GVT-BDNet

Convolutional Neural Network with Global Voxel Transformer Operators for Building Damage Assessment

Remondini, Leonardo

Award date:
2021

Awarding institution:
Royal Institute of Technology

DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING, SECOND CYCLE, 30 CREDITS

STOCKHOLM, SWEDEN 2021

GVT-BDNet: Convolutional Neural Network with Global Voxel Transformer Operators for Building Damage Assessment

LEONARDO REMONDINI

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

GVT-BDNet: Convolutional Neural Network with Global Voxel Transformer Operators for Building Damage Assessment

LEONARDO REMONDINI

Master's Programme, ICT Innovation, 120 credits
Date: October 1, 2021

Supervisors: Hao Hu, Sergiu Petre Iliev
Examiner: Gionis Aristides
School of Electrical Engineering and Computer Science
Host company: Spacept
Swedish title: GVT-BDNet: Convolutional Neural Network med Global Voxel Transformer Operators för Building Damage Assessment

Abstract

Natural disasters strike anywhere, disrupting local communication and transportation infrastructure, making the process of assessing specific local damage difficult, dangerous, and slow. The goal of Building Damage Assessment (BDA) is to quickly and accurately estimate the location, cause, and severity of the damage to maximize the efficiency of rescuers and saved lives. In current machine learning BDA solutions, attention operators are the most recent innovations adopted by researchers to increase generalizability and overall performances of Convolutional Neural Networks for the BDA task. However, the latter, nowadays exploit attention operators tailored to the specific task and specific neural network architecture, leading them to be hard to apply to other scenarios. In our research, we want to contribute to the BDA literature while also addressing this limitation.

We propose Global Voxel Transformer Operators (GVTOs): flexible attention-operators originally proposed for Augmented Microscopy that can replace up-sampling, down-sampling, and size-preserving convolutions within either a U-Net or a general CNN architecture without any limitation. Dissimilar to local operators, like convolutions, GVTOs can aggregate global information and have input-specific weights during inference time, improving generalizability performance, as already proved by recent literature.

We applied GVTOs on a state-of-the-art BDA model and named it GVT-BDNet. We trained and evaluated our proposal neural network on the xBD dataset; the largest and most complete dataset for BDA. We compared GVT-BDNet performance with the baseline architecture (BDNet) and observed that the former improves damaged buildings segmentation by a factor of 0.11. Moreover, GVT-BDNet achieves state-of-the-art performance on a 10% split of the xBD training dataset and on the xBD test dataset with an overall F1-score of 0.80 and 0.79, respectively.

To evaluate the architecture consistency, we have also tested BDNet's and GVT-BDNet's generalizability performance on another segmentation task: Tree & Shadow segmentation. Results showed that both models achieved overall good performances, scoring an F1-score of 0.79 and 0.785, respectively.

Keywords

Attention Operators, Convolutional Neural Networks (CNNs), Deep Learning, Building Damage Assessment, Generalizability, Global Voxel Transformer

Operators (GVTOs).

Sammanfattning

Naturkatastrofer sker överallt, stör lokal kommunikations- och transportinfrastruktur, vilket gör bedömningsprocessen av specifika lokala skador svår, farlig och långsam. Målet med Building Damage Assessment (BDA) är att snabbt och precist uppskatta platsen, orsaken och allvarligheten av skadorna för att maximera effektiviteten av räddare och räddade liv.

Nuvarande BDA-lösningar använder Convolutional Neural Network (CNN) och ad-hoc Attention Operators för att förbättra generaliseringsprestanda. Nyligen föreslagna attention operators är dock specifikt skräddarsydda för uppgiften och kan sakna flexibilitet för andra scenarier eller neural nätverksarkitektur.

I vår forskning bidrar vi till BDA -litteraturen genom att föreslå Global Voxel Transformer Operators (GVTO): flexibla attention operators som kan appliceras på en CNN -arkitektur utan att vara bundna till en viss uppgift. Nyare litteratur visar dessutom att de kan öka utvinningen av global information och därmed generaliseringsprestanda.

Vi tillämpade GVTO på en toppmodern CNN-modell för BDA. GVTO: er förbättrade skadessegmenteringsprestandan med en faktor av 0,11. Dessutom förbättrade de den senaste tekniken för xBD-testdatauppsättningen och nådde toppmodern prestanda på en 10% delning av xBD-träningsdatauppsättningen. Vi har också utvärderat generaliserbarheten av det föreslagna neurala nätverket på en annan segmenteringsuppgift (Tree Shadow segmentering), vilket uppnådde över lag bra prestationer.

Nyckelord

Attention Operators, Convolutional Neural Networks (CNNs), Deep Learning, Building Damage Assessment, Generalizability, Global Voxel Transformer Operators (GVTOs).

Acknowledgments

I would like to thank my examiner, supervisors, and particularly Sergiu Iliev, CTO of Spacept, for helping me throughout the development of the master thesis and for being available at any time. I also thank Spacept's team for allowing me to exploit the Spacept Tree & Shadow dataset.

Finally, I would like to thank my family, Adriana, Roberto, Ilaria, and Chiara for always supporting me, and my friends, Mattia, Nicoleta, Tommaso, and Sebastiano, for making me enjoy this last year of master quarantine life with wonderful chats and memes.

Stockholm, October 2021
Leonardo Remondini

Contents

List of Figures

List of Tables

List of acronyms and abbreviations

AI Artificial Intelligence

BDA Building Damage Assessment

BDNet Building Damage Neural Network

CC Concatenated Channel model

CNN Convolutional Neural Network

CNNs Convolutional Neural Networks

DL Deep Learning

GVTO Global Voxel Transformer Operator

GVTOs Global Voxel Transformer Operators

ML Machine Learning

MLP Multilayer Perceptron

PO Post-image Only model

SAR Synthetic Aperture Radar

T&S Tree and Shadow segmentation

TSNet Tree Shadow Neural Network

TTC Twin-Tower Concatenate model

TTS Twin-Tower Subtract model

Chapter 1

Introduction

This chapter introduces the problem of Building Damage Assessment (BDA) and discusses the state-of-the-art solutions and their limits. Moreover, it summarizes the technical background needed to understand the thesis and outlines the thesis's goals, motivation, and contribution.

We start with an introduction to the problem background, a theoretical introduction to Machine Learning (ML) and other essential techniques that define the foundation of which this thesis builds on. Next, we present a discussion regarding satellite data analysis, data types, and what is the most suitable one for BDA in terms of spatial and temporal resolution. We continue studying the limitation of the latest solutions and how they can be tackled at a high level. Finally, we outline the goals, motivation, contribution of the thesis, and the thesis structure.

1.1 Problem Background

Long before the industrial revolution, humanity started to alter the Earth's environment, with an exponential and disruptive incremental tendency in the last decades. Nowadays, the impact of human activities on the Earth ecosystem can be seen almost everywhere on Earth: the global atmosphere, the world ocean, lands, and particularly the global temperature. As of 2019, the global temperature increased by 1 °C above the pre-industrial level. With the current rise of warming (0.2°C per decade), global warming will reach 1.5°C between 2030 and 2052 [9].

Even though this temperature rise might appear small, it has catastrophic consequences for the world's climate and ecosystem. One of the problems caused by the rising of global temperature is that it increases the amount,

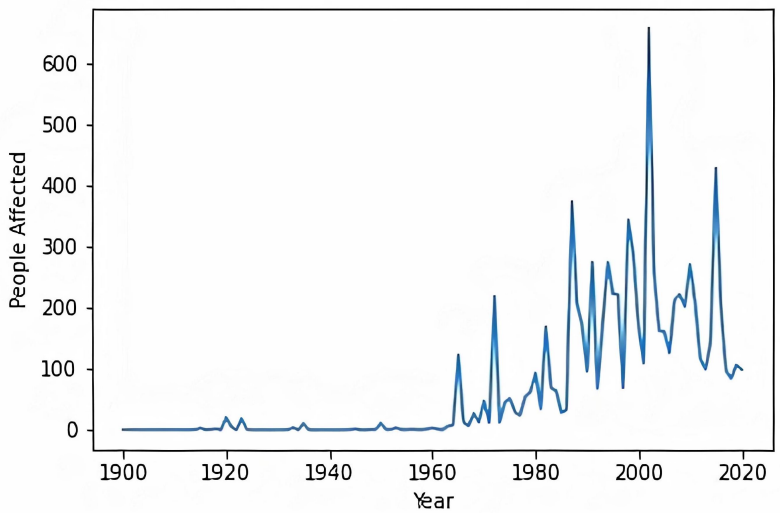

Figure 1.1: *Number of people affected by natural disasters yearly from 1900 to 2020 . The data has been collected from the EM-DAT disaster database [1]*

intensity, and destructive power of natural disasters such as tornadoes, floods, and wildfires. As a consequence, the number of people affected by such catastrophic events has also increased, producing more and more life losses, economic damages, and property damages worldwide. Figure 1.1 shows the trend of people affected by natural disasters per year from 1900 to 2020. We can see that from the 1960s to 2020, the trend gradually increased [1].

Nowadays, natural disasters kill around 90,000 people every year and affect nearly 160 million people worldwide, according to the World Health Organization [10]. Given the growing number of occurrences and the rising intensity of natural disasters, today, more than ever, immediate and accurate post-disaster workflows are needed to increase the efficiency of resource deployment in order to maximize the number of saved lives. In such scenarios, responders need to know basic damage information such as location, cause, and severity, etc before they can act.

Satellite imagery offers a powerful source of information to mitigate the hazardous effect of natural disasters. It can be used to assess the extent and areas of damages of wide geographic regions with restricted time delay. Currently, after a natural disaster strikes, satellite and aerial images are annotated for building damage manually for days or weeks, which is expensive in both time and labor costs. This creates an analytical bottleneck in the post-disaster workflow [11][12][13] that makes damage analysis impossible to consult before rescues are sent. Moreover, after a natural disaster strikes,

many casualties are caused by untimely rescuers. According to D. Vukovic et al. [14], the first 72 hours are crucial to finding disaster survivors. As time passes, the chances to find them alive decrease. In the Haiti earthquake of 2010, CNN reports that one of the major causes of death was a condition called Rhabdomyolysis [15]. It occurs when the muscles get crushed and rupture, causing kidney failure, which may cause death if not treated promptly.

Hence, right after a disaster strikes, the consulting of a reliable and complete building damage analysis of the affected area is crucial to better deploy resources as it helps sending humanitarian support to where damages are concentrated the most.

