques to support humanitarian activities. We have developed an end-to-end damage assessment workflow based on Deep Learning specifically designed for the natural disaster response. To our knowledge, this is the first work that takes into account both the time and data limitations of the emergency context. In addition, the results of our extensive experiments across multiple disaster events highlight the complexity of the task and expose the diversity of disaster damage outcomes. They demonstrate the necessity to evaluate models on a variety of disaster events in order

to assess generalizability.

Our paper is organized as follows. First, we ground our work by describing the humanitarian and emergency context (Section 3.2). Then, we present the related works (Section 3.3) and the fundamental machine learning concepts (Section 3.4) leveraged in this work. In the sequel, we present the dataset (Section 3.5), our methodology (Section 3.6) and the experimental setup (Section 3.7). Then, we discuss our computational experiments (Section 3.8) and propose a new incident workflow based on our results. Finally, we provide concluding remarks and open the discussion for future works (Section 3.9).

3.2 The Humanitarian Context

In this section, we present the World Food Programme (WFP) [26] as the beneficiary organization of this work. The WFP mission and current post-incident workflow inform the design of our solution.

3.2.1 The World Food Programme

The World Food Programme is the food assistance division of the United Nations (UN). It is the world's leading food-aid humanitarian organization, and it received the prestigious Nobel Peace Prize [27] in 2020 for its effort to combat hunger in conflict zones. Its activities are focused on two main areas: (i) emergency assistance and relief, and (ii) nutrition, education and prevention.

Emergency assistance and relief is the immediate and direct response to extreme scarcity of food. In such circumstances, WFP is responsible to relieve the population by transporting and distributing food supplies. The emergency assistance may refer to sporadic activities, triggered by an unexpected major event or a natural disaster, or a more established long-running effort in conflict zones. In humanitarian organizations responses to natural disasters, time is extremely sensitive. Each incident requires unique considerations, yet decision-makers must assess the situation quickly in order to deploy resources in the most effective way.

This project is done in collaboration with the emergency relief division of WFP, more specifically the Geospatial Information System (GIS) unit. This team is responsible for integrating airborne imagery across the organization to make its processes more efficient. Such imagery includes satellite and unmanned aerial vehicle (UAV) images.

3.2.2 On the Use of Satellite Images

The humanitarian emergency response is complex, and multiple tasks can benefit from using remote sensing images. It allows humanitarian organizations to rapidly retrieve critical information from the ground without the need for human resources on the field. Indeed, involving field workers in life-threatening situations is precarious. Moreover, such ground effort requires a high degree of coordination and relies upon the availability of mobile services. Oftentimes, it leads to partial or incomplete information.

Compared to remote sensing, drone images have higher resolution and can typically be obtained more quickly. Furthermore, drones usually fly below the clouds, as opposed to satellites being above them, which allows them to capture images in almost any environmental condition like cloudy, foggy or smoky air. On the other hand, remote sensing imagery offers more consistency (projection angle, ground resolution) and much higher coverage of the devastated area. Ultimately, those two approaches operate at very different scales: drones are preferred for quick micro assessment whereas satellites are better suited for large-scale assessment.

3.2.3 Damage Assessment

In this work, we study the damage assessment task from satellite imagery. As mentioned before, damage assessment may be based on ground observations. However, satellite images offer a safe, scalable and predictable alternative source of information.

Incident Workflow

Damage assessment should be conducted as rapidly as possible, right after the rise of a natural disaster event. However, the process can only really begin after the reception of the post-disaster satellite images. This critical delay typically varies from hours to many days when the meteorological conditions do not allow image captures. When conditions allow, post-disaster satellite images are quickly shared through the strategic partnerships between earth observation providers and humanitarian organizations. The delay to retrieve pre-disaster images is typically shorter; the data already exists, it only needs to be retrieved from archives and shared. Upon reception of satellite images, the goal is to produce an initial damage assessment report as rapidly as possible. This process includes two main steps: mapping and data analytics (Figure 3.1).

Mapping is the backbone task in a damage assessment process. It consists of locating buildings from satellite imagery and tagging those which are damaged according to a predefined scale. Large devastated areas may be processed to find impaired structures. Maps can

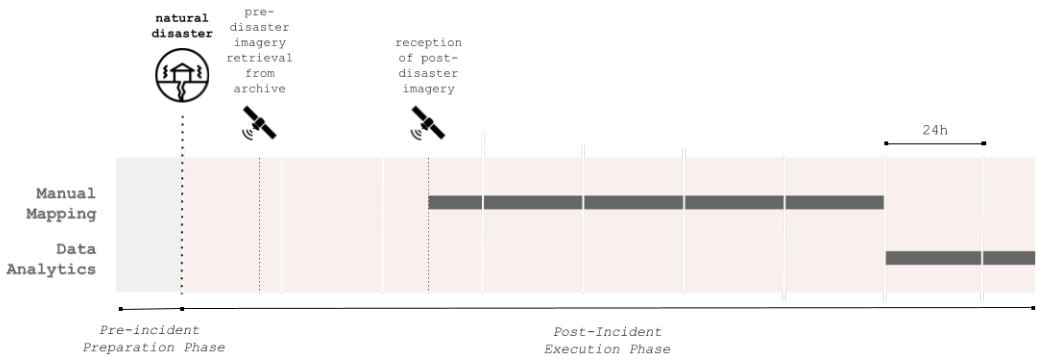

Figure 3.1 Building Damage Assessment Incident Workflow. The post-incident execution phase is triggered by a natural disaster but only initiated upon retrieval of post-disaster satellite images from an imagery archive. Those images are then used to produce maps and analyzed to produce a damage assessment report. The duration of each task is approximate and depends upon many external factors.

then be used as-is to seek precise information by field workers or further analyzed to inform decision-making. They include critical information, such as the density of damaged buildings in a given area.

The data analytics step combines the raw maps of damaged buildings along with other sources of demographic information to inform decision-making. It takes into account the disaster event's specificity to organize an appropriate and dedicated response. For instance, demographic data may indicate if the disaster affected a vulnerable population, in which case the need for food-assistance is even more important.

3.3 Related Works

Satellite images contain highly valuable information about our planet. It can inform and support decision-making about global issues, from climate changes [28] to water sustainability [29], food security [30] and urban planning [31]. Many applications such as fire detection [32], land use monitoring [33], disaster assistance [34] utilize remote sensing imagery.

Data have limited the development of machine learning models for damage assessment since few suitable public datasets exist. The first works were conducted in the form of case studies, i.e. that works targeted one or few disaster events to develop and evaluate machine learning approaches.

In 2016, Cooner and al. [3] took the 2010 Haiti earthquake case to apply machine learning techniques to the detection of damaged buildings in urban areas. Fujita and al. [4] took it

a step further by applying CNN to solve damage assessment using pre- and post-disaster images from the 2011 Japan earthquake. They released the *ABCD* dataset as part of their work. Sublime et al. [5] studied the same disaster by applying change detection techniques. Doshi and Al. [6] leveraged two publicly available datasets for building and road detection: SpaceNet [7] and DeepGlobe [8], to develop a building damage detection model. Their approach relies on the relative changes of pre- and post-disaster building segmentation maps.

Since then, Gupta et al. [1] have released the xBD dataset, a vast collection of satellite images annotated for building damage assessment. It consists of Very High-Resolution (VHR) pre- and post-disaster images from 18 disaster events worldwide, containing a diversity of climate, building, and disaster types. The dataset is annotated with building polygons classified according to a joint damage scale with four ordinal classes: `No damage`, `Minor damage`, `Major damage` and `Destroyed`. A competition was organized along with the dataset release. The challenge's first position went to Durnov [10] who proposed a two-step modelling approach composed of a building detector and a damage classifier.

The release of the xBD dataset sparked further research in the field. Shao et al. [11] investigated the use of pre- and post-disaster images as well as different loss functions to approach the task. Gupta et al. [12] and Weber and al. [13] proposed similar end-to-end per-pixel classification models with multi-temporal fusion. Hao and al. [14] introduced a self-attention mechanism to help the model capture long-range information. Shen et al. [15] studied the sophisticated fusion of pre- and post-disaster feature maps, presenting a cross-directional fusion strategy. Finally, Boin et al. [16] proposed to upsample the challenging classes to mitigate the class imbalance problem of the xBD dataset.

All of these methods share the same training and testing sets, and hence, they can be easily compared. However, we argue that this dataset split does not suit the emergency context well since the train and test distribution is the same. Therefore, it does not show the ability of a model to generalize to an unseen disaster event. In this work, we want to study a model's ability to be trained on previous disaster events to be ready when a new disaster unfolds.

The model's ability to transfer to a future disaster was first studied by Xu et al [9]. That work included a data generation pipeline to quantify the model's ability to generalize to a new disaster event. The study was conducted before the release of xBD, being limited to three disaster events.

Closely aligned with our work, Valentin et al [17] evaluates the applicability of CNNs under operational emergency conditions. Their in-depth study of per disaster performance led them to propose a specialized model for each disaster type. Benson et al. [18] highlighted the unrealistic test setting in which damage assessment models were developed and proposed a

new formulation based on out-of-domain distribution. They experimented with two domain adaptation techniques, multi-domain AdaBN [19] and Stochastic Weight Averaging [20].

To our knowledge, Lee et al. [21] is the first successful damage assessment application of a semi-supervised technique to leverage unlabeled data. They compared fully-supervised approaches with MixMatch [35] and FixMatch [22] semi-supervised techniques. Their study, limited to three disaster events, showed promising results.

Finally, the use of CNNs in the emergency context is also thoroughly discussed by Next et al. [23] who evaluated the transferability and computational time needed to assess damages in an emergency context. This extensive study is conducted on heterogeneous sources of data, including both drone and satellite images.

Finally, in this work, we focus on automatic damage assessment from satellite images, and more specifically, on Very-High-Resolution imagery. Some work has rather investigated the use of drone images [36, 37], multi-sensors satellite images [38], social media images [39] and a mix of multiple data sources [40].

3.4 Machine Learning Fundamentals

In this section, we briefly describe machine learning concepts used throughout this work. Machine Learning is a family of algorithms that learns from data. It seeks for patterns that can best describe a distribution. The result is a function f that can extract information from a dataset D, composed of many data points x of all sorts: feature vectors, images, time series, etc.

3.4.1 Neural Networks

Neural networks are parametric machine learning models composed of many units called artificial neurons. Artificial neurons are made of connection weights, bias and an activation function. Activation functions g are typically non-linear. The output of a single neuron h can be described as

$$h(x) = g(a(x)) = g(b + \sum_i w_i x_i)$$

The weights w_i and bias b are learnable. The capacity of the artificial neuron is such that it can solve linearly separable problems.

Neural networks are composed of multiple artificial neurons assembled in layers. Layers can be stacked to increase the depth of a network, hence the Deep learning appellation. In feedforward neural networks, each neuron from a layer is connected to all neurons from

the subsequent layer. Intermediate layers are known as hidden layers; they are not directly connected to the input. Neural networks can achieve non-linear separation by connecting those hidden layers to the prior layer's output.

Neural networks are parametric models; each layers' weights and biases constitute its set of parameters θ. A neural network f thus infer the output $\hat{y}$ from the input x, given the set of parameters θ.

$$\hat{y} = f(x; \theta)$$

3.4.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are neural networks specifically adapted for problems that involve inputs with a known topology, e.g. time series (1D topology), images (2D topology). They can deal with high dimensional inputs and exploit the intrinsic topology of input data.

Convolutions

As their name indicates, CNNs are based on a mathematical operation called convolution. The convolution implies two matrices: the first matrix is called the input, typically an image, and the second is called the kernel. The output is sometimes referred to as a feature map. Figure 3.2 shows an example of a convolution operation.

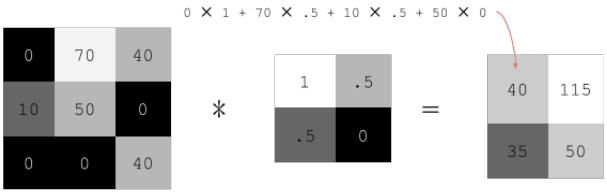

Figure 3.2 Example of convolution. The first 3x3 matrix is the input, then the 2x2 matrix is the kernel, and the last 2x2 matrix is the result. Here, the kernel is flipped for a more straightforward representation.

The convolution of an image x at position ij with a kernel k is defined as follows, where p and q are the kernel size.

$$(x * k)_{ij} = \sum_{p} \sum_{q} w_{i+p,j+q} \times k_{r-p,r-q}$$

Convolutions are commonly used for image processing with predefined kernel values. In

convolutional neural networks, kernel values are learned. In practice, discrete convolutions are computed with constrained, sparse matrix multiplication, known as doubly block circulant matrix.

CNNs are generally constituted of a stack of convolutional layers, followed by activation functions and pooling layers. Pooling layers downsample or upsample input image or feature maps; statistics such as the minimum, maximum or average value over input neighbours define the output values. Pooling layers also help small translation invariance making possible the use of input with varying sizes.

Data Augmentation

Translation invariance is built into the convolution and pooling layer themselves. However, CNNs are not robust to other types of transformations: illumination, scale, rotation, elastic distortion and more. Hence, artificially augmented data is incorporated into the training dataset such that the networks learn to be invariant to such transformations. Figure 3.3 shows an example of artificially augmented data.