1.2 Technical Background

1.2.1 Machine Learning and Deep Learning for BDA

Traditional data analysis for BDA typically relies on ground-based assessments, which require a tremendous amount of labor, manual work, and are difficult and time-consuming to obtain. From the 2010s, the scientific community has started to research the automation of BDA using machine learning algorithms, as it enables immediate results and the optimization of rescue plans. Nevertheless, the integrity and immediacy of building damage analysis profoundly depend on the structure of the model and the data type used.

Within building damage analysis, the input data are sequences of either images or video frames. From examples of historical visual disaster data, ML models gradually learn how to classify the intensity of building damage from feature extraction. Deep Learning (DL) is a branch of Machine Learning that introduces artificial neural networks and is specialized in the extraction of high-level features from structured and unstructured data (e.g. images) with representation learning. Convolutional Neural Network (CNN) are deep artificial neural networks specifically tailored to cope with visual data. They have become dominant in various computer vision tasks such as face recognition [16], [17] and brain-tumor detection [18], therefore they have been also widely used within the BDA literature [5][19][20][7].

Convolutional Neural Networks (CNNs) enable feature-learning for imagery data and exploit deep image representations to perform image classification and image segmentation tasks. The architecture of CNNs is analogous to the connectivity pattern of neurons in the human brain and has been inspired by the organization of the visual cortex [4]. However, differently

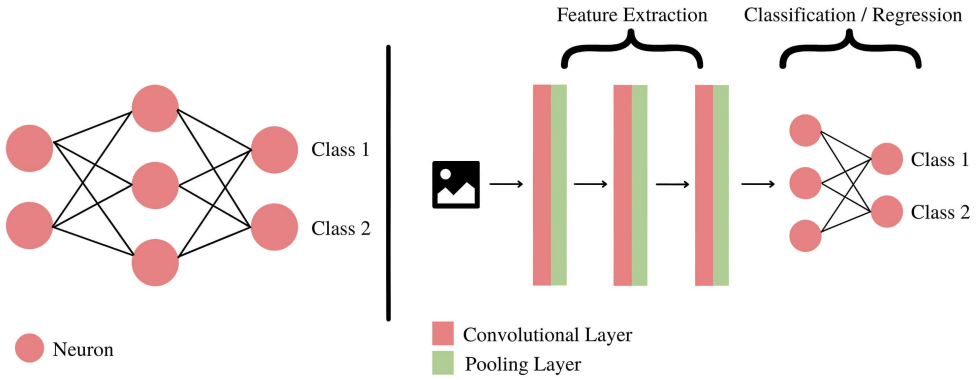

Figure 1.2: *Basic structure of a MLP (right) and basic structure of a CNN (left) for image classification. Due to sequences of convolutional and pooling layers, CNN can better represent spatial and temporal dependencies of an image, thus achieving better performances. The workflow of a CNN is divided into Feature Extraction (convolution and pooling layers) and Classification (typically composed by a fully connected layer).*

from basic neural network architectures, (e.g., Multilayer Perceptron (MLP)), CNNs can successfully capture the spatial dependencies in an image and extract progressively higher level features that enables the learning process. Neurons are organized in layers, and feature maps are the activation of the output of the layers' hidden neurons. In CNNs, they can describe low-level patterns within the shallow layers (colors, edges, basic shapes) and more semantic-rich features within the deepest layers (body parts, vegetation, buildings). Figure 1.2 shows the structure of an MLP compared to the one of a CNN.

A convolutional block can be seen as a single step of the learning process and is usually divided into a convolutional layer and a pooling layer. A typical CNN for image classification consists of a sequence of convolutional blocks followed by a fully connected layer, responsible for feature extraction and the classification task, respectively. When an image is fed into a convolutional block, a sliding window that acts as an $M \times M$ filter is applied over the image. Features are gradually collected by performing a dot operation between the filter and the local patch of the image captured by the sliding filter. The output feature map is then down-sampled to decrease the feature dimensionalities and fed into the following convolutional blocks, which will extract more and more significant image features. Once the last convolutional block is completed, the deepest and highest-level feature map which learns the most semantic-rich patterns, is fed into a fully connected layer to perform the classification task.

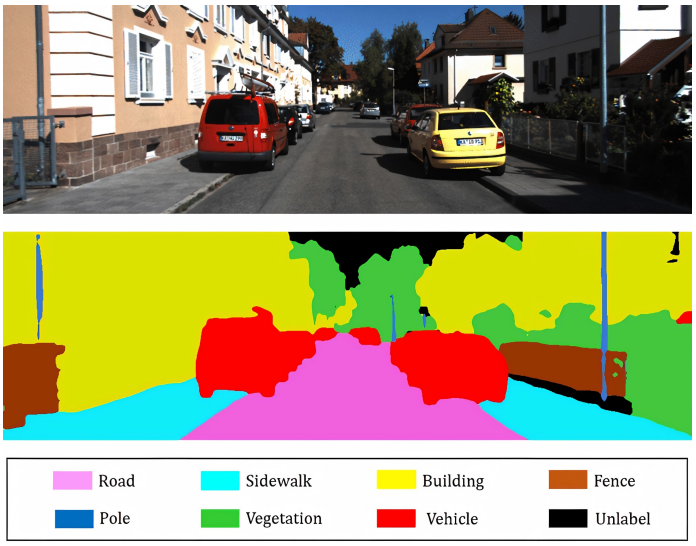

Figure 1.3: *An example of Input and Output of Image Segmentation. Each pixel of the street view figured in the Input has been labeled as either one of the classes listed in the table, and a segmentation mask has been computed accordingly [2]*

Other than image classification, the fully connected layer can be replaced by a different task-oriented architecture. In our case, one can address BDA with a two-step approach. First, buildings have to be detected and segmented from the background. Secondly, they have to be classified based on the severity of the damage. This workflow can be translated into a building detection task followed by a damage classification task, and they can be both addressed with an image segmentation algorithm, whose goal is to assign each pixel of the image a label that belongs to a specific class and output a pixel-wise mask of the image. An example of image segmentation can be seen in Figure 1.3. In our case, we want to classify each pixel of the image as either the background, or a building falling into one of the different damage classes.

Since the output of image segmentation is a pixel-wise mask of the input image, after extracting features with sequences of convolution blocks, we have to gradually return to the image's original size and perform the segmentation task. The ML architecture that is most suitable for image segmentation is the convolutional autoencoder, and it consists of an encoder followed by a decoder and one (or more) output layer(s). The encoder is responsible for feature extraction during down-scaling, and it is equivalent to a feature extractor of a CNN. On the other hand, the decoder takes feature maps outputted by encoder as inputs. Then it gradually up-scale inputs features into their original sizes for

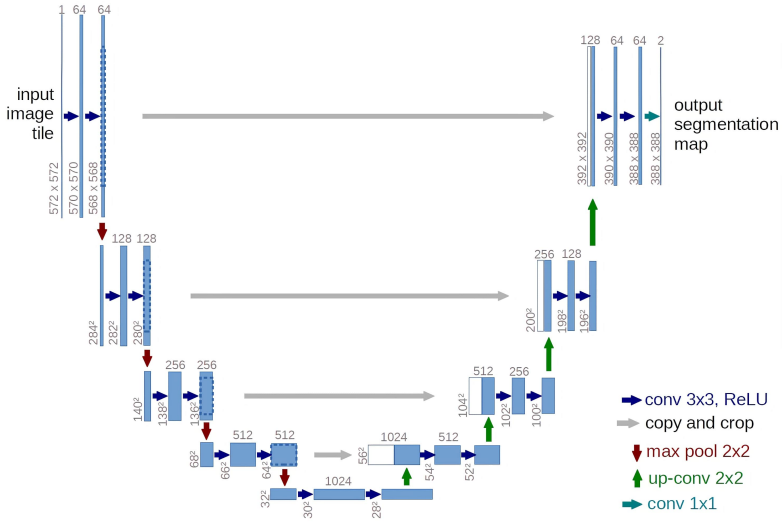

Figure 1.4: *U-Net architecture (example for 32x32 in the lowest resolution). It represents an example of convolutional autoencoder architecture with long skip-connection colored in grey [3]*

pixel-wise classification. Therefore, the decoder's pooling layers are replaced with up-sampling layers, while the convolutional layers are maintained.

Figure 1.4 shows the architecture of U-Net [3], a convolutional autoencoder originally developed for segmenting biomedical images, that the scientific community has widely acclaimed due to its flexibility to similar segmentation tasks, with remarkable performances, such as autonomous driving [21] and the segmentation of satellite images [22]. What makes U-Net innovative is that, besides the canonical autoencoder architecture, it presents long skip-connections, linking the contracting path (encoder) to the expanding path (decoder). Skip-connections recover low-level spatial information lost during down-sampling and enable fine-grained details in the output.

1.2.2 Transfer Learning

A limitation of CNNs, and deep learning models in general, is that they need a significant amount of training time and data to be trained without a priori knowledge, which can be impractical to obtain for BDA. Transfer Learning is a popular technique used in deep learning to remedy this problem, making CNNs and deep learning models more accessible for tasks with small dataset. In Transfer Learning, we first train a base naural network on a base (large) dataset and (supervised) task, and then we either repurpose the learned features

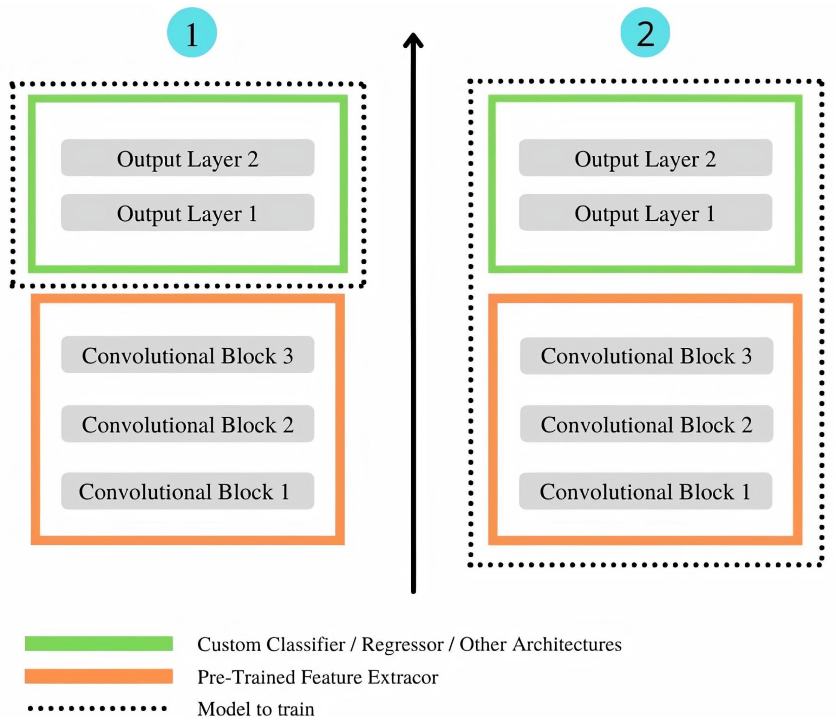

Figure 1.5: *Two approaches of Transfer Learning. With the first approach (left), the classifier's weights are the only ones updated. In the second one (right), both classifier and feature extractor's weights are updated.*

or transfer them to a second target network to be trained on a target dataset with target task. The process is meaningful if the features are general, meaning they can be easily adapted to both base and target tasks, instead of specific to the base task [23]. An example would be using the knowledge of a model that does skyscraper segmentation from satellite images as a starting point to train a general building segmentation model. In this way, we are no longer training our neural network from scratch; instead, we are training the neural network from a pre-existing state, reducing the amount of data needed and saving training time. Moreover, Transfer Learning improves network generalization abilities, as it leverages the base knowledge of models trained with massive input data. This technique is widely used in Computer Vision [24] and Natural Language Processing [25].

Particularly, there are two main approaches to Transfer Learning, as Figure 1.5 shows. The first approach applies when one has little data and wants to be time-effective with great performances. It freezes the Feature Extraction weights of a pre-trained model and uses them as your own Feature Extractor

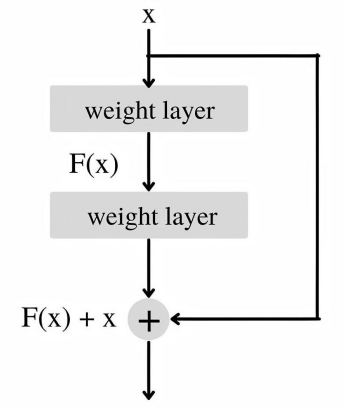

Figure 1.6: *Residual block in ResNet-50. The skip connection bypasses some layers (two in this example) and is added to the output of the very last bypassed layer, which is then fed as input to the following layer. By doing so, we add a bias to the backpropagation and avoid the vanishing of the gradient.*

to train, for example, an image classifier. Doing so, one will update only the weights of a classifier/regressor (or other architectures), thus speeding up the training.The second approach applies when one has large data and wants to push the network's learning process even further to be more adaptied to the task. It consists of training the neural network composed of the pre-trained Feature Extractor followed by a customized task-oriented architecture (e.g., fully connected layer, convolutional decoder).

One of the most famous pre-trained models that is widely used in Transfer Learning is ResNet-50 [26]. It is a 50 layers deep CNN residual network developed by Microsoft which obtains excellent results in image classification and image segmentation. The reason why ResNet-50 is successful relies on its resistance against the vanishing gradient problem, which every deep model encounters during backpropagation.

The latter is an algorithm for supervised learning of artificial neural networks using gradient descent as optimization strategy. Given an artificial neural network and an error function (loss function), it calculates the gradient, which is a multi-variable derivative of the loss function with respect to all the network parameters. The gradient describes the direction in which the loss function increases faster, so neural network's weights are changed in the opposite direction to minimize the loss function. At each iteration, the calculation of the gradient and weights' update proceeds backwards through the network from the output layer to the input layer. In deep neural networks, this process could lead the gradient to drastically diminish in deep learning

models because, as it back-propagates through the network, it may have a minimal effect when it arrives at the lower level layers, and in the worst scenario, it could become zero.

ResNet-50 solved the vanishing gradient problem with the introduction of skip connections between layers. Given a deep neural network composed of X layers, a skip connection copies the output of a layer $x \in X$ and adds it to the output of a layer $x + i$, where i is the number of layers skipped. The final output is then fed as input into layer $x + i + 1$. By doing so, we add a bias to the outcome of layer $x + i + 1$, hence limiting the negative effect of the vanishing gradient problem. Figure 1.6 shows the structure of residual blocks, the primary component of ResNet-50, characterized by skip connections.

As we further analyse in Section 2, BDA is usually tackled by exploiting either CNN or convolutional autoencoders with a pre-trained feature extractor (typically ResNet-50), merging the benefit of convolutional operators with the strength of residual blocks and Transfer Learning.

1.2.3 Attention Operators for CNN

A limit of current CNN-based models and U-Net-like neural networks is that they implement the encoder/encoder-decoder architecture by stacking local operators like convolutions with small kernels, which do not aggregate information from the entire input if its spatial size is larger than the receptive field [4]. Each output unit follows a local path through the network and only has access to the information within its receptive field on the input image [27] [28]. This approach could lead to poor generalizability performances as it fails to capture long-range dependencies, which are crucial for results' accuracy and consistency. Attention operators represent a solution to the limitations of local operators [27].

As shown in Figure 1.7, the main difference between attention operators and local operators is that the former computes each output unit as a weighted sum of all input units, while the second have a receptive field determined by the kernel size. In attention operators, weights are obtained through interaction between different representations of the inputs, making them non-local-operators with a global receptive [28]. Moreover, the weights in attention operators are not fixed after training (as convolutional weights), which makes them input-dependent. In this way, attention operators can leverage extracted information accordingly when transforming different input images.

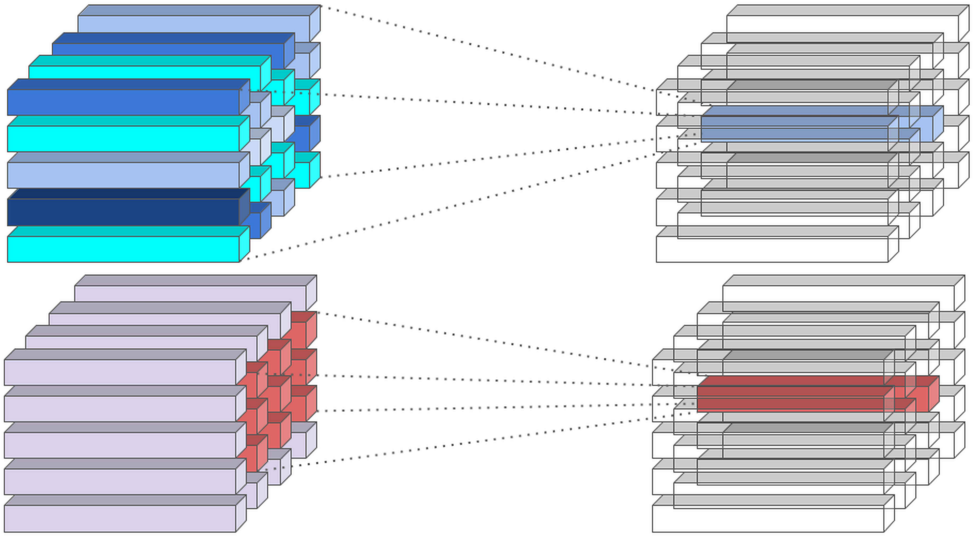

Figure 1.7: *Attention operator (top figure) versus convolutional operator (bottom figure). Note that in the attention operator weights are input-dependent during prediction time, and that there is no limit in the receptive field. On the other hand, convolutions have a limited receptive field determined by a fixed weighted kernel (in this case 3 × 3). [4]*

1.3 Remote Sensing Analysis and Data Imagery

In remote sensing analysis, and ML applications, the choice of a good data type and data source is crucial to train a model architecture efficiently. Here we provide background information on remote sensing analysis, and explain why it is important to understand the data sources for practical applications and understand their limitations in terms of resolution.

Remote sensing refers to technologies for gathering visual information or other data about a site from the air or space. The collected data represents the input of neural networks built for remote sensing analysis. The data types available for remote sensing analysis are categorized based on different resolution, both spatial and temporal. The primary ones are the following:

1. *UAV (Unmanned Aerial Vehicle) Data*: Known also as drones, UAVs offer a broad range of solutions for different applications and are very versatile. They can be equipped with onboard sensors as optical and hyperspectral camera-based, GPS, etc., with low operational costs and very high spatial resolution, down to 1 cm. They are usually adopted

when there is a need for high-resolution images of quite limited areas.

2. *Aircraft Data*: Nowadays, aircraft data is collected by human-crewed aircraft specifically sent for data gathering. Nonetheless, they can give very detailed images of broad areas with few hours of delay. Like UAVs, aircraft can be equipped with onboard sensors, yet their resolution is lower due to the need to fly at higher altitudes (from 0.1 to 6.0km on demand)

3. *Satellite Data*: It is the most time-efficient airborne data source. Satellite data can be gathered online in a few hours and exploited instantly. They are already equipped with specific sensors, but they offer relatively lower image resolution than other airborne sources (commercial satellite imaging is limited to 30 cm/px resolution). They are most suitable when immediate data imagery of huge areas is needed and data resolution is not a priority.

As one might think, satellite data is the most suitable data type for either building damage analysis or other image data analyses in response to a natural disaster, since authorities need immediate post-disaster imagery of the affected area. Moreover, modern satellites offer medium image resolution (up to 10 meters [29]) at low prices or for free. An example is Sentinel 2 that scans the European continent with high-resolution images on a 5-days basis [29]. Costs start rising when real-time data imagery is needed, but it is always guaranteed a pseudo-real-time image availability of vast areas (e.g., big cities, forests, countryside). Note optical satellite images require no clouds, while radar can work for clouds as well.

However, to extract the damage level from each building, post-disaster data must be supported with pre-disaster data. As Google AI researchers discovered, when neural networks are fed with pre-disaster and post-disaster imagery in two different branches, independently of how they are processed through the network (e.g., by concatenation, subtraction), performances generally increase [5]. By comparing pre-disaster images with post-disaster images, neural networks and particularly CNNs can better describe each pixel's damage level depending on the difference between pre-image features and post-image features.

1.4 Problems and Limitations of BDA

One of the biggest challenges of BDA is to build models that are widely applicable across many disasters and countries. That is, an ideal BDA model should predict the damage level without decreasing performances regardless of the building construction style (which may vary depending on where the disaster occurred) and the natural disaster that may have caused the damage (e.g., earthquake, tornado, tsunami). This would enable multiple disaster response agencies to potentially reduce their workload by using a single model with a known deployment cycle.