Figure 3.3 Artificially augmented image. Left: original image. The rest are examples of images augmented through combinations of **cropping**, **rotation** and **color jittering**.

3.4.3 Siamese Networks

Siamese neural networks are specialized neural networks that can handle and compare two distinct inputs (Figure 3.4). Typically, a Siamese network is composed of two sequential models: an *encoder* G that extracts features, and an optional features similarity function H, each defined by their own set of parameters θ_g and θ_h. The core idea is that the same encoder is applied on both inputs. Such a Siamese Network f, with inputs x_A and x_B, can generally be defined as

$$f(x_A, x_B; \theta) = H(G(x_A; \theta_G), G(x_B; \theta_G); \theta_H).$$

3.4.4 Transfer Learning

Transfer learning is the idea of sharing knowledge from a model trained on a source distribution to an untrained model to be trained on a similar or a different distribution. It leverages learned features and weights from a pre-trained model to warm-up another model's training.

The motivations for transfer learning are simple. First, training deep neural networks with millions of parameters requires large-scale datasets. However, such datasets are rare and difficult to gather. Second, training large models from scratch may take up to weeks to converge. In fact, state-of-the-art CNNs, and more generally neural networks, are rarely trained from scratch [41, 42].

There are many ways to perform transfer learning. In this work, we leverage pre-trained models and standard fine-tuning. Models are generally pre-trained on large-scale datasets. ImageNet is one of the most commonly used for vision tasks [43] . It consists of a collection of over 14 million images annotated for classification. Pre-training a model on ImageNet effectively means that a model A is trained to classify images.

To do so, it must learn to extract relevant visual features, both low- and high-level, such as edge detectors or building detectors. Ultimately, most of these features, especially the low-level, can help solve almost any computer vision task. This knowledge of feature extraction is encoded by the model's weights. Therefore, to leverage this knowledge, another model B may initialize its weights with the weights values of the pre-trained model. The process of further training model B on a smaller amount of data is called **fine-tuning**; the weights are simply adjusted or fine-tuned to the new domain and task.

3.4.5 Attention Mechanism

Attention is a concept that has been introduced for Deep Learning model to focus on different parts of the input. To infer predictions, not every pixel in an image has the same importance:

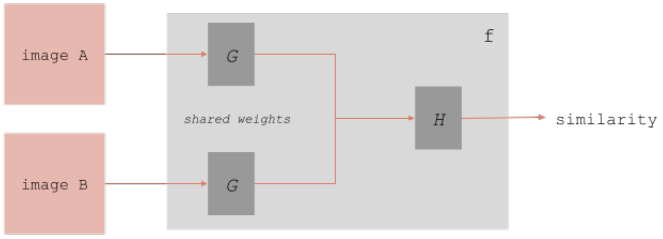

Figure 3.4 Siamese Network. The model first extracts features from images through an encoder G and then compares the features through H.

the attention mechanism allows the model to pay visual attention to certain key parts of the image without discarding the least important parts. In Deep Learning, attention can usually be interpreted as a vector of importance weights. The dot-product is one of many attention mechanisms [44].

3.5 Dataset

To train neural networks, large-scale and preferably annotated datasets are required. While every day remote sensors are capturing a visual snapshot of our planet from above, the annotation of those images remains rare.

This work relies on the xBD dataset [1], a collection of satellite images annotated for building damage assessment. To-date, xBD is the largest dataset for building damage assessment. It consists of Very-High-Resolution (VHR) pre- and post-disaster image pairs from 18 different disaster events worldwide. Images come along with building location and damage level tags. Overall, the dataset contains more than 800k annotated buildings.

3.5.1 Annotation

The xBD dataset is annotated for building damage assessment. Therefore each image pair is accompanied by building polygons corresponding to building locations along with damage assessment scores. These scores correspond to a joint damage scale with four ordinal classes: `No damage`, `Minor damage`, `Major damage`, and `Destroyed`. Each of these classes correspond to different damage features depending on the nature of the disaster. For instance, a partial roof collapse and water surrounding a building would be classified as `Major Damage` (see Table 3.1).

In this work, we consider a simplified binary classification problem, grouping `No damage` and `Minor damage` into one category, and `Major damage` and `Destroyed` into another. We assume that damages classified as `Minor damage` do not require immediate emergency attention from humanitarian organizations and can therefore be ignored from the damage assessment. Ignoring `Minor damage` reduces the task's complexity since, by definition, it is generally more subtle and consequently harder to predict.

The distribution of damage varies across disaster events (Figure 3.5), but it favours undamaged buildings for all disaster events. Over the whole dataset, there is a 5:1 ratio of `No Damage` versus `Damage` buildings. This data imbalance should be taken into account in the design of the optimization loss and the evaluation metric (see sections 3.6.3 and 3.6.3).

Table 3.1 Description of damage assessment scores. Our work is based on a simplified binary classification scheme. The original scheme is presented in [1].

xBD Original Class	Simplified Class	Description
0 (`No damage`)	0 (`No damage`)	Undisturbed. No signs of water, structural or shingle damage, or burn marks.
1 (`Minor Damage`)	0 (`No damage`)	Building partially burnt, water surrounding structure, volcanic flow nearby, roof element missing, or visible crack.
2 (`Major Damage`)	1 (`Damage`)	Partial wall or roof collapse, encroaching volcanic flow, or surrounded by water/mud.
3 (`Destroyed`)	1 (`Damage`)	Scorched, completely collapsed, partially / completely covered with water/mud, or otherwise no longer present.

3.5.2 Images

The database contains image tiles of 512×512 pixels, and the resolution is at most 0.3m per pixel. Each sample consists of spatially aligned image pairs: a first snapshot is taken at any time before a natural disaster occurred in a given location, and a second co-located image is taken after the incident.

The coupling of pre- and post-disaster images reveals essential information to assess the damage. Although the post-disaster image alone might might suffice in some cases, one can better evaluate damage knowing the building's original state and its surrounding. Figure 3.6 shows a counterexample where the post-disaster image alone is insufficient for a confident damage assessment. The contrast between pre- and post-disaster image features helps distinguish the presence of damage and thus contributes to a more confident evaluation. This contrast is even more critical for detecting peripheral damages, as opposed to structural, and more specifically the less severe ones.

Each image pair covers roughly the same $150m \times 150m$ ground area, which is larger than a regular building. The image provides a larger context to make a correct damage assessment. Figure 3.6 shows how floods, for instance, are hard to perceive given only the building and local context. Humans, too, reflect and evaluate potential damage to a building by seeking for visual cues in the surrounding area.

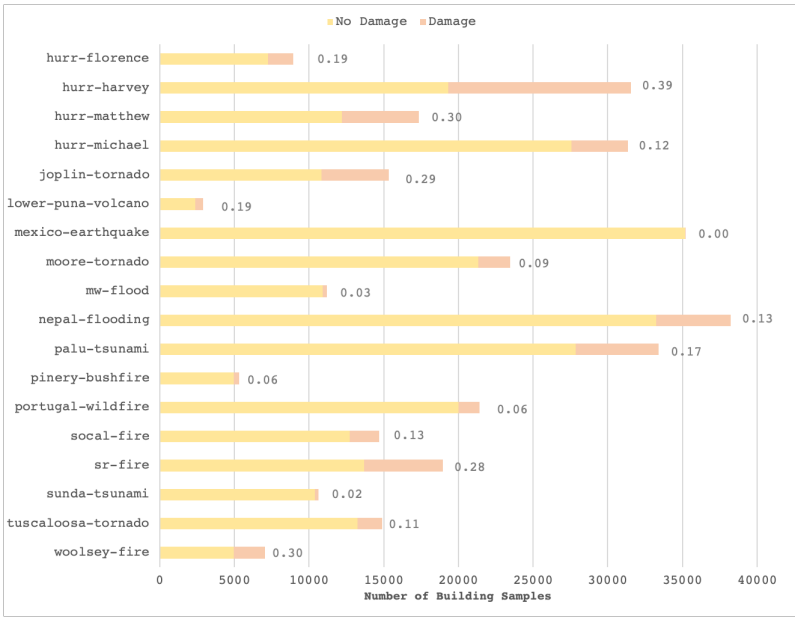

Figure 3.5 Per disaster empirical distribution of building damage. The numbers are ratios of `Damage` buildings per disaster.

Figure 3.6 Image pair before and after Hurricane Florence. The bounding box focuses on a single building. The area surrounding the building is flooded.

By nature, global remote sensing problems are often high-dimensional: they must include images from around the globe to capture the inherent geodiversity. For building damage assessment, the disaster types and the time dimension contribute to further complexity. Time contributes to complexity in view of the fact that any dynamic information must be captured.

Location

The xBD dataset includes events from 18 different locations throughout the world. It covers both rural and urban regions (respectively, sparse and dense in buildings). Each site is unique, in its climate and demographic characteristics: Climate determines the presence of grass, sand, snow, etc. Demographics influence the infrastructure, such as roads, buildings, etc. Buildings vary in shape, size, materials and density of arrangement. For example, a low density of buildings is commonly found in rural areas, wealthier neighbourhoods tend to have bigger houses, and Nordic countries require resistant construction materials, etc.

The distribution of samples across locations is not uniform either: the number of samples and buildings per site is not consistent. Moreover, although including worldwide images, the xBD dataset remains biased in favour of American locations. The dataset also does not fully capture the diversity in climate conditions: snow and ice climates, among others, do not appear in the dataset.

Table 3.2 serves as the abbreviations index to the disasters and locations used throughout this work.

Disaster and Damage Type

The dataset also contains numerous disaster types, leading to different damage types, depending on the event's location. Disaster types include: hurricane, earthquake, tornado, tsunami, wildfire, volcano eruption and flooding.

Depending on the destructive force (wind, water, fire, etc.) and the location, different types of damage are visible from the satellite imagery: collapsed roofs, flooding, burned buildings, etc. Damages can be described by their severity and can be divided into two groups: *peripheral* and *structural*. Structural damages are on the building structure itself (e.g. collapsed roof), peripheral damages on its periphery (e.g. flooded area), for examples see Figure 3.7. There is a reasonably uniform distribution of those two types of damage across the dataset. However, each disaster type is typically the cause of either peripheral or structural damages, but rarely both. Ultimately, regardless of the disaster type, buildings are classified under the binary

Table 3.2 Disaster event, abbreviation and location represented in the xBD dataset.

Disaster event	Abbreviation	Country
Hurricane Florence	hurr-florence	USA
Hurricane Harvey	hurr-harvey	USA
Hurricane Matthew	hurr-matthew	Haiti
Hurricane Michael	hurr-michael	USA
Joplin Tornado	joplin-tornado	USA
Lower Puna Volcano	lower-puna-volcano	USA (Hawai)
Mexico Earthquake	mexico-earthquake	Mexico
Moore Tornado	moore-tornado	USA
Midwest Flood	mw-flood	USA
Nepal Flooding	nepal-flooding	Nepal
Palu Tsunami	palu-tsunami	Indonesia
Pinery Bushfire	pinery-bushfire	Australia
Portugal Wildfire	portugal-wildfire	Portugal
Socal Fire	socal-fire	USA
Santa Rosa Fire	sr-fire	USA
Sunda Tsunami	sunda-tsunami	Indonesia
Tuscaloosa Tornado	tuscaloosa-tornado	USA
Woolsey Fire	woolsey-fire	USA

schema of `damage` vs. `no damage`.

Time and Seasons

The temporal dimension tracks anything that differs between the pre- and post-disaster images, including damage-related changes. Changes can be due to moving objects, such as cars, new infrastructure, and seasonal changes, such as vegetation colour. The temporal dimension can be used to compare pre- and post-disaster images.

Although temporal information can be rich and informative, it adds further complexity to the modelling. Assessing damage based on the peripheral information is more challenging because

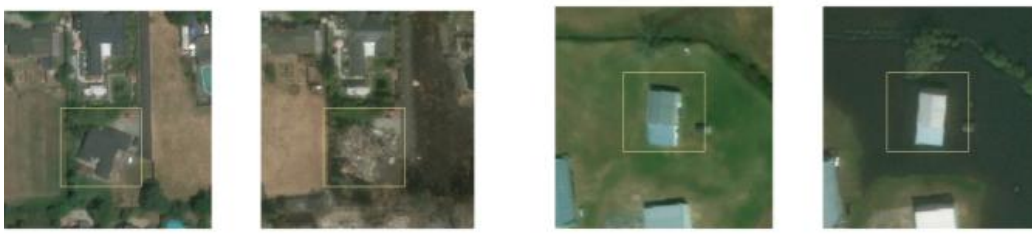

Figure 3.7 Damage types: Structural (left) and peripheral (right).

the model must learn to discriminate based on damage information and ignore seasonal changes. For instance, it should detect the presence of water in a flooded region, while ignoring change in vegetation colour due to seasonal change.

Temporal changes are usually effortless to identify: humans and machines are good at filtering through noise. However, for a model to be able to differentiate seasonal changes from damage changes, it must understand the semantics of the changes in remote sensing imagery. Going so is therefore much more complex and requires a high diversity in seasonal changes and damage types. Irrelevant differences, such as sun exposure, can influence the model predictions. That said, the dataset contains only 18 instances with seasonal changes (one per location), which arguably does not cover enough temporal diversity for the model to generalize.

Other Factors of Complexity

The variation in projection angles (aka. off-nadir angles) typically seen in satellite imagery is not captured in the dataset: all xBD samples are taken with a nadir angles.