To build such a model, the training dataset has to include both pre-disaster and post-disaster imagery of multiple disaster types that occurred in different countries. Unfortunately, within the last decade, the limited availability of heterogeneous datasets with a coherent damage scale has constrained the research to non-heterogeneous datasets with fixed resolutions, locations, and disaster types [30][31][32]. Consequently, even though many ML models achieve good performances, they are only specific to locations and disasters, and achieve lower results when transferred to new environments [20]. In 2019, the the Defence Innovation Unit (DIU) of the USA, to assess such a problem, proposed the xView2 Challenge [33] and published the xBD dataset [8], which is the largest BDA dataset to date. It includes five disaster types dislocated in 17 different areas.

Another key challenge of BDA is the great unbalance between damaged buildings, undamaged buildings, and the background within the input data. From a sample of the xBD dataset, the dataset has found to be highly unbalanced towards background pixels (94%), and when looking at building pixels (5.5%), undamaged buildings pixels (4%) were numerically greater than damaged buildings (1.5%). Generally, ML algorithms are expressly required to be trained on balanced training set. So when someone works with highly skewed datasets, models tend to have poor predictive performances, specifically for the minority classes.

Therefore, to assess unbalanced datasets, one may use a few balancing methods. A few techniques widely popular to assess unbalanced datasets are:

- Resampling the training set: Before feeding the network with the data, one may change the dataset class distribution by either oversampling pictures that present a more significant amount of pixels of the minority class or undersampling images that do not present pixels of the minority class.

- Adopting balancing loss functions: Here, one does not modify the input data, but on the other hand, a cost function is defined to penalize wrong classifications of the minority class more than wrong classifications of the majority class. Therefore, the learning process will be skewed towards reasonable classifications of rare classes. Some well-known loss functions that act very well with unbalanced data are the Focal Loss [34] (initially shaped for image recognition tasks) and the Dice Loss [35] (designed for segmentation tasks).

Although many different re-sampling techniques have been largely assessed in recent years, very little research has been done to either assess or compare the most promising balancing loss functions and state which one achieve the overall higher F1-score performance on the building damage segmentation task. Heretofore, researchers mainly focus on new ML model architectures and compared them to the state-of-the-art, but have used the Cross Entropy loss function or its slight variants primarily. Hence, up to now, BDA literature lacks of a clear comparison between balancing loss functions tested on a common neural network architecture.

1.5 Motivation and Goals

This research aims to contribute to the xView2 Challenge and improve upon the state-of-the-art of BDA and generalizability among imbalanced multi-class segmentation tasks. Moreover, we align with #9 (Industry, Innovation, and Infrastructure) and #13 (Climate Action) sustainable development goals for the United Nations [36]. The primary goals and objectives of this study are:

- Discuss and analyze the most promising state-of-the-art architectures for BDA and multiclass segmentation in general, emphasizing solutions for unbalanced datasets and generalizability.
- Benchmark the best performing BDA neural network architecture on the xBD dataset with many balancing loss function (Dice Loss, Focal Loss, Weighted Cross-Entropy, and J regularization [37]).
- Implement state-of-the-art attention operators (Global Voxel Transformer Operators [4]) for the BDA task on the proposed baseline neural network architecture, and test them with the xBD dataset.

- Assess the generalizability and flexibility of the model in a brand new environment within the remote sensing analysis field. Particularly, we test the proposed neural network architecture on another multi-class imbalanced segmentation task: Tree & Shadow segmentation. The latter helps to reduce the risks of power outages and fires sparked by falling trees and storms. The aim is to save lives, reduce CO2 emissions while also radically reducing time and infrastructure inspection costs.

1.6 Research Methodology

Our research presents elements from both a quantitative and qualitative research analysis.

Firstly, we conduct an in-depth literature review to evaluate state-of-the-art BDA neural network architectures and choose the best performing one as the baseline architecture for our study. Then, to collect our results, we conduct experiments by testing various neural network architectures with the xBD dataset, which has been retrieved from the xView2 Challenge website [33]. Results are then analyzed via a quantitative analysis approach. Particularly, the performance of each model is described (and compared with other models) with three main statistical metrics. The discussion of a model's performance is also supported by qualitative results, which are the model's predictions of a dataset input sample.

1.7 Structure of the thesis

The thesis is structured as follows. Chapter 2 presents a discussion and analysis of the state-of-the-art for BDA. It discusses the evolution of BDA neural network architectures and analyses how researchers have coped with generalizability problems and data imbalance. Chapter 3 describes the chosen research methodology and outlines the network architectures proposed. It also motivates each conducted experiment and the thesis workflow. Chapter 4 presents, describes, and discusses obtained results and compares them with the state-of-the-art model for BDA. Finally, chapter 5 summarizes the conclusions, limitations, and future works.

Chapter 2

Literature Review

This chapter summarizes the previously published research for BDA and discusses the state-of-the-art solutions addressing generalization and data imbalance. Furthermore, the discussion highlights literature gaps that are addressed throughout the thesis.

2.1 BDA State-of-the-Art

From the early 2000s to the early 2010s, building damage analysis has been widely adopted by various real world applications such as building reconstruction optimization, post-disaster safety estimation and development of hazard maps [11][12][38][39][40]. The damage analysis was usually conducted manually by visual estimation without Artificial Intelligence (AI), taking days, weeks, or even months to be completed. Such limitations prevent BDA's technology from being directly adopted for improving quick humanitarian assistance, making BDA only serve as an approach for prevent humanitarian and economic disasters in a post-natural disaster environment.

Satellite imagery offers a powerful source of information to mitigate the hazardous effect of natural disasters. It can be used to assess the extent and areas of damages of wide geographic regions with restricted time delay. Synthetic Aperture Radar (SAR) satellite images were among the most commonly used resources for building damage analysis and disaster recovery. In 2012, A. Suppasri et al. [11], exploited SAR satellite images to determine tsunami affected-areas and assess building/vegetation recovery using the reflection property or backscattering coefficient. The latter is a physical value that is strongly dependent on the roughness of the ground surface and is likely to decrease when it is detected after building collapse or

inundation. The authors then classified building damages into four categories by looking at the difference between the backscattering coefficients of pre- and post-disaster images, and supported the results investigating the presence of rooftops manually. After statistical data analysis, the research outcomed a probabilistic damage graph with respect to the inundation depth that could be useful to develop future hazard plans.

In the early 2010s, many other studies focused on the manual and visual interpretation of pre-disaster and post-disaster images with approaches from statistical analysis [12][38][39], which may be statistically effective, but time-inefficient. Because of that, it was not possible to include accurate building damage analyses in the evaluation of rescue plans, even though research shows that it is extremely beneficial. It has been noticed that building damage maps may be a proxy for victim localization [41] and could also aid in planning and delineating more efficient recovery plans [42].

Another limitation of pre-AI BDA solutions is that researchers might had different damage interpretations for the same buildings, as manual interpretation is strictly subjective. Therefore, the outcome of the damage analysis could have been a trade-off between all interpretations, which could have been inaccurate and led to unprecise hazard plans. Researchers from Tohoku University also argued that the disastrous effect of tornados, earthquakes, and other disasters, cannot be statistically predicted with high fidelity [39]. Hence, disaster recovery plans cannot rely solely on hazard plans and statistical studies. For these reasons, with the exponential improvement of AI and the development of CNNs and DL in the late 2010s, researchers started to use DL algorithms to assess building damage. The benefit of CNN over traditional approaches is that it automatically detects discriminative features without any human supervision. It also drastically decreases the amount of time required to process data from weeks to hours, opening prospects to use building damage analysis to plan immediate humanitarian assistance. Moreover, in recent years, it has been demonstrated that using deep neural network models provides higher performances compared to other supervised ML models like Support Vector Machines and Random Forests, especially for high-dimentional data [43].

The firsts CNNs developed for BDA focused on estimating damages of single disasters (mainly earthquakes and tornados), and single locations, using various aerial imagery. M. Ji et al. [44] proposed a method to estimate the number of collapsed buildings given post-earthquake imagery, which could be interpreted as a classification algorithm. The authors developed a CNN based on SqeezeNet, a CNN able to replicate ImageNet [45] performances but with

50x fewer parameters, and achieved 0.807 F1-score on building localization and 0.766 on building classification.

While this approach is suitable for statistical analysis, it could be inefficient for humanitarian assistance, as they would have to compose the hazard map manually. Consequently, most studies tackled BDA as a segmentation problem [5] [6] [19] [30][44][46] since it would have been more accessible for rescuers. The output of BDA is then a building damage map that classifies each pixel as either the background or a building falling into damaged buildings or undamaged buildings. Depending on the dataset, the damaged class can be divided into more specific damage levels.

Beside approaching BDA as a segmentation technique, in recent years it has been showed that the best input format for BDA models is paired pre-disaster and post-disaster imagery, rather than post-disaster imagery only [5] [6]. One of the studies that supports this fact is a research conducted in 2019 by Google Researchers, where they compared the performances of four different CNN models in detecting damaged buildings using the 2010 Haiti earthquake dataset [5]. These four different architectures (Figure 2.1) were based on AlexNet [45] architecture but differed on how the input was fed and preprocessed throughout the network. The neural network architectures have been implemented as follows:

1. *Concatenated Channel model (CC)*: Pre-disaster and post-disaster images are concatenated into a single 6-channels image and fed into a unique convolutional encoder.

2. *Post-image Only model (PO)*. A 3-channels post-disaster image is used as input and fed into a single convolutional encoder.

3. *Twin-Tower Concatenate model (TTC)*. Pre-disaster and post-disaster images are preprocessed using separate convolutional encoders, then concatenated to extract features along the channel dimension.

4. *Twin-Tower Subtract model (TTS)*. Same as TTC, except that the extracted pre-disaster and post-disaster feature values, after the convolutional encoders, are substracted element-wise instead of concatenated.

Results showed that twin-tower models outperformed single tower models using 5-cross-validation. The TTS model achieved the best performance with a 0.8302 ± 0.0056 AUC score, while the TTC model achieved the second-best performance with a 0.8120 ± 0.0054 AUC score.