Additionally, some limitations are not explicitly addressed in our work, neither included in the dataset: occlusions (cloud, tree canopies, etc.), damages invisible from above (broken windows, damaged wall, etc.) and noisy labels (unintended annotation errors due to fatigue, misinterpreted images, etc.)

3.6 Methodology

3.6.1 Method requirements

Our method predicts building damage maps from satellite images in the aftermath of a natural disaster. It aims to provide a machine learning workflow to reduce assessment delays and support faster decision-making. The method requirements can be broken down to three main topics: model readiness and post-incident execution time, performance, and interpretability.

Model Readiness and Post-Incident Execution Time

For a mapping algorithm to be successfully applied in an emergency context, its post-incident execution time must be short. The **post-incident execution** phase includes any task that will be executed after the disaster, thereby influencing the response delay.

The **model readiness** refers to any ML model development tasks done in the pre-incident preparation phase to shorten the post-incident execution time. A ML model development cycle typically includes data gathering and annotation, as well as the model training, evaluation

and inference phases. To perform these steps in the pre-incident phase, they must be independent of the current disaster data. That is because, annotation is excessively time costly and should be done in the preparation phase as much as possible. Thus, the model should require as few annotated samples as possible from the current disaster event. The algorithm should leverage past events images and annotations to generalize to future disasters.

Similarly, deep learning model training may take up to many days. That said, whenever possible the model should be pre-trained on past disaster event samples as part of the pre-incident preparation phase for it to be ready to infer building locations and damage levels in a post-incident phase. Overall, the model post-incident execution time must be shorter than that of manual annotation for it to be of operational value.

Model Performance

Under distribution shifts, machine learning models tend to under-perform. The training of models on past disaster events (to reduce the post-incident execution time) can be hindered by a gap between the distributions of the train set and the test set: there is a trade-off between model performance and execution time.

The model prediction should provide an overall picture of the situation to decision-makers. Thus, a building-level granularity may not be required for the initial assessment. For instance, if the model predicts nine buildings out of ten correctly, the ensuing decision to set up a food-distribution center is likely to remain the same. Therefore under emergency constraints execution time might be favoured instead of performance. Incorrect information might gradually be corrected manually or based on ground observations to refine the mapping and support more low-level decisions eventually.

Interpretability

Damage maps derived from remote sensing are intended to be used to inform decision-making. Therefore, the output should be understandable and interpretable. The output of a deep learning model, like for classification or semantic segmentation, can be interpreted as a conditional probability at the pixel level. Hence, depending on the situation and the risk level, data analysts may decide to accept a lower or higher level of confidence in the prediction to adjust the output. Generally speaking, lowering the confidence level threshold is likely to yield higher precision, but lower recall.

3.6.2 Approach

The building damage assessment task can be decomposed into two assignments: first, locating the buildings and, second, assessing their integrity. Therefore, we propose an intuitive two-steps model design composed of a building localizer (*BuildingNet*), followed by a damage classifier (*DamageNet*), as shown in Figure 3.8.

First, *BuildingNet* is a binary semantic segmentation model, i.e. every pixel is assigned one of two classes: `building` or `background`.

Image patches are then cropped around each detected building and passed on to the damage classification model. *DamageNet* is a binary classification model whose output is either `Damage` or `No damage`.

While designing a model that can solve both tasks end-to-end is feasible, we argue that a two-step model is more suitable in an emergency context. First, both models can be trained, evaluated and deployed separately, thus each model is computationally cheaper compared to the end-to-end approach. The decoupling may eventually reduce the post-incident execution time. Moreover, concurrently optimizing one model for building location and damage classification is demanding in terms of GPU computational resources, and a two-model approach is likely to converge faster. End-to-end learning is known to have scaling limitations and inefficiencies [45].

Another argument for a two-step approach, is that the building detection task on its own only requires pre-disaster imagery and building location annotation. In a decoupled model design, the organization can proceed to building detection as soon as the pre-disaster imagery is made available. Only the damage classification task is awaiting post-disaster imagery to start. Objectively, building detection is also a much simpler task than damage classification because it does not suffer from complexity of the temporal dimension.

Finally, both model outputs are probabilistic, representing the probability of belonging to a given class. Decoupling them allows for more interpretability and flexibility as both the location and the damage sigmoid output can be threshold separately.

3.6.3 Model Architectures

The building localization is solved as a binary semantic segmentation problem using the Attention-U-Net architecture [46] with a binary cross-entropy loss (Figure 3.9). The model's input is a 512×512 pre-disaster image, and the output is a binary segmentation map.

Attention-U-Net is an extension of U-Net architectures [47] with attention gates that allows

Figure 3.8 Two-step modelling approach composed of (1) a Building Detection model (*BuildingNet*) and a (2) Damage Classification model (*DamageNet*). The input of *BuildingNet* is a pre-disaster image, the output a binary segmentation heatmap, i.e. that each pixel has a sigmoid ouput. The input of *DamageNet* is both the pre- and post-disaster image patches centred on a single building along with the building mask. The two models are applied sequentially.

the model to focus on structures of different sizes. It was originally implemented for medical imaging, but has been commonly used in many other fields due to its efficiency and relatively low computational cost.

The damage assessment model is a Siamese ResNet [48, 49] classifier (Figure 3.10). ResNet is a state-of-the-art classification architecture. The architecture performance relies on its skip connections that allow the gradient to back-propagate more easily as the model's depth increases. We experiment with ResNet architectures of different capacities: with 18, 34, 50 and 101 layers. The model input is a 224×224 patch of the aligned pre- and post-disaster images centred on the building to classify. The patch size is set so as to limit the memory usage while keeping sufficient contextual information. The first ResNet layers are computed in separate streams with shared weights for the pre- and post-disaster inputs. Then, the feature maps are concatenated, and the last convolutional layer blocks are applied.

The binary segmentation heatmap is multiplied with the 64-channels feature maps before the first downsampling layer. The mask is applied like an attention mechanism, such that the model focuses on the building but retains information on the whole image context. This mechanism is essential to make accurate predictions on certain types of damage, like floods and volcanic eruptions, where there is no visible damage to the building structure itself, but only on its surrounding. The attention mechanism combines a convolution layer and a matrix multiplication that allows the model to up-weight only the most relevant of features.

The damage classification model is optimized using binary cross-entropy with a weight of five for positive samples, set according to the ratio of positive and negative samples in the overall dataset. The output is bounded between zero and one using a sigmoid activation function.

Training Strategy

To minimize the post-incident execution time, the training strategy consists of training both models prior to the disaster to be ready for inference. That said, both the building detection and damage classification models do not have access to data from the current disaster event.

We hypothesize that the building detection model can generalize well to the current disaster, given the simplicity of the task. However, the damage classification model is less likely to generalize to the current disaster event given the complexity of the task. We believe that the xBD dataset is not diverse enough in terms of location, seasonal changes and disaster type for the model to learn features that transfers well to unseen disasters.

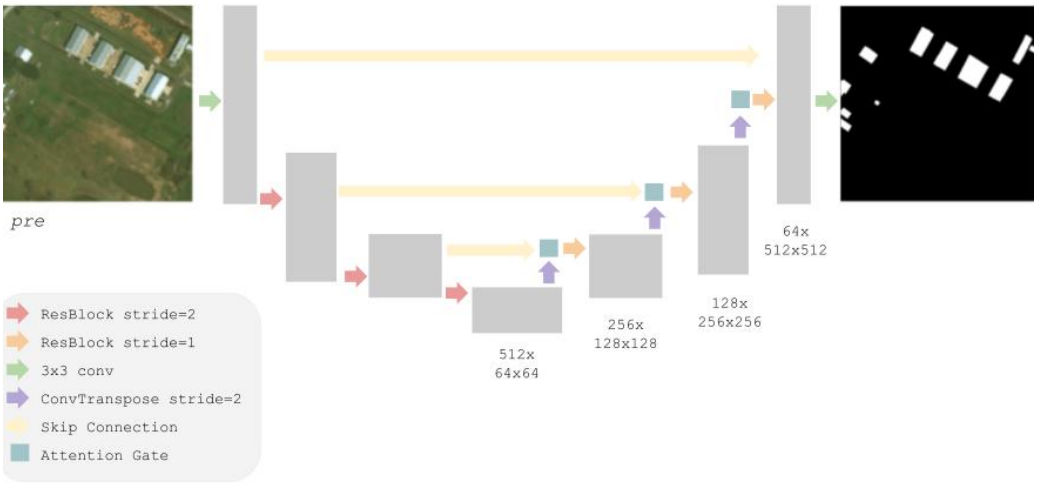

Figure 3.9 *BuildingNet* follows an Attention-U-Net architecture. The pre-disaster image is downsampled and then upsampled (i.e. a bottleneck architecture) at different spatial scales. The skip connections allow an encoding at a certain scale to skip through further downscaling and to merge with the upsampling stream after being filtered through an *attention* gate. The *attention* gate learns to focus on different structures.

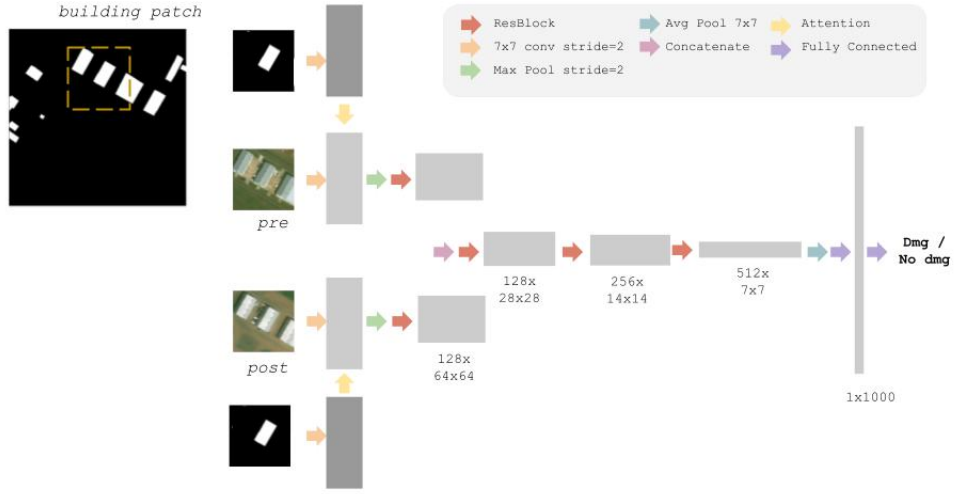

Figure 3.10 DamageNet follows Siamese-ResNet architecture. Both pre- and post-disaster feature streams are eventually concatenated into one damage classification stream. The building mask is applied as an attention mechanism. This figure shows the feature map shape for ResNet34.

Evaluation

Both the building detection and the damage classification problems are imbalanced in favour of the negative class: building detection is imbalanced in favour of the background pixels, while damage classification favours undamaged buildings. Hence, as opposed to the accuracy, the F_1 metric is used for its ability to describe the performance of the model to predict both the majority and the minority class reasonably. The F_1 is the harmonic mean of recall and precision. For both tasks, the F_1 score over the minority class is measured, i.e. building pixels for the building detection model, and damaged buildings for the damage classification.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{TP}{TP + 0.5FN + 0.5FP}$$

where TP represents the true positives, FN the false negatives, and FP the false positives.

To be aligned with the training strategy, the goal is to measure the model's ability to generalize to the current disaster event or predict damages accurately for a disaster event that the model has not seen during training. Therefore, we use all samples from a given disaster event to create the test set, and the remaining samples all other disasters form the train set.

As a result, train/test split uses 17 events for training and 1 event for testing. We create 18 different train/test splits, one for each event, to evaluate the model's performance on unseen events. For example, detecting damage in areas devastated by a wildfire does not guarantee success in assessing damage in imagery from a flood; thus, this ablation experiment is performed for each disaster event to assess the methods' generalizability under different circumstances.

3.7 Experimental setting

Experiments are run to evaluate if building localization and damage detection models trained on past disaster events can generalize.

First, we conducted an ablation study to determine the best architecture for the damage classification task. The model capacity and the position fusion of the pre-incident and post-incident streams were analyzed. The study covered ResNet architectures with an increasing capacity: `resnet-18`, `-34`, `-50` and `-101`, which refers to the total number of layers. The ResNet architecture consists of four blocks of convolutional layers that eventually output tensors with decreased shape in the spatial dimensions but with more feature maps (aka. channels). Our ablation study includes architectures where the streams (pre- and post-disasters) are fused after the first, second, third and fourth convolutional block (Figure 3.11).

The study was run on the Hurricane Florence dataset split for its reasonably challenging complexity. There were three runs per architecture to assess the training stability.

To assess the usability of both the building detection and the damage classification models in an emergency context, building localzation and damage detection models are separately trained and evaluated on each of the 18 disaster splits individually. That said, a single run consists of training a model on all 17 disaster events and testing on the remaining samples of a single target disaster event. To assess each model's training stability, the experiment was repeated three times with different random seeds for each target disaster event. The performance was measured with the F_1 score.

3.7.1 Training Hyperparameters

BuildingNet is trained with the Adam optimizer, with a learning rate of 0.001 and a batch size of 16. We use an early stopping policy with 10 epochs of patience, and learning rate scheduling with decay 0.5 and patience 5. We apply basic data augmentations during training: random flips, crops, colour jitter.

DamageNet weights are pre-trained on ImageNet, and we apply basic data augmentation during training. It is trained with the Adam optimizer, with learning rate 5e$-$5, batch size 32, and weight decay 0.01. We use an early stopping policy with 15 epochs of patience, and learning rate scheduling with decay 0.5 and patience 2. The final fully connected classification layer includes dropout with a probability of 0.5 for an element to be zeroed out.

For both models, *BuildingNet* and *DamageNet*, a random search determines the best hyperparameters. Hyperparameter tuning is done once using a shuffled dataset split with samples from all disasters in both the train and the test sets. All 18 disaster events are present in both the train and the test set, but with no overlap. The test set, therefore, includes representations of all disaster events. Although this method might not yield the optimal solution when applied to the individual disaster splits, this method seemed like a fair trade-off between performance and resource usage.

3.8 Results and Discussion

In this section we go over *BuildingNet* and *DamageNet* performance results individually, and then analyze the resulting incident workflow, from pre-incident preparedness to post-incident execution.

Figure 3.11 Ablation study configurations for the fusion of the pre- and post-disaster streams after the first (1), second (2), third (3) and fourth (4) blocks.

3.8.1 BuildingNet

Figure 3.12 shows the performance of the model to predict building location for each disaster event. The bar shows the average performance over the three runs, and the error bars the standard deviation. The F_1 score is measured per pixel with a threshold of 0.5 over the sigmoid output. The average score across all disaster events is 0.808 – shown with the dotted gray line. As shown by the error bars, the training of *BuildingNet* converges to stable solutions across the different disaster events, with nepal-flooding having the highest standard deviation (0.023).

The building detection model performs well on average and across disasters. Figures 3.19 and 3.20 shows the model predictions and their corresponding F_1 score.

The building detection task is independent of the disaster event since they can be identified from the pre-disaster imagery. Compared to damage classification, building detection is a relatively simple assignment: there is no temporal dimension involved. It is possible to identify buildings worldwide with different shapes and sizes. Climate also varies across locations. However, a building detection model quickly learns to ignore background pixels (snow, sand, grass, etc.) to focus on objects and structures. There are few objects or structures visible from satellite images. Roads, bridges, buildings, cars and pools are the most common human-built structures, and a well-suited model can learn to extract features to discriminate between them.

Figure 3.12 indeed shows that the performance is reasonably uniform across all disasters. This suggests that it is possible to train a generic building detector to have it ready and prepared to make predictions when a new disaster occurs. The distribution shift is not

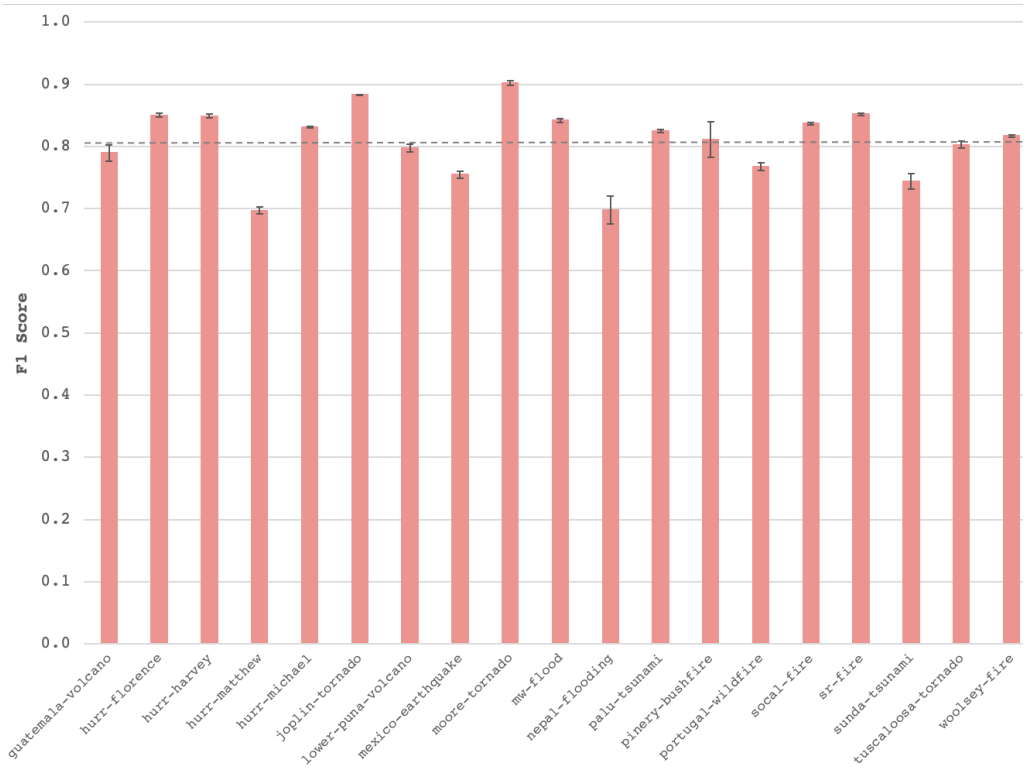

Figure 3.12 *BuildingNet* F_1 score per disaster event.

significant between the training set and pre-disaster images from the area of interest of the last disaster. No annotation, fine-tuning or adjustment is thus necessary to make predictions at test time.

By qualitatively assessing the model's performance on the examples in Figures 3.19 and 3.20, it is clear that the delineation of the buildings is not perfect. However, even with imprecise edges, buildings were detected, hence their damage can be later assessed. Besides, building detection errors do not directly impact decision-making. Detecting edges becomes especially problematic when the building view is obstructed by tree canopies or clouds, for instance.

Nonetheless, entirely missing buildings can cause significant issues, as the damage classification model would ignore the building. However, in practice, data analysts do not look at precise numbers of damaged buildings; they are mostly interested in finding the hot spots or the most affected regions. Damaged buildings tend to be located within the same neighbourhood, and therefore, skipping one building out of many is a tolerable error, as long as the recall does not influence the subsequent decisions. As per our visual observations of predicted buildings, we find that an F_1 score of 0.7 indicates that a fair number of building is detected, but that boundaries are not refined enough. The model stands above this threshold for almost all disaster events.

The five lowest performances are for hurr-matthew (Haiti), mexico-earthquake (Mexico), nepal-flooding (Nepal), portugal-wildfire (Portugal) and sunda-tsunami (Indonesia), for which the performance is below average. Lower performance is typically a result of distribution shifts. Those five disasters have common attributes. First, buildings tend to be smaller than average and, therefore, might be harder to detect. Their boundaries also tend to be blurrier, either because of the building density, or the heterogeneous rooftop materials. These characteristics are specific to the location and the demographic of the region.

Besides, none of these five disasters occurred in the USA. As mentioned in the Methodology section, the xBD dataset contains mostly USA-based disaster events - an imbalance that biases the model against non-US locations. Unsurprisingly, the top five scores are for disaster events that happened in the USA: moore-tornado, joplin-tornado, sr-fire, hurr-florence, and hurr-harvey. It is essential to identify and mitigate these biases in such sensitive humanitarian applications. This is even more true when the model discriminates against more vulnerable populations, which have higher risk to live food insecurity.

Having a building detector ready when a disaster arises simplifies the post-incident workflow. *BuildingNet* is pre-trained in the pre-incident phase and makes predictions based on pre-disaster imagery. Hence, the inference can almost immediately start to predict the buildings' locations. Upon the reception of post-disaster imagery, buildings' areas are already known.

3.8.2 Damage Classification

Figure 3.13 shows an ablation study over model capacity and streams fusion to determine the best architecture for *DamageNet*. Every data point represents the average performance over the three runs for each architecture, whereas the error bars represent the standard deviation. ResNet34 with the fusion of both streams after the first convolutional block performs the best with good training stability. We use this architecture for all further experiments.

Figure 3.14 shows the performance of the model to predict building damage for each disaster event. The bar shows the average performance over the three runs, and the error bars the standard deviation. The average score across all disaster events is 0.590, which is represented by the dotted gray line. As shown by the error bars, the training of *DamageNet* converges to stable solutions across the different dataset events, with the highest standard deviation across the three runs being 0.048 for mw-flood.

Damage classification is a much more complicated task for two main reasons. First, the task involves a temporal dimension that is too diverse and hard to capture with the current sample size. Beyond that, the model must not only learn to ignore some of the temporal changes when they relate to season, but also discriminate over other temporal changes when they relate to damage. This is particularly complex for the model with no prior knowledge of the geographical region and the expected climate or disaster type as well as the expected damage. Damages may have very diverse definitions and representations, depending on the disaster type and the pre-incident environment. Hence, seizing the temporal changes and the variety of damages requires a larger sample size than that of the xDB dataset.

As expected, the xBD dataset does not seem to encompass enough diversity to train a generic damage classification model (Figure 3.14). The pre-trained model results suggest that some disaster event test samples are out-of-distribution with respect to the training set; the model does not understand what damages look like in the current test disaster context. The performance across disasters is indeed far from uniform.

Disasters where the model performs the worst (hurr-harvey, sunda-tsunami, hurr-michael) are more challenging. First, these disaster events result mainly in peripheral damages, i.e. they are visible on the building's surroundings, which may be easily confused with seasonal changes. Moreover, hurr-michael damages are very suddle and human annotation might be noisy. Similarly, hurr-harvey and sunda-tsunami buildings are partially or entirely obstructed either by trees or clouds, making the assignment more difficult. Note that mexico-earthquake results are not considered since there are too few positive samples (22 damaged buildings against 35164 negatives) for the score to be significant.

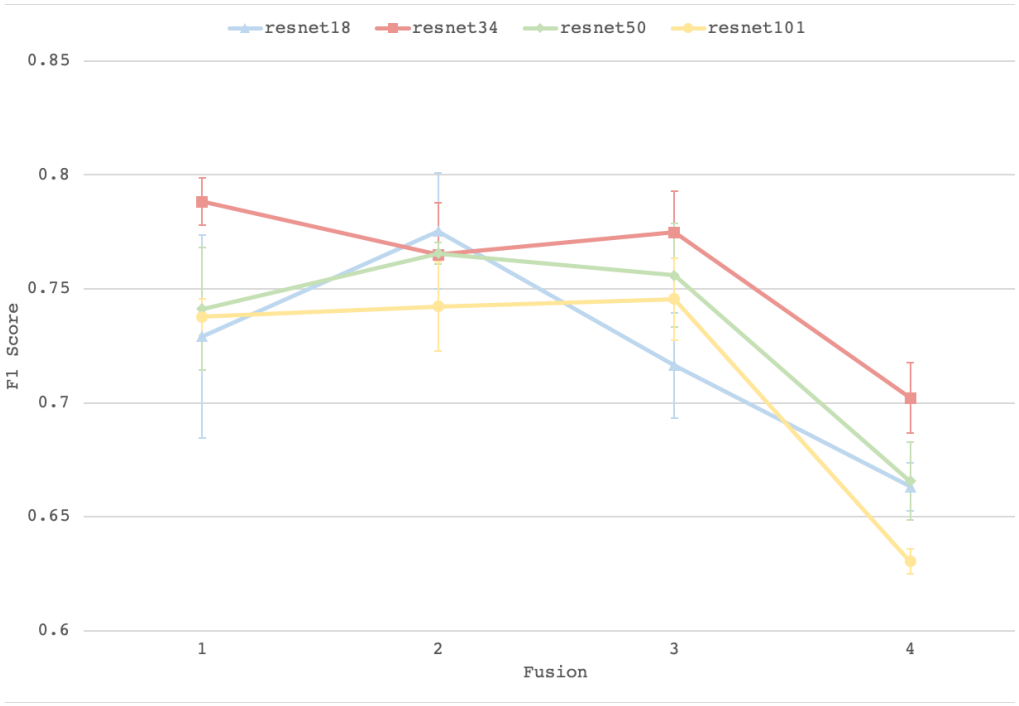

Figure 3.13 Ablation study results for the fusion of the pre- and post-disaster streams after the first (1), second (2), third (3) and fourth (4) blocks. Each line represents ResNet with a different capacity.

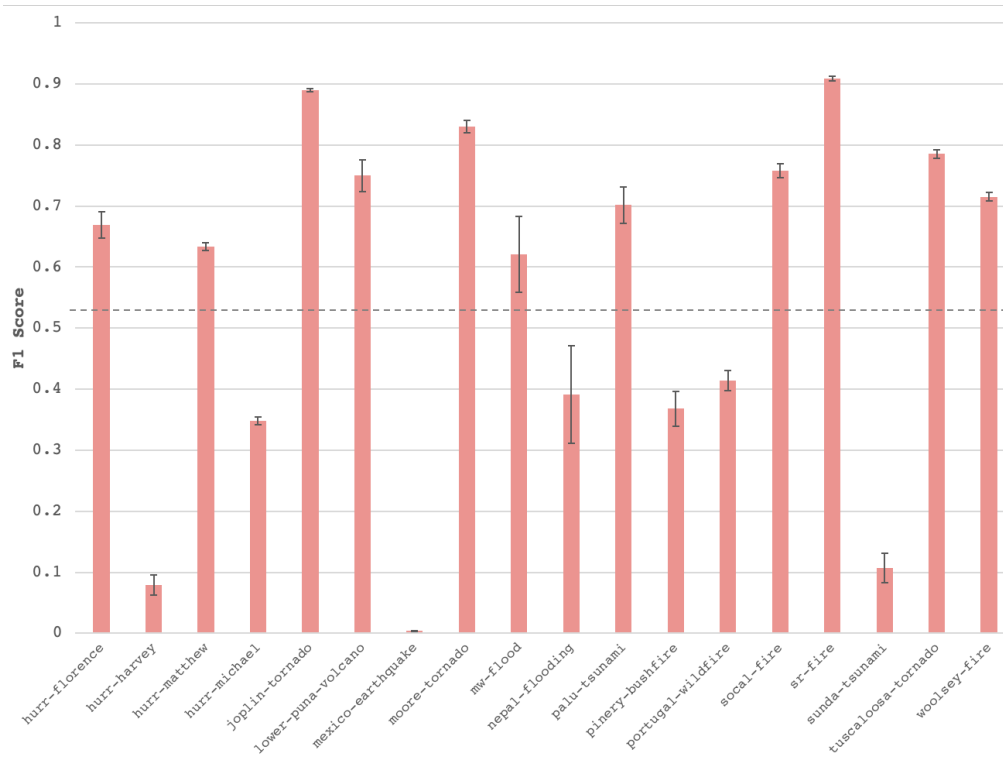

Figure 3.14 F1 score of *DamageNet* per disaster event.