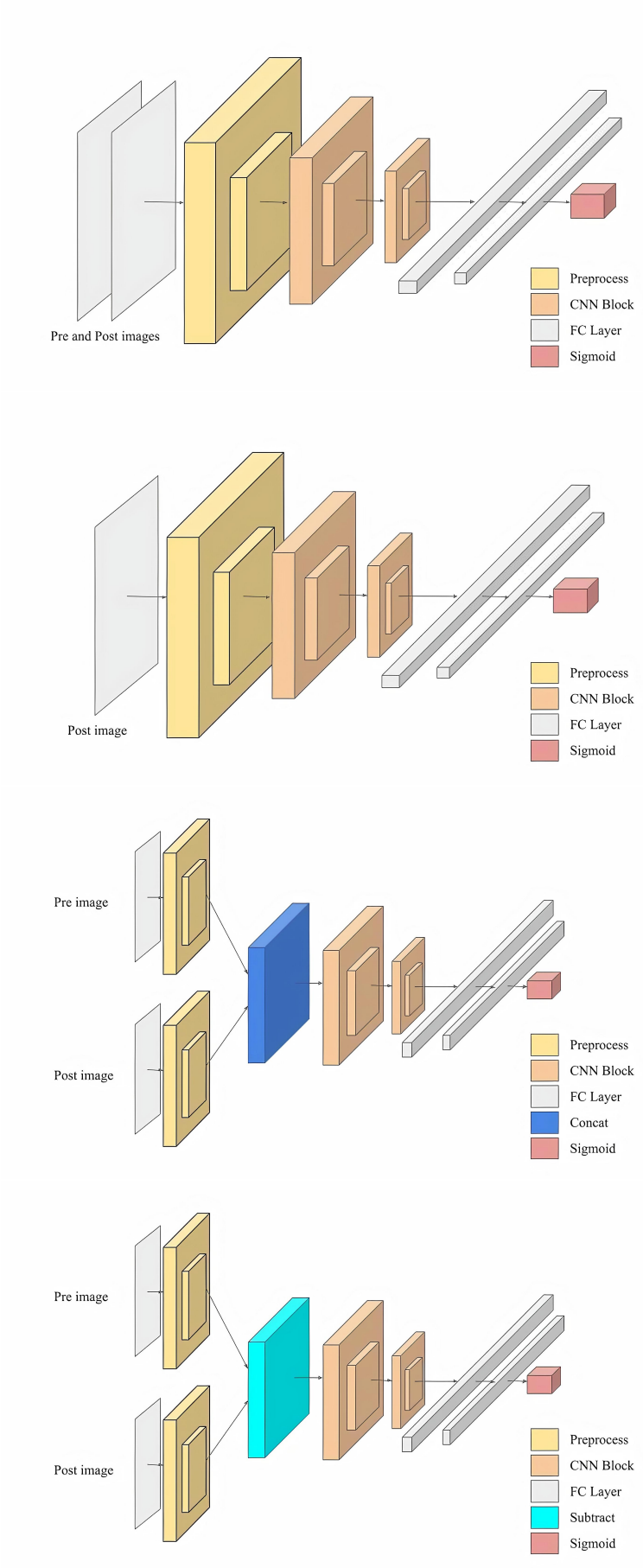

Figure 2.1: (a) *CC model* (b) *PO model* (c) *TTC model* (d) *TTS model*. [5]

Furthermore, the PO single-tower model outperformed the CC one, achieving a 0.8030 ± 0.0064 AUC score and 0.8008 ± 0.0033, respectively. The CC model was the least performant. This suggests that *a)* a simple concatenation of pre-disaster and post-disaster images is insufficient if we do not first extract high-level features from the images, and *b)* twin-tower models with paired pre-disaster and post-disaster imagery are the most suitable for such a task.

Results in line with Google Researchers findings were measured by B. Kalantar et al [30] in 2020, who assessed building damage after the 2016 earthquake in Kumamoto, Japan. The study compared the performances of three CNN models: one single-tower model, where pre-disaster and post-disaster images are treated as a single 6-channel image input, and two twin-tower models, where pre-disaster and post-disaster images are first fed into two different convolutional encoders and then concatenated throughout the network . Results showed again that twin-tower models outperformed single-tower model when paired pre-disaster and post-disaster imagery is used as input. Twin-Tower models achieved an overall accuracy of 76.85% and a F1-score of 0.761.

2.1.1 Generalizability in BDA models

For a damage detection model, to be practically useful, it must perform well in future disasters and generalize well to disasters it has not been trained on. The most straightforward technique used to enhance generalizability is to increase the training set. In this way, we may have more buildings characteristics, disaster types, camera angles, and weather conditions to test our model on, increasing data variety and improving overall performances.

In 2019, Google Researchers investigated the performances of a TTC Model (Figure 2.1 (c)) when trained and tested with different datasets [5]. Results showed that as the training set guaranteed a wide variety of disasters and locations, performances increased when tested on a brand new dataset. Performances maximize when the model was pre-trained with a subset of the target disaster dataset. Table 2.1 shows the tests run by Google Researchers using three different datasets from Haiti 2010 earthquake, Mexico City 2017 earthquake, and Indonesia 2018 earthquake.

Nonetheless, generalizability within BDA is still a challenge as there is only a small number of past disasters for which high-resolution images and manual damage assessment are available. The challenge becomes even more complicated when one wants to develop a model capable of estimating

Train Dataset	Test Dataset	AUC	Accuracy
Haiti	Mexico	0.62	0.60
Haiti + Indonesia	Mexico	0.73	0.68
Haiti + Indonesia + 10% of Mexico	90% of Mexico	0.76	0.72
Haiti	Indonesia	0.63	0.60
Haiti + Mexico	Indonesia	0.73	0.67
Haiti + Mexico + 10% of Indonesia	90% of Indonesia	0.80	0.70

Table 2.1: *Results of Google's generalizability experiments [5]. A TTC model has been trained and tested with different dataset combinations.*

the damages of multiple natural disasters. One of the main reasons is that there is a lack of datasets that comprehends a great variety of locations and disaster types with a common building damage scale. The only option to date is the xBD dataset [8], including five different disaster types (earthquakes, tornados, floodings, volcanic eruptions, hurricanes) from four geographic regions (America, Europe, Asia, Oceania). Since its publishment, several papers investigated the performances of single-tower and twin-tower CNN architectures for the xBD dataset using paired pre-disaster and post-disaster imagery [6] [7] [19][46].

E. Weber et al. [19] built a TTC model based on a ResNet-50 backbone with shared weights and scored an F1-score of 0.835 for building segmentation and an F1-score of 0.679 for damage assessment on the xBD validation dataset. On the same track, R. Gupta et al. [46] developed a CC model based on a dilated ResNet with a subsequent Atrous Spatial Pyramid Pooling model as an encoder to extract multi-level features. It achieved an F1-score of 0.84 for building segmentation and an F1-score of 0.74 for damage assessment when tested on a sample of the xBD validation dataset.

As data for BDA is very limited, to further boost generalizability within BDA models, recent literature started to include attention operators in CNN architectures. Attention operators are non-local-operators, opposed to convolutional operators, that compute output units as a weighted sum of all input units, increasing the extraction of global information and hence generalizability performance. The most interesting BDA models that presents attention-operators are the following.

H. Hao et al. [6] proposed Siam-U-Net-Attn model, a multi-class deep neural network with an attention mechanism to assess BDA with paired pre-disaster and post-disaster images. A U-Net architecture is shared by pre-disaster and post-disaster images, which are fed individually. The U-Net module's outputs are two segmentation maps, one for pre-disaster building

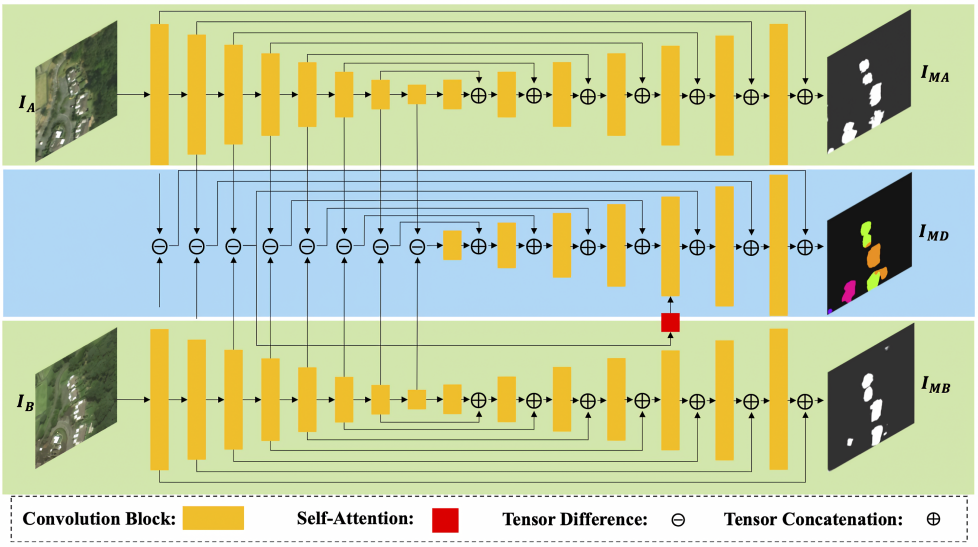

Figure 2.2: *Siam-U-Net-Attn model. I_A and I_B are the pre-disaster and post-disaster input images. I_{MA} and I_{MB} are the corresponding output building segmentation masks. I_{MD} is the output damage scale classification map.[6]*

segmentation (I_{MA}), and one for post-disaster image segmentation (I_{MB}). The difference between the two-stream features produced by U-Net encoder and a new separate decoder, constitute a Siamese network used for damage assessment. The model estimate building damage by extracting features from the difference between pre-disaster and post-disaster feature maps. In order to capture long-range information, an attention module is used within the Siamese network. Results showed an F1-score of 0.73 for building segmentation and an F1-score of 0.7 for damage estimation when trained on 60% of the xBD training set and tested on 20% of the xBD training set. The other 20% of the xBD training set was used as validation set. The structure of the network is showed in Figure 2.2.

Another interesting approach is the one developed by Y. Shen et al. [7] Researchers proposed a CNN module with a Cross-Directional features strategy based on attention operators to better explore the correlation between pre-disaster and post-disaster images. As the previous example, images are fed individually into a shared U-Net architecture with ResNet-50 used as encoder. What is crucial here is that building damage is assessed with a two-step approach. First, the pre-disaster image is fed into the U-Net architecture, and a building segmentation map is extracted. Then weights from the first step are used as a starting point for the second step, which uses post-disaster

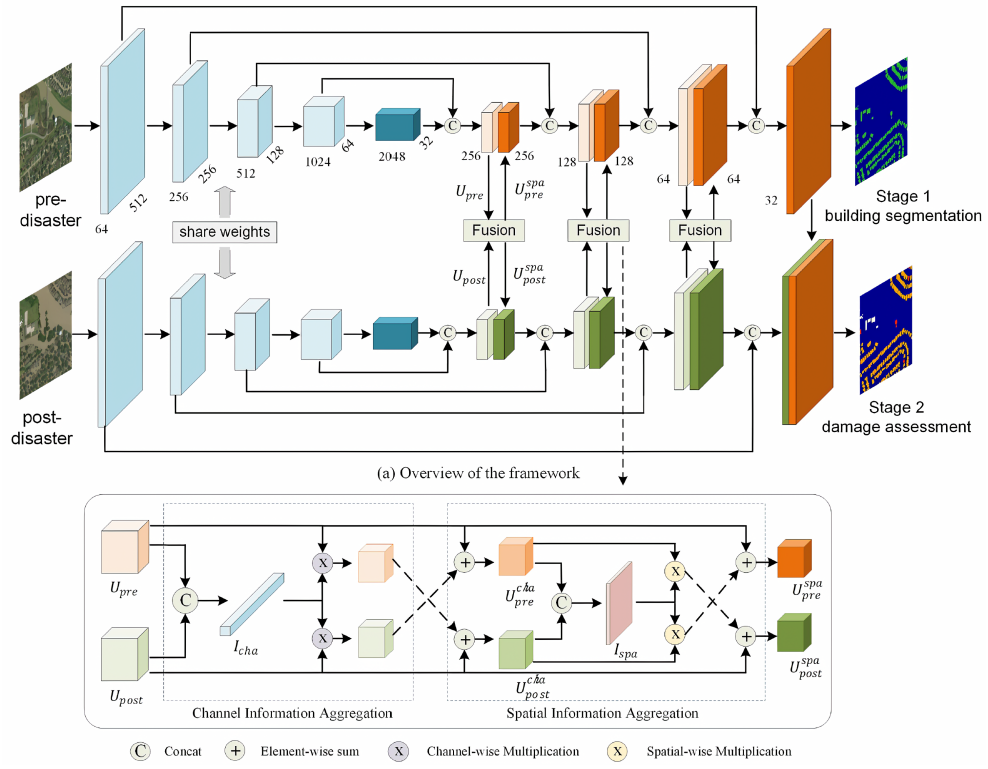

Figure 2.3: (a) *U-Net-like neural network with fusion-attention modules porposed by Y.Shen et al. [7]. In the building segmentation phase, only pre-disaster images and the upper U-Net branch is used. In the damage classification stage, pre-disaster and post-disaster images are fed into the shared U-Net architecture separately.* (b) *Architecture of the cross-directional fusion model.*

images to compute a damage map, using the previously computed building segmentation map as a filter in post-processing. Cross-Directional features modules are added at each convolutional step of the shared decoder to extract more information about the relationship between pre-disaster and post-disaster images. Each module m_i takes as input the two decoder convolutional steps at level i. It uses attention operators to aggregate spatial and channel information in a cross manner and then embeds them into the network. Figure 2.3 shows the structure of the network. With this two-step approach, Y. Shen et al. achieved the state-of-the-art for the xBD dataset, as they scored an F1-score of 0.864 in building segmentation and an F1-score of 0.778 for damage estimation.

Attention operators are indeed the most recent innovation for BDA model, as they are capable to boost generalizability and increase damage classification

performances even when little data augmentation is applied. However, research behind attention operators for BDA is still at the beginning, since very few studies have investigated the performances of state-of-the-art attention modules in BDA models. Moreover, attention modules that have been already tested for BDA (e.g. the ones described before) are specifically tailored to the BDA task and they have never been tested on other segmentation/classification tasks to investigate their versatility. There is the need to test a more versatile attention module that could be used either on another task or another neural network architecture too without loosing quality.

An example of versatile attention modules from recent literature for augmented microscopy are Global Voxel Transformer Operators (GVTOs) [4]. GVTOs combine local and non-local operators and can capture both local and long-range dependencies. In particular, they can be used as a flexible building block in the U-Net architecture, thus potentially applicable in various scenarios. GVTOs have been designed to support size-preserving, down-sampling, and up-sampling tensor processing, covering all kinds of operators in the U-Net framework.

With GVTOs, GVT-Nets [4] have been proposed for Augmented Microscopy as an advanced tool to address the limitation of the convolutional U-Net framework. Studies showed that: *a)* basic GVT-Nets with a single size-preserving GVTO at the bottom level improves upon the U-Net baseline on 13 different datasets for augmented microscopy; *b)* GVT-Nets obtained a more promising Transfer Learning performance than state-of-the-art 3D-to-2D image projection models, indicating a better generalization ability.

Hence, GVT-Nets are very promising, and an investigation of those for BDA could enrich its literature and improve state-of-the-art U-Net architectures.

2.1.2 Handling bias (data imbalance) in BDA models

Besides generalizability, CNN and ML performances are strictly related to the data distribution of the training dataset. When the distribution of examples across the known classes is biased or skewed, we face an unbalance classification/segmentation problem. This is a challenge for ML models, as they are built to assume equal data distribution among classes. If not appropriately addressed, this results in models with poor predictive performances, especially among the minority classes.

xBD and other datasets designed for BDA are examples of imbalanced datasets, where the distribution of damage classes is highly skewed towards

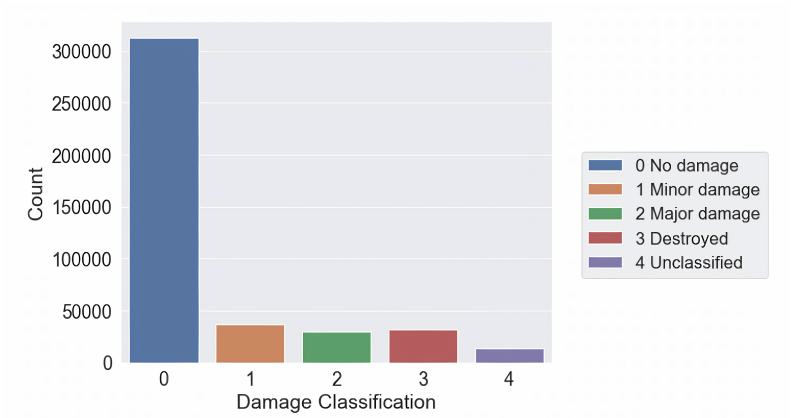

Figure 2.4: *Distribution of damage class labels in the xBD dataset [8].*

"no damage". Specifically, in the xBD dataset, the number of "no damage" buildings is eights times the number of buildings of all other classes (Figure 2.4). Consequently, state-of-the-art solutions for BDA introduce many algorithms to address data imbalance.

As of today, the leading solution adopted by the state-of-the-art is the introduction of class-balancing weights inside the implementation of the loss function, giving higher weights to minority classes and lower weights to majority classes. In this way, wrong predictions of the minority classes will be more influential than good predictions of the majority class in the loss function, making the training process more equally distributed. We can see this approach applied with the Weighted Cross Entropy loss functions by most of the BDA state-of-the-art models [6] [7] [19] [44].

Another approach adopted by the state-of-the-art to address data imbalance during preprocessing time is to either over-sample minority classes or under-sample majority classes. That means to either increase the class distribution of minority classes or decrease the class distribution of majority classes intentionally before training. We can see a combination of random over-sampling and random under-sampling techniques in [44] and a specific over-sampling technique, named CutMix [47], used by Y. Shen et al. [7]. CutMix is a data augmentation technique that generates a new image by combining two image samples. In BDA and imbalanced classification/segmentation tasks in general, it is used to copy-paste hard classes on top of images with a higher density of background pixels.

Although BDA's state-of-the-art has proved the beneficial effect of weighted loss functions and data re-balancing techniques, no real comparison is available that investigates the ones that achieves better building damage

segmentation performances for the BDA task. Moreover, most of BDA's state-of-the-art uses the Weighted Cross Entropy as a loss function, even though other loss functions have already outperformed it in image segmentation (Dice Loss [35], J regularization [37]) and image classification tasks (Focal Loss [34]). A comparison of those could be beneficial for the BDA literature to understand what loss function is the best performing one for building segmentation and damage classification and to what extent they could improve the prediction of hard classes.

2.2 Summary

This chapter summarized the most recent innovative architecture for BDA, addressing generalizability and techniques that improve the handling of imbalanced datasets. Moreover, the literature review highlighted knowledge gaps that are going to be addressed in the following chapters.

Specifically, attention operators have found to be the state-of-the-art for improving BDA generalizability, even though the research is still at its beginning. Current architectures lack versatility and generalizability. They are particularly tailored either for the BDA task or to particular neural network architectures, and have not been tested on other segmentation/classification tasks. GVT-Nets are proposed to improve upon current attention operators as recent studies showed their flexibility within the U-Net architecture and improved upon augmented microscopy state-of-the-art.

Moreover, current BDA imbalanced datasets are addressed with the Weighted Cross Entropy loss function, even though ML literature showed that other loss functions seem more practical for imbalanced segmentation and classification tasks. A comparison of state-of-the-art balancing loss functions is needed to understand which is the best performing one for BDA.

Chapter 3

Method

This chapter describes the materials used throughout our investigation, the neural network architectures adopted for our research, and the techniques used to evaluate the obtained results. The study is mainly divided into four experiments, which consist of: (1) investigating the effects of Transfer Learning for the BDA task and particularly for the xBD dataset; (2) benchmarking state-of-the-art balancing loss functions with the most promising state-of-the-art CNN for BDA ; (3) testing GVTOs with the best performing loss function and compare it with state-of-the-art BDA models; (4) testing neural network flexibility with a brand new imbalance segmentation task. At the end of the chapter implementation details are also given.

3.1 xBD Dataset

The dataset used throughout our investigation is a data split of the xBD dataset [8], published in 2019 by the Defence Innovation Unit (DIU) of the USA. It was used as a benchmark for the 2019 xView2 Challenge [33]. The goal of the challenge was to identify buildings and rate them based on how badly they have been damaged by past natural disasters, using satellite images taken before and after the disaster occurred.