Conversely, disaster events for which *DamageNet* performs well are defined by heavy, structural damages. sr-fire, joplin-tornado, moore-tornado and socal-fire leave buildings either intact or destroyed, and can be easily classified.

These unsatisfactory results suggest that the model should be fine-tuned to learn features from the current disaster event. Accordingly, these results invalidate the proposed training strategy on past disaster samples and need further tuning to predict the current disaster's damage.

Further Experiments

Since pre-training *DamageNet* on past disaster event samples is not sufficient for the model to generalize to the current disaster, we established a strategy to fine-tune the model weights but still limit the post-incident execution time. The goal is to readjust *DamageNet* weight with the current disaster event images. We propose two standard transfer learning and fine-tuning methods. The later relies on the human annotation of the current disaster event (Figure 3.15). Because it depends on post-disaster satellite imagery reception, annotation ought to be performed in the post-incident execution phase.

As illustrated in Figure 3.15, *DamageNet* is first pre-trained on all past disaster events. Then, upon reception of recent satellite images, a minimal number of building samples are annotated for damage classification. Finally, *DamageNet* is trained again on the current disaster samples to adjust the model's weights on the current disaster features.

Nonetheless, annotation is highly time-consuming and the annotation of current disaster samples necessarily takes place after the event. To be consistent with the objective of minimizing the post-execution incident phase, fine-tuning a model should require as few training samples as possible. Therefore, to reduce the annotation effort to its bare minimum, we estimated the number of annotated building samples required to train a model for damage classification.

To do so, for each disaster event, we extracted 10k building samples from the test set to fine-tune *DamageNet* using an increasing number of annotated current disaster samples from those withdrawn samples. We use 80% of the samples for training, 20% for validation and evaluate the performance on the test set remaining samples. Again, the experiments were conducted for all 18 disaster events. We executed three trials with different random seeds for each combination of target disaster and number of training samples. The test set remains the same for each target disaster. We applied the same set of data augmentation (as described in earlier sections) during fine-tuning.

Figure 3.16 shows the increasing performance of *DamageNet* for each disaster with a growing number of annotated samples. These results suggest that given enough annotated samples from the current disaster event, *DamageNet* can predict damaged buildings: the model's performance increases with the number of annotated samples until it reaches a plateau.

The distribution of damage classes per disaster confounds the comparison of the minimum number of annotated samples required. Fine-tuning indeed requires both positive and negative samples (or damaged and undamaged buildings). For instance, mw-flood and sunda-tsunami contain fewer damaged buildings in proportion compared to the average (see Figure 3.5), explaining the fine-tuning approach's instability for these events. For that same reason, training is also fairly unstable with less than 100 samples.

The supervised fine-tuning method did not seem to hurt the performance for any of the disasters, and for most of them, there is no significant gain past 1500 annotated samples. On average, the disaster represented in the xBD dataset covers roughly 19000 potentially damaged buildings. Based on visual assessment and after consulting with our domain experts from WFP, we consider that an F_1 score below 0.6 is unacceptable, while above 0.7 is within the error tolerance for operational purposes. Regarding those scores, the performance stagnates to scores below the acceptance level for disasters like hurr-michael, sunda-tsunami, pinery-bushfire, and portugal-wildfire. These disasters' scores were among the lowest before

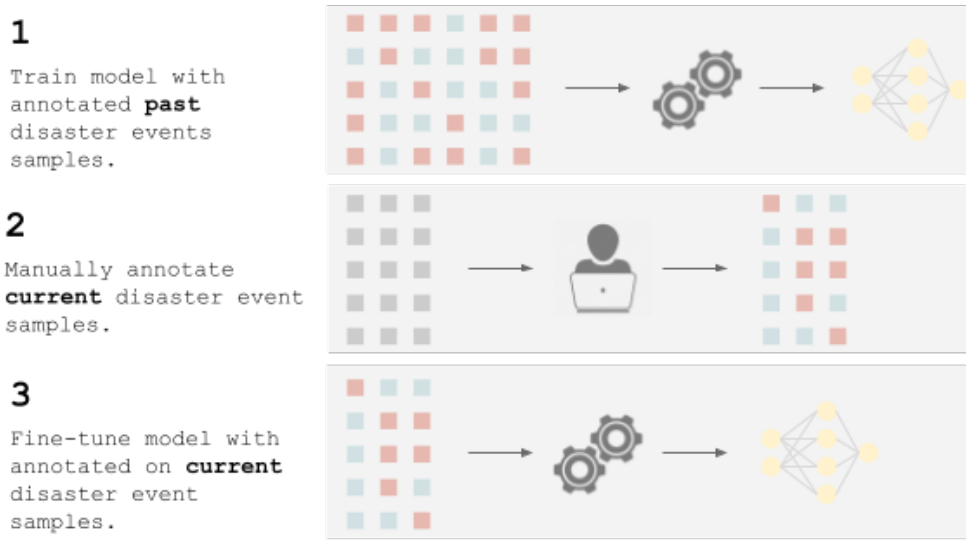

Figure 3.15 Fine-tuning steps.

fine-tuning and the method did improve those scores. However, results suggest that the training distribution is too far from the test distribution for the weights to simply be readjusted with few samples. In contrast, hurr-harvey, which also had a low initial score, impressively benefits from the fine-tuning approach with very few samples.

The fine-tuning method saves considerable time compared to manual annotation (Figure 3.17). The approach relies on the pre-training of *DamageNet* in the pre-incident preparation phase; however, it still involves tedious annotation. Depending on the number of samples to annotate, the duration of the method varies greatly.

The results show that xBD alone is not diverse enough to help with damage classification within the proposed workflow. Some more straightforward use cases (sr-fire, joplin-tornado, and more) proved the method's feasibility. However, the performance level is still not convincing enough among all disaster events for such a solution to be deployed in an emergency. Although data gathering and annotation are tedious, the time investment is essential for the long-term applicability of machine learning in supporting damage assessment. Additional data should include more instances of damage types and season changes.

3.8.3 Proposed incident Workflow

Figure 3.18 summarizes the final incident workflow supported by machine learning. In the pre-incident phase, both the building detection and damage classification models are pre-trained in order to be ready to be queried at any time. The post-incident execution phase

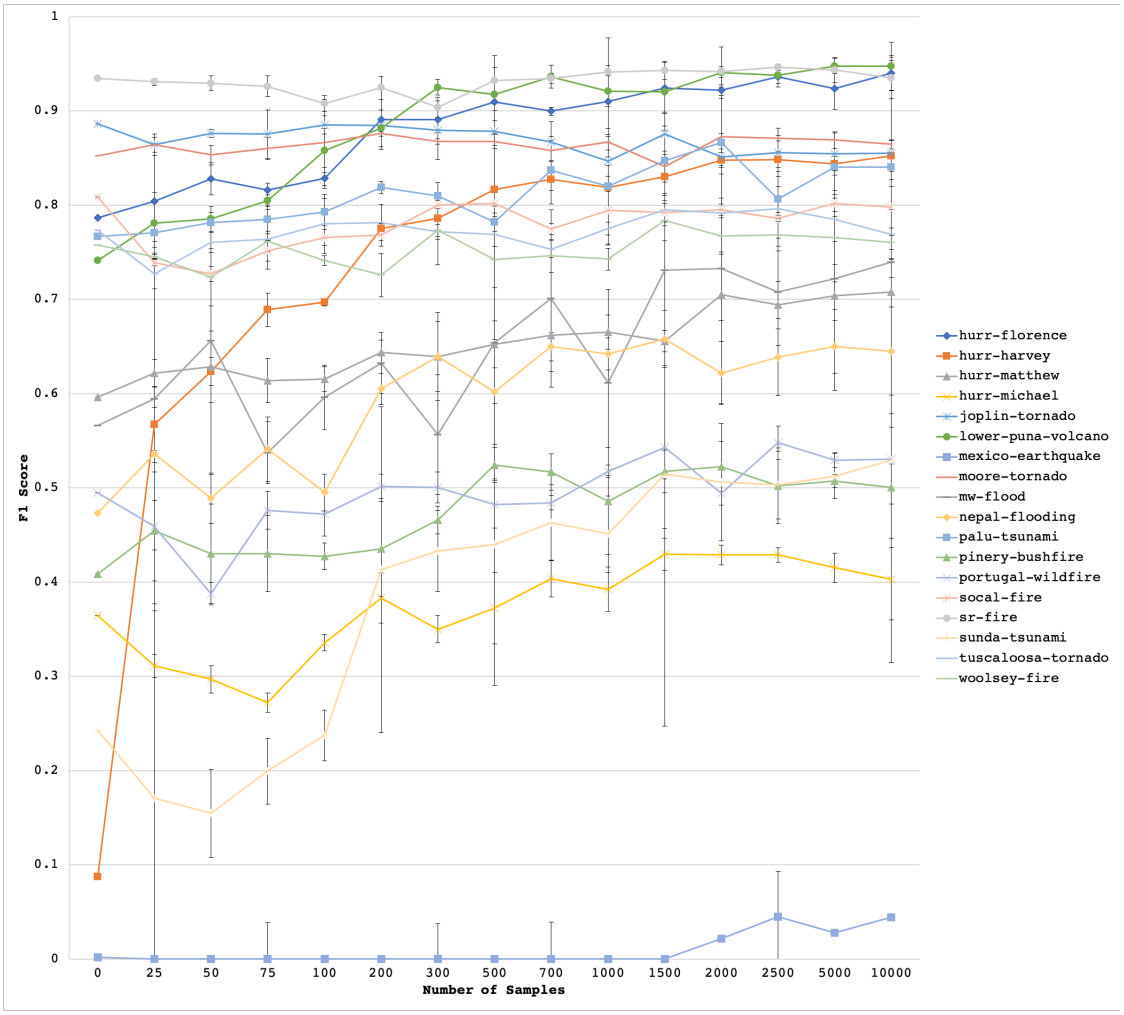

Figure 3.16 Results of *DamageNet* fine-tuned with supervision on annotated samples of the current disaster event. Each line represents the F_1 score for a given disaster event with an increasing number of samples from the current disaster.

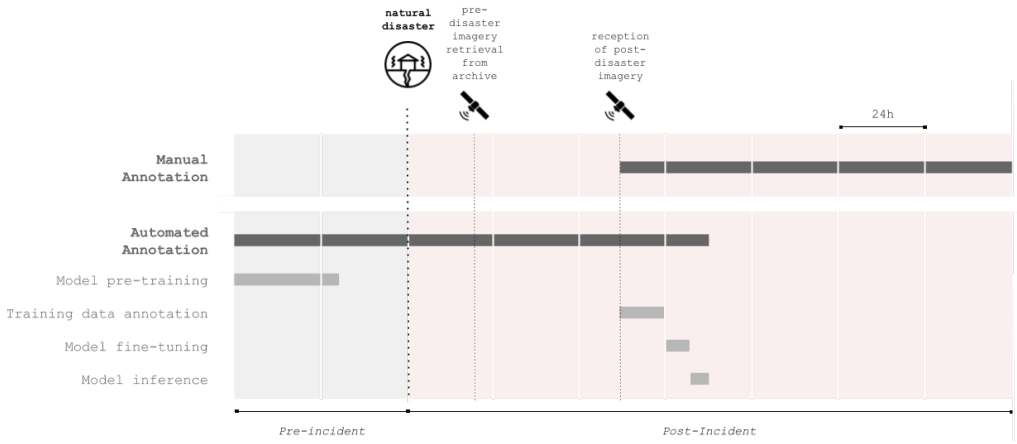

Figure 3.17 Comparison of manual and automatic damage classification Incident Workflows. Manual annotation takes up to days after the reception of post-disaster satellite images. Supervised fine-tuning still involves manual annotation but for more than 10 times fewer samples. All durations are approximate. Data annotation durations are relative to each other.

is triggered by the acknowledgement of a natural disaster. Quickly, the pre-disaster satellite images are retrieved, and the building detection model can predict the building locations in the area under investigation. Once building locations are known, the process awaits for post-disaster satellite imagery. Only then, the damage classification process may start. First, a minimum of 1500 buildings are annotated with damage classification: damage or no damage. Then, the damage classification model is fine-tuned and ready to infer damages for the entire affected area.

Ultimately, produced damage maps and demographic data are paired and analyzed to extract relevant information and support decision-making.

3.9 Conclusion

Natural disasters make affected population vulnerable, potentially affecting their shelter and access to clean water and food. Humanitarian organizations play a critical role in rescuing and assisting people at risk, demanding a high level of preparedness and exemplary processes. Building damage assessment is the process by which humanitarian authorities identify areas of significant concerns. It directly informs decision-making to mobilize resources in these time critical situations.

In this work, we proposed to leverage machine learning techniques to optimize the post-incident workflow with a two-step model approach composed of a building detector and

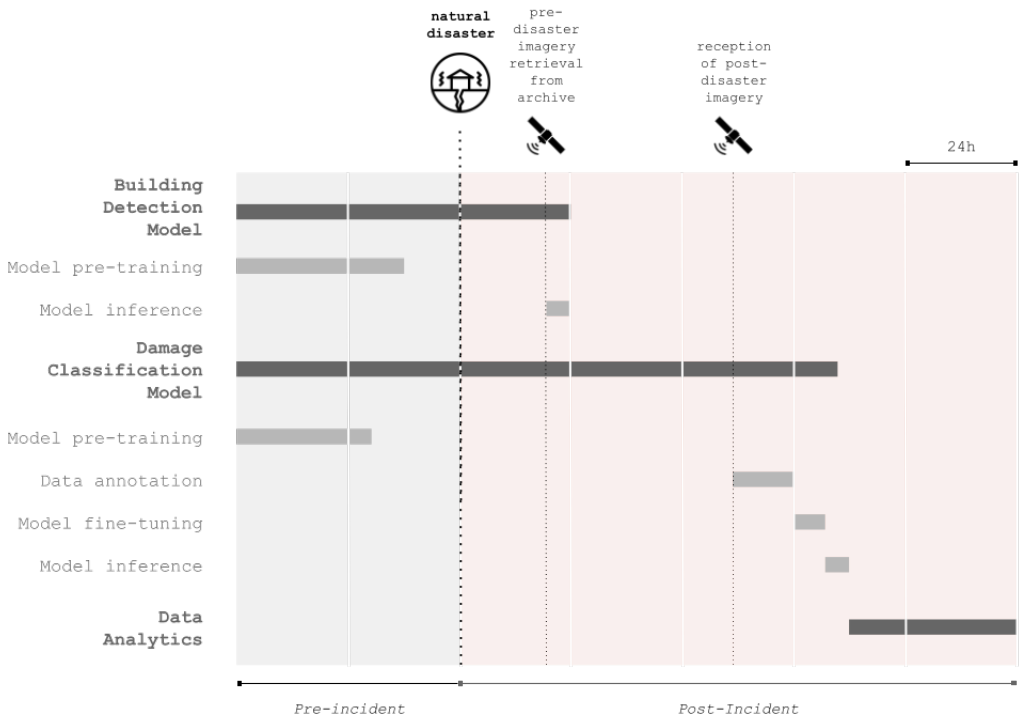

Figure 3.18 Complete Building Damage Assessment Incident Workflow supported by Machine Learning. Building detection inference depends on the pre-disaster satellite images only. Damage classification depends on both the pre- and post-disaster images. It also depends on building detection model inference. Data analytics depend on the damage classification model inference. All durations are approximative.

a damage classifier. We have shown that our approach effectively shortens the damage assessment process compared to the manual annotation of satellite images. Our approach is designed for emergency context and takes into account time and data limitations. Our approach is designed for emergency context and takes into account time and data limitations.

First, we have shown that building detection is generalizable across locations. As a result, the building detector training may be performed during the pre-incident preparation phase, and the model may infer building location immediately after the event. However, our experiments showed a bias towards locations that are over-represented in the training set. Therefore, we advocate for a dataset intentionally sampled regarding population over-exposed to natural disasters. Future work for building detection should focus on training on a more extensive collection of images annotated with building polygons, and more importantly on more balanced datasets.

Also, we have recognized through our extensive experiments across locations that damage classification is a high-dimensional problem that must be handled as a domain adaptation problem. A model solely trained on past disaster events is not guaranteed to detect damages on a newly unfold disaster event. The diversity in climate, disaster type and seasonal changes would require a massive dataset: the 18 disaster events represented in the xBD dataset are insufficient to represent the global diversity. We think that annotated data for damage assessment still represents a critical bottleneck in developing machine learning models for production. To overcome this, we proposed to fine-tune the model's weights on the current disaster events. The approach boosts the model performance with only 1500 annotated buildings, representing roughly 8% of the average coverage. In practice, this significantly reduces the time to respond to a natural disaster compared to manual annotation. In practice, this significantly reduces the time to respond to a natural disaster compared to manual annotation.

Nevertheless, we believe that unsupervised or weakly supervised domain adaptation approaches are well suited for urgent situations and should be considered for further investigation [50–52]. The damage detection task is tightly coupled to the emergency context; therefore, any effort to increase the performance should consider the post-disaster execution time equally. Ultimately, the combination of multiple sources of information (drone and multi-spectral remote sensing, social media posts, etc) may provide a more complete overview of the situation.

Finally, disaster relief deserves the scientific and research community's attention to contribute to the humanitarian effort. Through this work, we aim to raise awareness in the machine learning community for the challenges of applying deep learning in Humanitarian Assistance

and Disaster Response. It is crucial to design solutions with operational conditions in mind and to acknowledge the diversity of the damages caused by natural disasters.

Acknowledgements

A very special thanks to Marco Codastefano and Thierry Crevoisier from the World Food Programme for their continuous feedback during the course of this project. We would also like to acknowledge the contribution of Element AI who provided resources throughout the project.

3.10 Appendix - *BuildingNet* Results

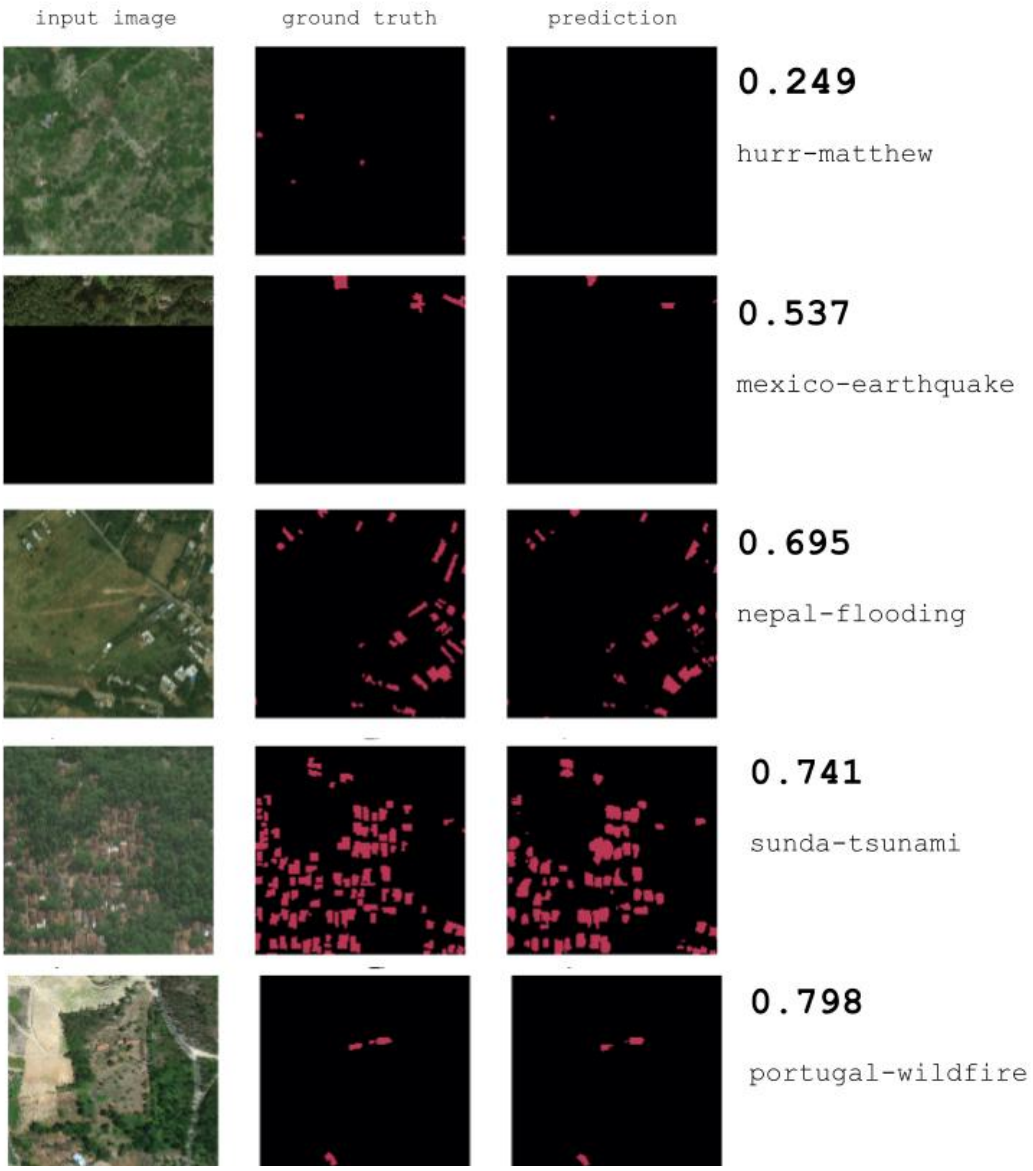

Figure 3.19 Pre-disaster samples from different disaster events along with the ground-truth and *BuildingNet* prediction. Samples are from the five disaster events on which *BuildingNet* **performs the worst**.

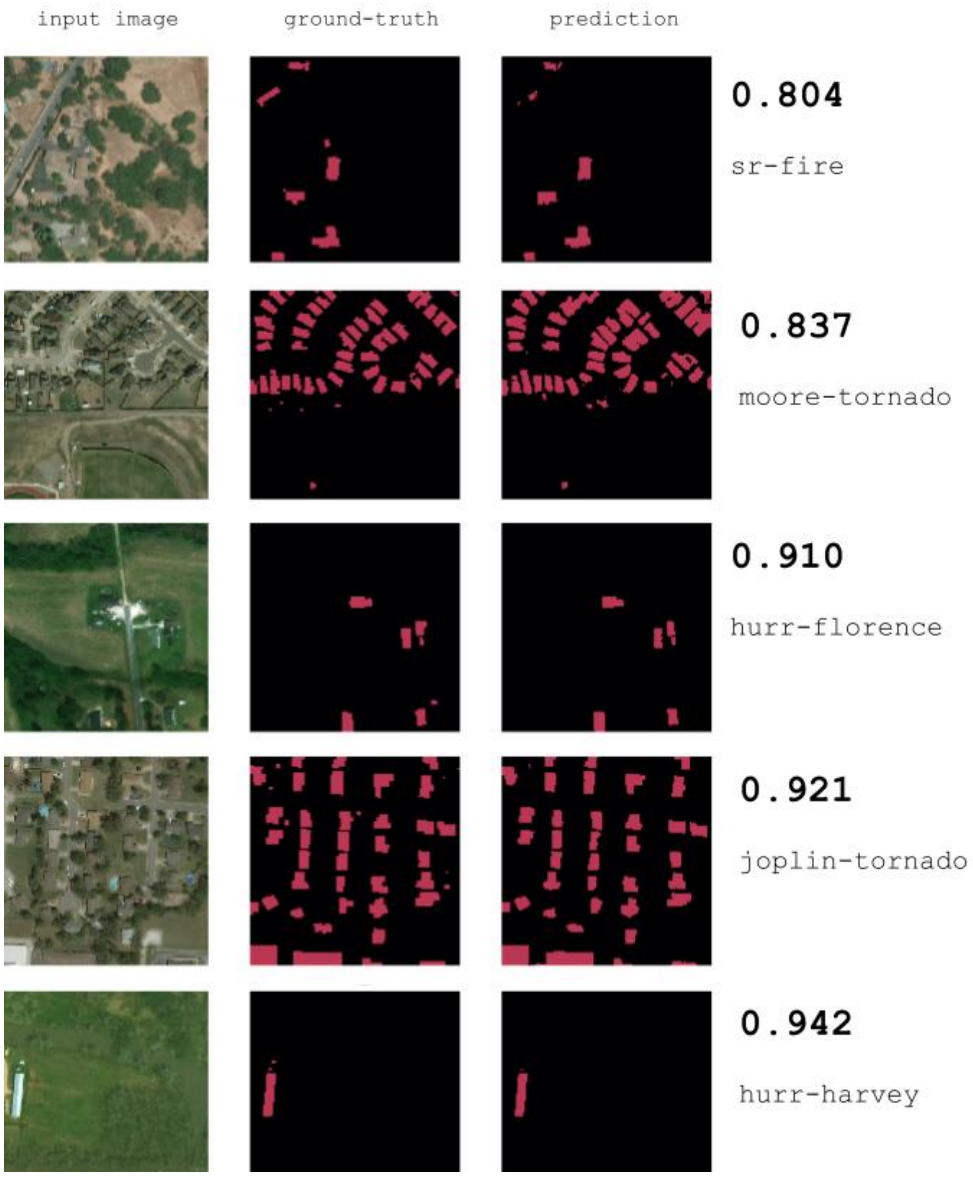

Figure 3.20 Pre-disaster samples from different disaster events along with the ground-truth and *BuildingNet* prediction. Samples from the five disaster events on which *BuildingNet* **performs the best**.