The dataset contains 850'736 annotated buildings and spans 45'365 square kilometers of satellite imagery across 4 different geographical regions (Americas, Europe, Asia, Oceania). It captures 19 natural disasters of 5 different types (volcanic eruptions, earthquakes, floods, wildfires, and tornadoes). For training models, it includes 18'336 pairs of pre-disaster/post-disaster 1024x1024 high-resolution color images (Figure 3.1). Each building of post-disaster images is labeled based on the amount of damage they

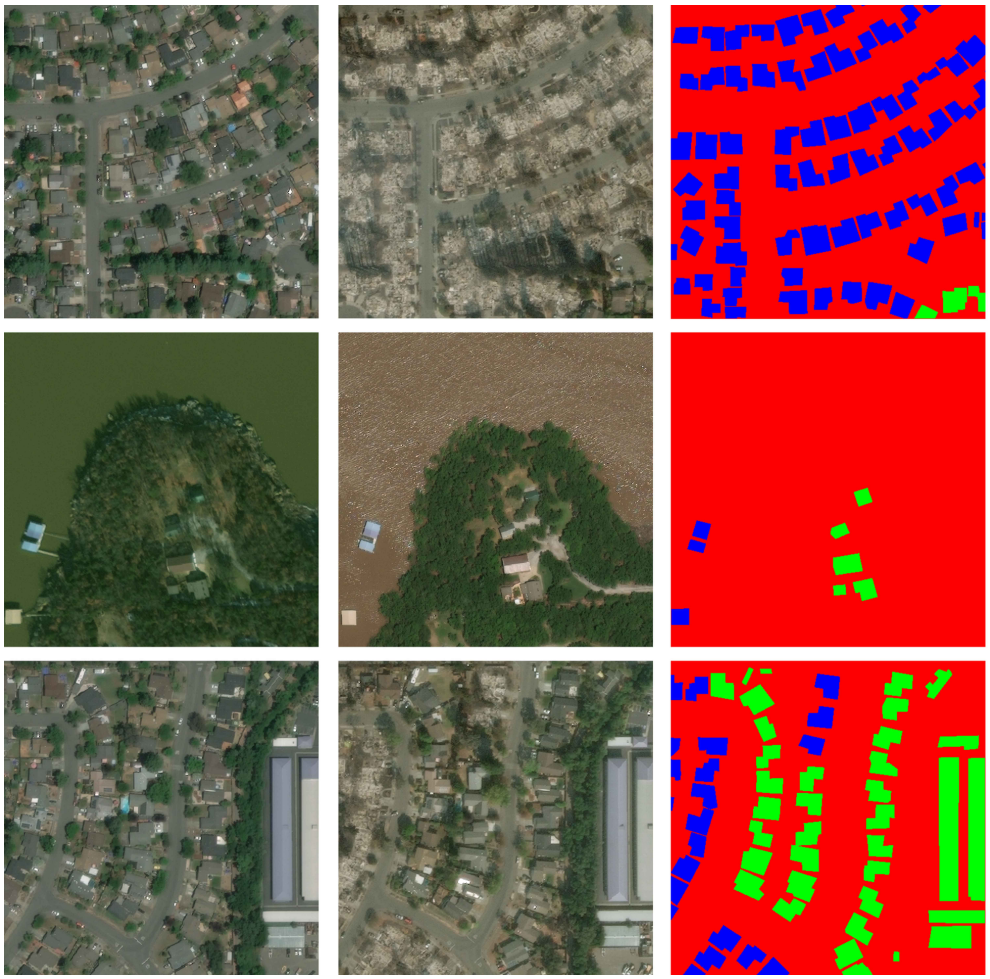

Figure 3.1: *Examples of xBD pre-disaster and post-disaster images (with labels) caused by a wildfire (first and last image sequences) and a flood (second image sequence). Labels are defined as follows: color red represents the background, color green represents undamaged buildings, and color blue represents damaged buildings*

sustained from a given natural disaster, ranging from "no damage" to "destroyed" (Table 3.1).

As we see from Figure 2.4, the dataset is highly biased towards the "no damage class" and the number of buildings affected by each disaster varies from less than 200 to more than 100'000. Moreover, data is sparse since building pixels are only 5.5% of the total dataset pixels. Undamaged buildings represent 73% of total buildings and makeup 4% of the dataset's total pixels. On the other hand, damaged buildings (regardless of the type of damage)

Disaster Level	Structure Description
0 (*No damage*)	Undisturbed. No sign of water, structural or shingle damage, or burn mark.
1 (*Minor damage*)	Building partially burnt, water surrounding structure, volcanic flow nearby, roof elements missing, or visible cracks.
2 (*Major damage*)	Partial wall or roof collapse, encroaching volcanic flow, or surrounded by water, mud.
3 (*Destroyed*)	Scorched, completely collapsed, partially/completely covered with water/mud, or otherwise no longer present

Table 3.1: *Joint Damage Scale descriptions on a four-level granularity scheme*

represent 27% of total buildings and makeup 1.5% of the dataset's total pixels. Data imbalance and data sparcity is thus a big challenge when the dataset is used as training set.

Structurally, the dataset is divided into four main folders. There are two training folders (Tier1 and Tier3), one validation folder, and one testing folder. Originally, the dataset was published with the training and testing datasets only, as the validation dataset was adopted by the jury to evaluate the solutions proposed by the participants of the xView2 Challenge. However, even though the xBD test dataset was available since the beginning of the challenge, many state-of-the-art papers tested on different data splits, leading the comparison between state-of-the-art models to be inconsistent. The major evaluation data splits used by the state of the art are either the xBD test dataset, or a split (usually from 10% to 20%) of the train dataset. We can see that, if a researcher want to conduct an exhaustive research, he would have to evaluate both on a subset of the train dataset and on the xBD test dataset, which is redundant and time-inefficient. For this reason, and because there is not an official benchmark for the xBD dataset, we would like to delineate a new evaluation standard to facilitate future research: the training set may be either one between Tier1 and Tier3 folder, or both of them; the validation data may be either a data split of the training dataset or the xBD validation folder; the testing data must be covered by the xBD testing dataset. If such a schema is followed by future literature, researchers will find the comparison between state-of-the-art models much more accessible and easy to understand. The new benchmark can be found on PapersWithCode's website [48]. Moreover, it is very important that the dataset or data split used is clearly stated for each experiment that is going to test, as follows.

Experiment	Train set	Validation set	Test set
(1) *Investigating Transfer Learning effects in BDA*	90% Tier1-slice	10% Tier1-slice	Test-slice
(2) *Loss Function Benchmark*	90% Tier1-slice	10% Tier1-slice	Test-slice
(3) *GVT-Net Ablation study*	90% Tier1-slice	10% Tier1-slice	Test-slice
(4) *Comparison of GVT-Net with state-of-the-art BDA models*	90% Tier1	10% Tier1	Test

Table 3.2: *Data settings for each experiment*

3.1.1 Data Split used for each experiment

In our case, we used different xBD dataset splits depending on the experiments that we wanted to run. As we had limited time and computational power, we decided to run intermediate experiments with a subset of the Tier1 folder and a subset of the xBD test dataset. We named the subsets Tier1-slice and Test-slice, where image pairs were sampled from each disaster equally. Tier1-slice has 396 image pairs in total, while Test-slide has 44 of them. For the most important experiments, we trained our models on the entire Tier1 folder and tested them on the xBD test dataset. Table 3.2 shows data splits used throughout each experiment. In our research, to minimize the difficulty of the task, we grouped all damaged buildings from "minor damage" to "destroyed" as a single "damaged" class. The splitting of the latter into the three different damage types would be one of the future works of this research.

3.2 CNN baseline: BDNet

We decided to implement our baseline neural network based on the state-of-the-art baseline neural network that achieves the overall best F1-score performance in the building damage segmentation task , which is the double-branch CNN proposed by Y. Shen [7]. We also followed their two-steps protocol discussed in Section 2. The first step is for building segmentation maps, where only pre-disaster images and the first branch are used. The second step is for damage classification, where paired pre-disaster and post-disaster images are adopted as well as both branches. We named the network BDNet. Figure 3.2 shows an overview of the neural network architecture, and Table

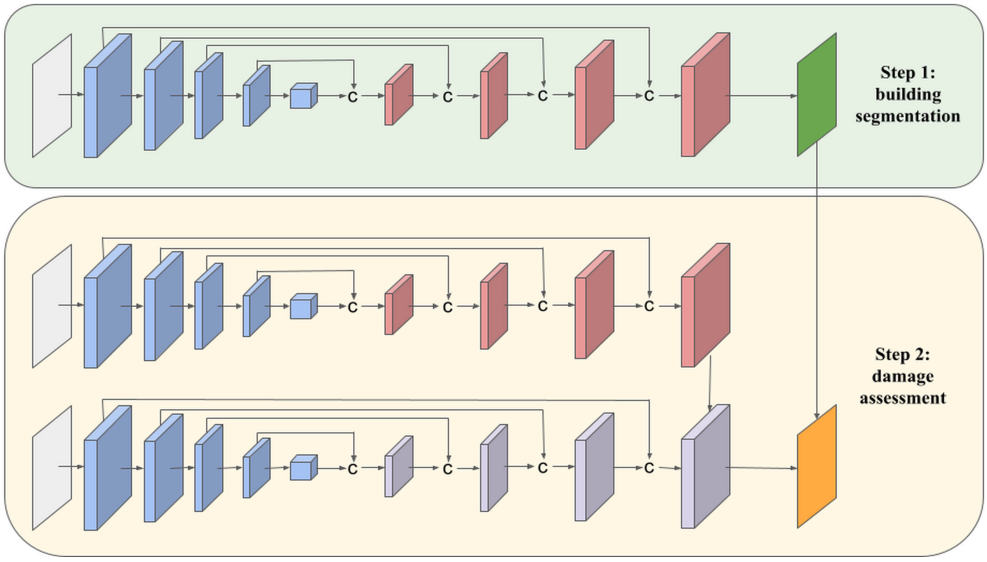

Figure 3.2: *BDNet architecture. The encoder (blue convolutions) has identical implementation for each step and each branch. The red decoder is the one used for pre-disaster images while the grey encoder is used for post-disaster images. The output of the 1st step is used as a mask to optimize the output of the 2nd step. The parameter used at each convolutional level are stated in Table 3.3*

.

3.3 shows the parameters of BDNet.

Each branch of BDNet is composed of an encoder and a decoder, which contain five and four convolutional blocks respectively. Each convolutional block is divided into a convolutional layer, a batch-normalization layer, and a ReLU activation layer. A pooling/upsampling layer is added to either down-sample or up-sample the feature maps for an/a encoder/decoder block. Output layers are similar to a standard convolutional block. However, in this case they equip a different activation function in the last layer. For step 1, the sigmoid activation function is used to produce the building segmentation map. For step 2, the softmax activation function is used to produce the damage segmentation map. Moreover, in step 2, before the output layer, pre-disaster features and post-disaster features are concatenated, and then and fed as input into the output layer.