CHAPTER 4 SEMI-SUPERVISED LEARNING EXPERIMENTS FOR DAMAGE CLASSIFICATION

This chapter presents semi-supervised experiments applied to the damage classification task. As discussed in the article presented in Chapter 3, unsupervised approaches should be investigated in order to reduce the post-incident response delays. In that intent, we now investigate a fine-tuning approach that does not require manual annotation of the current disaster. Similar to the supervised fine-tuning approach, the technique leverages learnings from past disaster events. However, instead of depending on manual annotation to fine-tune the model, we propose to rely on the existing signal and to fine-tune the model on the current disaster event. This technique is called pseudo-labelling.

4.1 Pseudo-labelling

Pseudo-labelling is a simple semi-supervised technique for deep neural networks. It allows for leveraging both labelled and unlabeled sources of data. The idea is to use a model pre-trained on labelled data to generate so-called pseudo-labels for unlabeled data (Figure 4.1). Pseudo-labels correspond to the maximum probability predicted by the pre-trained model. The model can further be trained with labelled and pseudo-labelled samples altogether.

Since the unsupervised fine-tuning does not require human annotation, it is better suited for the time-limited post-incident execution phase.

4.2 Unsupervised Fine-tuning Experiments

The unsupervised fine-tuning approach does not rely on a human annotation to adjust *DamageNet* weights to the current disaster event. Instead, the pre-trained model is used to generate some approximative annotation or so-called pseudo-labels (Figure 4.2).

The hypothesis is that *DamageNet* pre-trained on past disaster samples can emit a signal for damage classification on current disaster samples. This signal is captured by the pseudo-labels to be amplified in the fine-tuning phase.

Similar to the supervised approach, *DamageNet* is first trained on past disaster event samples. Then, the pre-trained model infers predictions for every sample of the current disaster event. This prediction, between 0 and 1, is then thresholded to obtain hard annotation, such that only predictions with high confidence are kept. For instance, if the prediction for a given

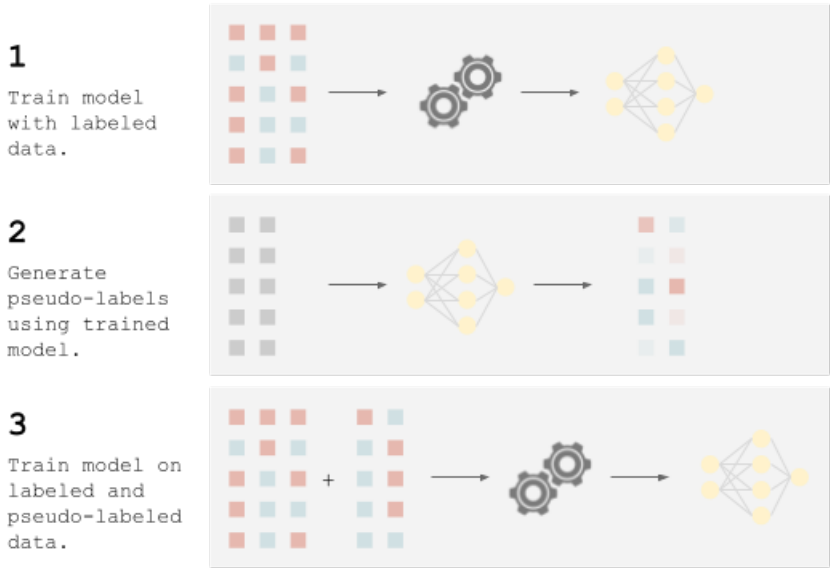

Figure 4.1 Generic semi-supervised pseudo-labelling technique. First (1), a model is trained on labelled samples, then (2) this model is used to generate pseudo-labels for unlabeled samples, and finally (3) the model is retrained with labelled and pseudo-labelled data altogether.

sample is 0.94 confident and the threshold is set to 0.9, the sample will be annotated as `Damaged`. Another sample that predicts 0.88 will not be considered for fine-tuning. We run an ablation study over all disasters, such that every disaster is kept out of the training samples and only used for fine-tuning and testing. Samples used for fine-tuning and testing do not overlap. Each fine-tuning experiment setting is repeated three times with different seed. Weak augmentation is applied on current disaster samples before pseudo-labels inference, and strong augmentation is applied during fine-tuning.

4.3 Results

Figure 4.3 compares the supervised approach with 1.5k annotated samples, the unsupervised approach and the model baseline, i.e. with no fine-tuning. These results show that the unsupervised approach cannot capture enough signal from the pre-trained model to be used as annotation for fine-tuning. The unsupervised method showed some encouraging gains for some disaster events, but fails to guarantee better or even results overall. The score did significantly lower following unsupervised fine-tuning (see nepal-flooding, for instance) and is thus to be used with care.

These results suggest an important gap between the training and testing distributions. The method proposed is thus inefficient to classify damage on a disaster event out of the training

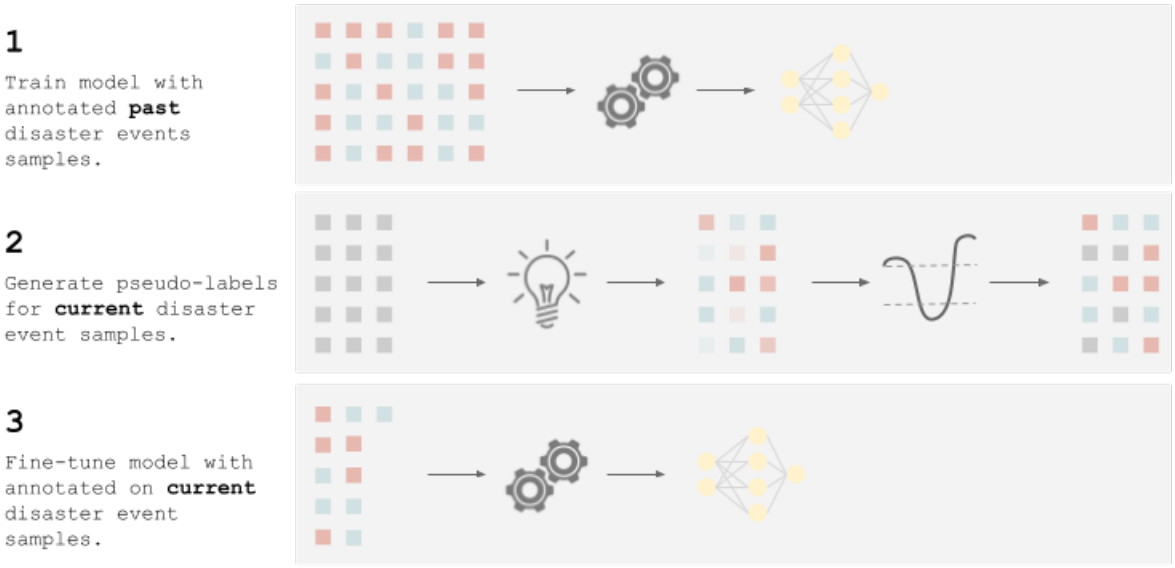

Figure 4.2 Unsupervised fine-tuning steps.

domain. Indeed, for the method to be working, a signal must be captured from the pre-trained model prediction. For out-of-domain samples, the baseline model is simply unable to detect any signal for damage. Even worse, the pre-trained model may confidently mis-classify the test samples. In addition, the approach highly depends on the threshold value set for predictions to be considered confident enough to be carried on for fine-tuning.

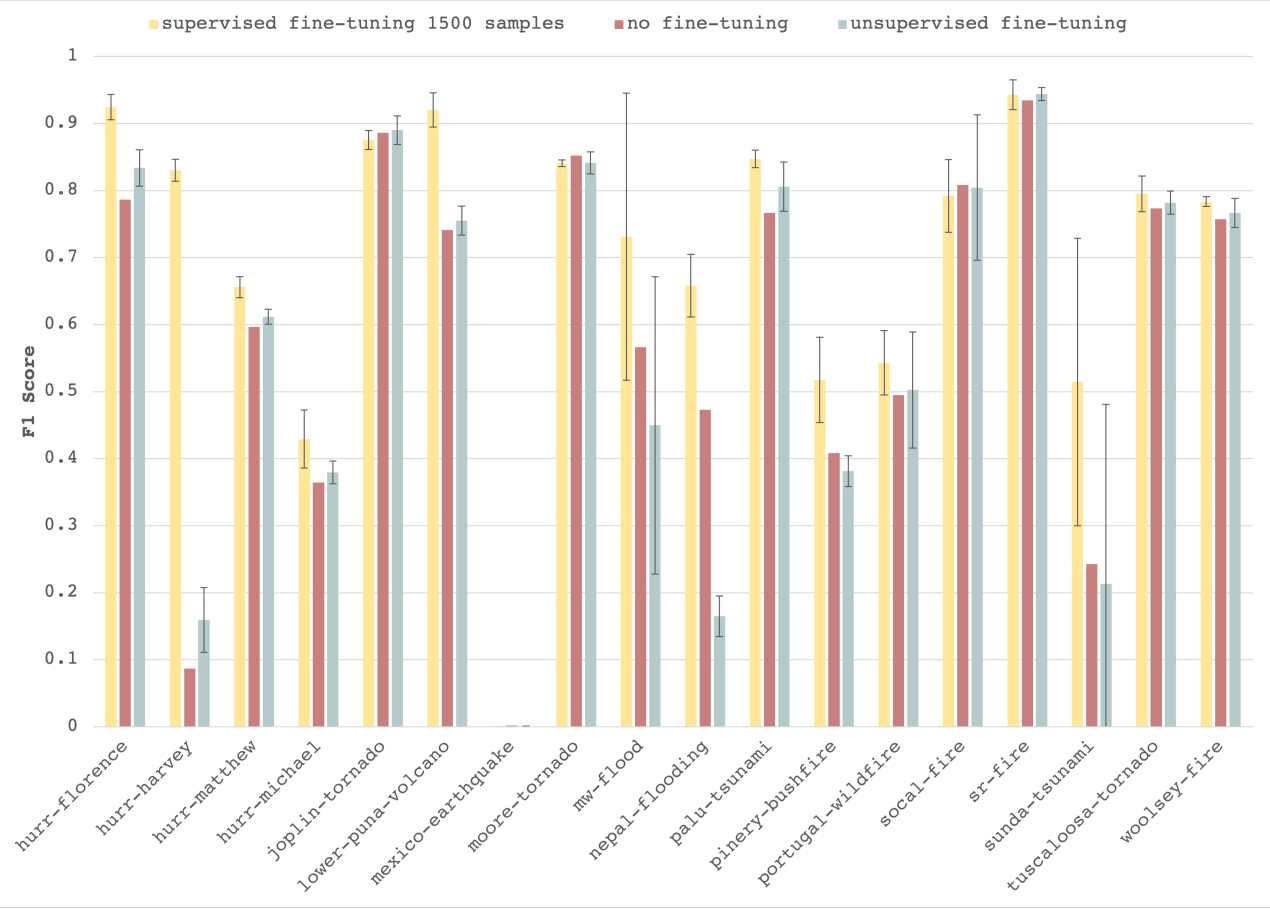

Figure 4.3 Comparison of the fine-tuning approaches (supervised and unsupervised) with the model baseline with no fine-tuning for all 18 disaster events.

CHAPTER 5 DISCUSSION

In this work, we tried to centre our decisions on the emergency context in order to design a method that supports humanitarians when they need it the most. We proposed the use of machine learning to support the damage assessment task and ensured that the approach was consistent with the operational requirements. By and large, we thoroughly discussed how manual annotation prevail as the bottleneck of the post-incident workflow based on remote sensing imagery.

First, we proposed a two-step sequential model made of building localization and the damage classification. This decision was mainly motivated by the access to data and the complexity of the building localization task compared to damage classification. Decoupling the two models allowed for the building localization model to be independent of the current disaster event and, as a result, to be fully trained prior to the disaster. We showed that building localization can generalize across locations and building types. On that account, having the first part of the model ready for inference when a natural disaster occurs already conceded a fair lead compared to fully manual annotation.

Still, damage classification inevitably depends on the post disaster imagery. Not only that, but our experiments showed that the task is undoubtedly more complex; the model does not generalize to new disaster events. In order to adapt the model to new disasters, we proposed to use standard transfer learning techniques and fine-tune the models' weights on a limited number of annotated samples. This approach proved to be valuable in inferring damages, yet, it still depends on manual annotation. Unfortunately, our unsupervised fine-tuning experiments failed to signal that the method was any effective.

All in all, we considered other unsupervised approaches to tackle this challenging problem. Mainly, our work has brought us to converge to domain adaptation solutions. Fundamentally, a disaster event can be thought of as a domain on its own, with its own climate, season, buildings, disaster types and eventually remote sensors. The diversity is too broad to be included in a single dataset, especially given the frequency of natural disaster occurrence to gather new data.

That said, here are three general unsupervised techniques that we have briefly explored and for which we have identified the main difficulties.

5.1 Unsupervised Domain Adaptation Difficulties

First, vanilla unsupervised domain adaptation consists of adding a term to the loss function to classify the original sample distribution (in this case, the disaster event). The loss is back-propagated through a gradient reversal layer to fool the classifier. The learned features are then supposedly invariants to the shift between distributions. In the damage assessment setting, distributions could be the disaster events. This approach works well for domain adaptation settings but has not proved to be efficient in out-of-domain context.

Self-supervision is also an emerging technique for unsupervised domain adaptation. The idea is to generate annotation from the data itself to solve a so-called pretext task. To solve the pretext task, the network must learn visual features. The difficulty of self-supervision is to design a pretext task that serves the downstream task. The damage assessment setting is delicate because of the two-streams siamese architecture and the rotation and colour invariant nature of satellite images. For instance, contrastive approaches are not well-suited for damage assessment because the learned features tend to ignore the background and would thus pass over peripheral damages. There are few prior works applying self-supervision to remote sensing imagery problems [53–55].

Finally, entropy minimization is an unsupervised approach that modulates a pre-trained network signal for a shifted test distribution. It is an elegant extension of a pseudo-labelling approach and relies on the pre-trained model's signal emitted. For the disaster events where there is no signal to be captured, this method is unlikely to be successful.

Other domain adaptation techniques exist and might be worth exploring. The above are merely cautionary advice to keep in mind before approaching damage assessment in an unsupervised manner. Out-of-domains techniques might be worth investigating as well [56, 57].

CHAPTER 6 CONCLUSION

In conclusion, the work done in this thesis contributes to the development of machine learning techniques to support humanitarian activities. It shows the importance of considering the emergency context to develop models that can be used in operational situations. Our work is grounded onto the existing emergency workflow and accounts for its main limitations: the access to post-disaster imagery and the urgency to infer predictions after the event. We developed a machine learning-based workflow to support decision-making that is more efficient and allows for a more rapid response on the field.

The solution proposed is a first step towards an automated building damage assessment. It involves training two different models on historical disaster imagery in preparation to a future event. The first model locates buildings given an image at any point in time before the disaster. The inference can therefore happen immediately after the emergency protocol is triggered and engendered non-significant delays. Building localization was shown to be a task that generalizes well across disaster. The second model assess damage to the detected buildings. It requires both pre- and post-disaster imagery and, therefore, can only really infer predictions when post-disaster imagery is made available. This can cause important delays, when environmental factors such as smoke or cloud coverage obstruct the view from above. Moreover, building damage assessment was shown to perform poorly on some out-of-distribution disasters. To prevent this, we suggest to annotate a small number of post-disaster images and fine-tune the model's weights before inference. Indeed, the manual annotation engender significant delays, though, all in all, the solution proposed shorten the response time compared to a fully-manual annotation approach.

The bulk of this thesis was presented as part of the manuscript submitted to Expert Systems with Applications. In this manuscript, we reviewed the humanitarian emergency context to make sure the the solution proposed would align. Then, we conducted an in-depth study of damage assessment data. We recognized the complexity of the task, given the diversity of climate and seasons, disaster types, construction types, etc. Next, we presented baseline models for both building localization and damage classification, and evaluated the ability of the models to generalize to a forthcoming disaster. To do so, we conducted an extensive ablation study across all eighteen disaster events available in the xBD dataset. We trained models on all dataset except for one, and test on the remaining. These experiments showed good generalization for the building localization model, but a poor one for the damage assessment model. We proposed to use standard transfer learning techniques to mitigate the gap in data

distributions. Starting off of our baseline model for damage assessment, we fine-tuned the weights with supervision on small number of samples from the same disaster as the test set. In order to limit the manual annotation, we fine-tuned on an increasing number of annotated samples to estimate the minimal required amount. Finally, we integrated our solution into an emergency workflow to ensure that the solution was practical.

To completely avoid manual annotation of the current disaster event, we conducted follow-up experiments on unsupervised fine-tuning approaches. The idea is to use the signal produced by our baseline model to generate so-called pseudo-labels, and fine-tune the model's weights. Unfortunately, these experiments have not shown to be promising given the amount of data available for training. In fact, we argue that data is the main limiting factor for this approach to be effective.

6.1 Limitations and Future Works

The main limitation of our method is the delay to generate building damage classification annotation for supervised fine-tuning in the post-disaster execution phase. Not only manual annotation directly engender delays, but it also requires a well ordered coordination amongst people involved in the process. It results in extended disaster relief response time.

Moreover, the overall model accuracy could be further improved. Simple tweaks to data pre-processing or model's architectures and hyper-parameters could potentially lead to significant gains.

Finally, the work done in this thesis assumes training and inference occurs on Graphics Processing Units (GPU). Processing on Central Processing Units (CPU) would significantly increase both training and inference time as they are not optimized for costly DL models matrix operations.

Without a doubt, future work should first and foremost focus on collecting a more diverse set of annotated data for damage assessment. From our experiments, it is clear that data is a limiting factor in the development of machine learning approaches for damage assessment in an emergency context. A larger, but mostly more diverse, dataset would result in a stronger baseline, and eventually enable unsupervised approaches.

Also, we believe that active learning could be used to further improve our solution. Active learning [58] is an iterative supervised learning technique where a pre-trained model query a human to annotate the most uncertain samples from a pool of unlabeled samples. These newly annotated samples are gradually added to the model training. This method generally leads to higher accuracy with fewer annotated samples and aligns with our supervised fine-

tuning approach.

In addition, we believe that unsupervised domain adaptation should be prioritized for their ease of use in the emergency context. Amongst other solutions, self-supervision could be investigated. More and more remote sensing-specific self-supervision approaches are published and would be relevant to apply within humanitarian applications.

Finally, through this work, we aim to raise awareness in the ML community fot the importance of considering the environment in which solutions are to be deployed. Humanitarian agencies operate in chaotic emergency environment and the solutions developed to support their processes should consider it.

REFERENCES

[1] R. Gupta *et al.*, "xbd: A dataset for assessing building damage from satellite imagery," *arXiv preprint arXiv:1911.09296*, 2019.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[3] A. J. Cooner, Y. Shao, and J. B. Campbell, "Detection of urban damage using remote sensing and machine learning algorithms: Revisiting the 2010 haiti earthquake," *Remote Sensing*, vol. 8, no. 10, p. 868, 2016.

[4] A. Fujita *et al.*, "Damage detection from aerial images via convolutional neural networks," in *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, 2017, pp. 5–8.

[5] 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, no. 9, p. 1123, 2019.

[6] J. Doshi, S. Basu, and G. Pang, "From satellite imagery to disaster insights," *arXiv preprint arXiv:1812.07033*, 2018.

[7] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, "Spacenet: A remote sensing dataset and challenge series," *arXiv preprint arXiv:1807.01232*, 2018.

[8] I. Demir *et al.*, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 172–181.

[9] J. Z. Xu *et al.*, "Building damage detection in satellite imagery using convolutional neural networks," *arXiv preprint arXiv:1910.06444*, 2019.

[10] "DIUx-xView," `https://github.com/DIUx-xView/xView2_first_place`, Accessed: 2021-02-15.

[11] J. Shao *et al.*, "Bdd-net: A general protocol for mapping buildings damaged by a wide range of disasters based on satellite imagery," *Remote Sensing*, vol. 12, no. 10, p. 1670, 2020.

[12] R. Gupta and M. Shah, "Rescuenet: Joint building segmentation and damage assessment from satellite imagery," *arXiv preprint arXiv:2004.07312*, 2020.

[13] E. Weber and H. Kané, "Building disaster damage assessment in satellite imagery with multi-temporal fusion," *arXiv preprint arXiv:2004.05525*, 2020.

[14] H. Hao *et al.*, "An attention-based system for damage assessment using satellite imagery," *arXiv preprint arXiv:2004.06643*, 2020.

[15] Y. Shen *et al.*, "Cross-directional feature fusion network for building damage assessment from satellite imagery," *arXiv preprint arXiv:2010.14014*, 2020.

[16] J.-B. Boin *et al.*, "Multi-class segmentation under severe class imbalance: A case study in roof damage assessment," *arXiv preprint arXiv:2010.07151*, 2020.

[17] T. Valentijn *et al.*, "Multi-hazard and spatial transferability of a cnn for automated building damage assessment," *Remote Sensing*, vol. 12, no. 17, p. 2839, 2020.

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

[19] Y. Li *et al.*, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.

[20] B. Athiwaratkun *et al.*, "There are many consistent explanations of unlabeled data: Why you should average," *arXiv preprint arXiv:1806.05594*, 2018.

[21] J. Lee *et al.*, "Assessing post-disaster damage from satellite imagery using semi-supervised learning techniques," *arXiv preprint arXiv:2011.14004*, 2020.

[22] K. Sohn *et al.*, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[23] F. Nex *et al.*, "Structural building damage detection with deep learning: Assessment of a state-of-the-art cnn in operational conditions," *Remote sensing*, vol. 11, no. 23, p. 2765, 2019.

[24] S. Voigt *et al.*, "Global trends in satellite-based emergency mapping," *Science*, vol. 353, no. 6296, pp. 247–252, 2016.

[25] S. Ben-David *et al.*, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.

[26] "United Nation World Food Programme (WFP)," http://www.wfp.org, accessed: 2021-02-15.

[27] "The nobel peace prize 2020 - press release," http://https://www.nobelprize.org/prizes/peace/2020/press-release/, accessed: 2021-02-15.

[28] D. Rolnick *et al.*, "Tackling climate change with machine learning," *arXiv preprint arXiv:1906.05433*, 2019.

[29] L. Rausch *et al.*, "A holistic concept to design optimal water supply infrastructures for informal settlements using remote sensing data," *Remote Sensing*, vol. 10, no. 2, p. 216, 2018.

[30] F. Kogan, *Remote sensing for food security*. Springer, 2019.

[31] M. M. Nielsen, "Remote sensing for urban planning and management: The use of window-independent context segmentation to extract urban features in stockholm," *Computers, Environment and Urban Systems*, vol. 52, pp. 1–9, 2015.

[32] F. Filipponi, "Exploitation of sentinel-2 time series to map burned areas at the national level: A case study on the 2017 italy wildfires," *Remote Sensing*, vol. 11, no. 6, p. 622, 2019.

[33] G. M. Foody, "Remote sensing of tropical forest environments: towards the monitoring of environmental resources for sustainable development," *International journal of remote sensing*, vol. 24, no. 20, pp. 4035–4046, 2003.

[34] G. J. Schumann *et al.*, "Assisting flood disaster response with earth observation data and products: A critical assessment," *Remote Sensing*, vol. 10, no. 8, p. 1230, 2018.

[35] D. Berthelot *et al.*, "Mixmatch: A holistic approach to semi-supervised learning," *arXiv preprint arXiv:1905.02249*, 2019.

[36] Y. Pi, N. D. Nath, and A. H. Behzadan, "Convolutional neural networks for object detection in aerial imagery for disaster response and recovery," *Advanced Engineering Informatics*, vol. 43, p. 101009, 2020.

[37] C. Xiong, Q. Li, and X. Lu, "Automated regional seismic damage assessment of buildings using an unmanned aerial vehicle and a convolutional neural network," *Automation in Construction*, vol. 109, p. 102994, 2020.

[38] T. G. J. Rudner *et al.*, "Rapid computer vision-aided disaster response via fusion of multiresolution, multisensor, and multitemporal satellite imagery," in *First workshop on AI for Social Good. Neural Information Processing Systems (NIPS-2018), Montreal, Canada.* Curran Associates, Inc, 2018.

[39] C. D. Z. H. e. a. XLi, X., "ocalizing and quantifying infrastructure damage using class activation mapping approaches," *Social Network Analysis and Mining*, vol. 9, no. 44, 2019.

[40] D. Duarte *et al.*, "Satellite image classification of building damages using airborne and satellite image samples in a deep learning approach," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2, pp. 89–96, 05 2018.

[41] K. He, R. Girshick, and P. Dollár, "Rethinking imagenet pre-training," 2018.

[42] D. Mahajan *et al.*, "Exploring the limits of weakly supervised pretraining," 2018.

[43] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[44] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015.

[45] T. Glasmachers, "Limits of end-to-end learning," 2017.

[46] O. Oktay *et al.*, "Attention u-net: Learning where to look for the pancreas," *CoRR*, vol. abs/1804.03999, 2018.

[47] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[48] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[49] K. He *et al.*, "Deep residual learning for image recognition," 2015.

[50] Y. Li *et al.*, "Unsupervised domain adaptation with self-attention for post-disaster building damage detection," *Neurocomputing*, vol. 415, 07 2020.

[51] B. Benjdira *et al.*, "Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images," *Remote Sensing*, vol. 11, no. 11, p. 1369, Jun 2019. [Online]. Available: http://dx.doi.org/10.3390/rs11111369

[52] Q. Xu, X. Yuan, and C. Ouyang, "Class-aware domain adaptation for semantic segmentation of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–17, 2020.

[53] O. Mañas *et al.*, "Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data," *arXiv preprint arXiv:2103.16607*, 2021.

[54] S. Vincenzi *et al.*, "The color out of space: learning self-supervised representations for earth observation imagery," 2020.

[55] A. M. Swope, X. H. Rudelis, and K. T. Story, "Representation learning for remote sensing: An unsupervised sensor fusion approach," 2019.

[56] J. Cha *et al.*, "Swad: Domain generalization by seeking flat minima," *arXiv preprint arXiv:2102.08604*, 2021.

[57] Y. Wald *et al.*, "On calibration and out-of-domain generalization," *arXiv preprint arXiv:2102.10395*, 2021.

[58] H. H. Aghdam *et al.*, "Active learning for deep detection neural networks," 2019.