As BDNet is a U-Net-like CNN, skip connections are added, so that features from the encoder and decoder are integrated to enhance the learning ability of the network. Moreover, as Figure 3.2 shows, the building segmentation map from Step 1 is applied as a binary mask over the damage segmentation map from Step 2 to form the final output. The building

Step 1			Step 2				
Layer	Feature Size	Kernel Size	Layer	Feature Size	Layer	Feature Size	Kernel Size
input-pre	512x512x3	-	input-pre	512x512x3	input-post	512x512x3	-
convb1	256x256x64	5x5	convb6	256x256x64	convb11	256x256x64	5x5
convb2	128x128x256	3x3	convb7	128x128x256	convb12	128x128x256	3x3
convb3	64x64x512	3x3	convb8	64x64x512	convb13	64x64x512	3x3
convb4	32x32x1024	3x3	convb9	32x32x1024	convb14	32x32x1024	3x3
convb5	16x16x2048	3x3	convb10	16x16x2048	convb15	16x16x2048	3x3
dconvb1	32x32x512	3x3	dconvb5	32x32x512	dconvb9	32x32x512	3x3
dconvb2	64x64x512	3x3	dconvb6	64x64x512	dconvb10	64x64x512	3x3
dconvb3	128x128x96	3x3	dconvb7	128x128x96	dconvb11	128x128x96	3x3
dconvb4	256x256x32	3x3	dconvb8	256x256x32	dconvb12	256x256x32	3x3
Output	512x512x1	-	-	-	Ouput	512x512x3	-

Table 3.3: *Parameters of BDNet. The 1st branch is used in the 1st step to estimate building segmentation and it is maintained in the 2nd step. In the 2nd step both branches are used. The two branches differs only for the output layer*

segmentation threshold from Step 1 is not set to a fixed value. Instead, we calculated the overall binary segmentation F1-score for threshold from 0.01 to 0.99. The threshold with the highest F1-score value was then chosen.

3.2.1 Investigating Transfer Learning in BDA

Researches for Transfer Learning has made one of the greatest achievements within the last decade. As already mentioned in Section 2, most state-of-the-art models adopt a ResNet-50 with ImageNet pre-trained weights to initialize the BDA task. However, even though the benefits of Transfer Learning are consolidated, to the Authors' knowledge there is little work that investigate the effects of Transfer Learning within the building damage segmentation task.

For this reason, we compared the performance of a vanilla BDNet and a BDNet architecture boosted with a ResNet-50 encoder with ImageNet pre-trained weights. In this way, we were able to understand the impact of Transfer Learning for the BDA task. This experiment was one of the two ablation studies conducted throughout our research. Results are shown in Section 4.1.

3.2.2 Loss Function Benchmark

We decided to benchmark BDNet with various balancing loss functions to understand which one achieves better F1-score performance in the building damage segmentation task. Specifically, we wanted to compare the performance of the Weighted Cross Entropy, which is used by most of the state-of-the-art, with other balancing loss function, as we noticed that little

research has been done about. Weighted Cross Entropy is most of the time adopted without motivation and/or not specifying the reason why it is better than other loss functions. With our experiment we want to investigate whether the Weighted Cross Entropy loss function could be outperformed from other balancing loss function, and if so, to what extend. Due to time constraints, we adopted the best performing loss function for this experiment also for the next experiments. The loss functions that have been compared are the following:

1. *Weighted Cross-Entropy Loss*: It is a weighted version of the cross-entropy loss functions, where different weights are given to each predicted class. Weights can be either $1/ft$ with f the frequency of class t, or custom. This loss function has been widely used within the BDA literature within the last two years. It can be defined as:

$$WCE(p_t) = -\alpha_t log(p_t) \tag{3.1}$$

   where p_t are the predicted values for class t and α_t as the weighting parameter.

2. *Focal Loss*: Originally developed by Facebook AI Researchers [34], the focal loss adds a regularizer to the cross-entropy to prevent easy negatives from overwhelming the detector during training. It happens when there is an extreme foreground-background class imbalance (e.g., the one noticed between background pixels and building pixels within the xBD dataset). More formally, we define the Focal Loss as:

$$FL(p_t) = -(1 - p_t)^{\gamma} log(p_t) \tag{3.2}$$

   where p_t is defined as:

$$p_t = \begin{cases} p, & \text{if } y\text{=1} \\ 1 - p, & \text{otherwise} \end{cases} \tag{3.3}$$

   with y as the ground-truth of class t and $p\epsilon[0, 1]$ as the model's estimated probability. in Equation 3.2 $(1-p_t)^{\gamma}$ is a modulating factor to the cross-entropy loss with hyper-parameter γ. We can see that as p_t goes to 1, the regularizer goes to 0 and the loss for well-classified examples is down-weighted. The focusing parameter γ adjusts the rate at which easy examples are down-weighted. In our study, we added the regularizer to the Weighted Cross-Entropy loss function, with α_t as the inverse

frequency of class t:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} log(p_t) \tag{3.4}$$

3. *Dice Loss*: Initially proposed by F. Mitellari et al for cancer segmentation [35], the Dice Loss aims to address the extreme imbalance between the foreground and background class (as the Focal Loss does), but specifically for segmentation tasks. The loss function is based on the dice coefficient which measures the pixel similarity of predicted values and ground truth for each class between foreground pixels and background pixels. More specifically, we can define the Dice Loss as:

$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \tag{3.5}$$

where p_i are the predicted values of class i and g_i is the ground truth of class i. In this way, we estimate the ratio between good predictions and bad predictions specifically for each class. The final loss value is the sum of all class ratios. Note that weights are not strictly needed.

4. *J Regularization Loss*: Originally developed for cell segmentation [37], the J regularization loss function is based on the J statistic, formulated by statistician Willian J. Youden to improve rating the performance of diagnostic tests of diseases [49]. The J statistic gives equal importance to correctly classified samples no matter if they belong to a class or not. That is, it gives the same weights to the true positive ratio (*sensitivity*) and the true negative ratio (*specificity*) for each class. Considering a binary classification problem, we define the J regularization loss function as:

$$L_j(p_t, y_t) = -\lambda \log(\frac{\alpha + \beta}{2}) \tag{3.6}$$

with α and β as soft definitions for the true positive rate and the true negative rate, respectively, and λ as a custom weighting coefficient. The loss function can be converted into a multi-class surrogate by summing all pairwise binary surrogates as follows:

$$L_j(p_t, y_t) = -\sum_{i=0}^{C}\sum_{k=0}^{C} \lambda_{t,k} \log(\frac{\alpha_t + \beta_{t,k}}{2}) \tag{3.7}$$

In Equation 3.7, $\lambda_{t,k}$ is a pairwise class weight. α_t and $\beta_{t,k}$ are

soft definitions of the true positive rate and the true negative rate, respectively, where i represent the positive class and k the negative one. More specifically, α_t and $\beta_{t,k}$ are defined as:

$$\alpha_t = -\sum p_t \frac{y_t}{n_t} \tag{3.8}$$

$$\beta_{t,k} = -\sum (1 - p_t) \frac{y_k}{n_k} \tag{3.9}$$

where n_t and n_k are the numbers of pixels of the positive class t and the negative class k, respectively.

Each loss function has been tested singularly. Specifically, for each loss function, we trained BDNet with a binary variation for the 1st step (building segmentation), and with a multi-class variation for the 2nd step (damage classification). Results are shown in section 4.2.

3.3 Global Voxel Transformer Operators

As we discussed in Section 2, attention operators are nowadays the state-of-the-art to improve the overall performance within BDA models.

However, current BDA state-of-the-art attention operators lack versatility, that is, their architectures prevent them from being adapted to other tasks (or makes it costly in practice). On the other hand, GVTOs proved great flexibility within a U-Net-like architecture, as they can replace down-sampling convolutions, up-sampling convolutions, and size-preserving convolutions. This makes them potentially applicable to every segmentation and classification task where a U-Net-like architecture is exploited. GVT-Nets are U-Net architectures that feature GVTOs instead of common convolutions.

Figure 3.3 shows the structure of a general GVT-Net architecture. We can see that there are three types of GVTOs: Size preserving GVTOs, down-sampling GVTOs, and up-sampling GVTOs.

3.3.1 Size Preserving GVTO

We define a tensor $X \in \mathrm{R}^{c \times d \times h \times w}$ as the input tensor of the size-preserving GVTO, representing c feature maps of spatial size $d \times h \times w$. As the first step, GVTO performs three independent $1 \times 1 \times 1$ convolutions on X, obtaining three different $d \times h \times w \times c$ tensors, namely the Query (Q), Key (K), and Value

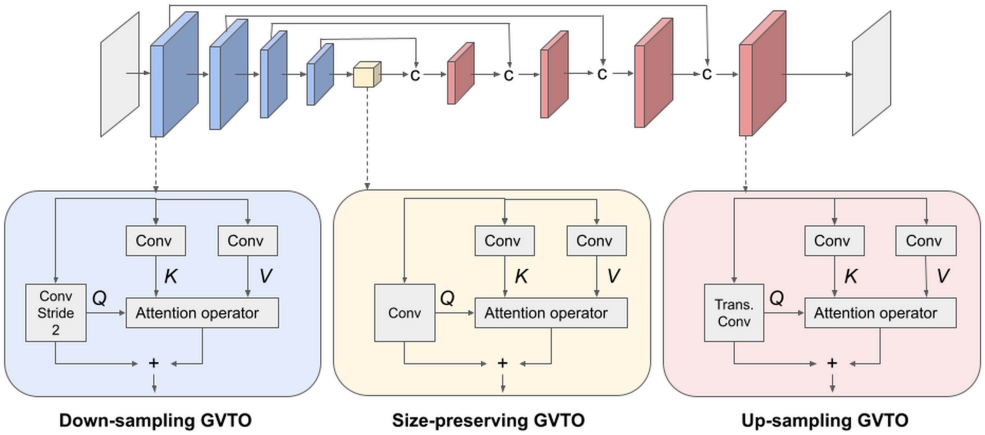

Figure 3.3: *GVT-Net architecture. It is a U-Net-like architecture where down-sampling convolutions are replaced with down-sampling GVTOs. Size-preserving and up-sampling convolutions are replaced too with size-preserving and up-sampling GVTOs. Under the network architecture we can see a detailed visualization of each GVTO block.*

(V). Subsequently, Q, K, and V are unfolded along the channel dimension. Therefore, we obtain three tensors namely Q', K', V' with shape $c \times dhw$. These three matrices are the input of the attention operator defined as:

$$Y = V \cdot Norm(K'^T Q') \in R^{c \times dhw}, \tag{3.10}$$

where the function $Norm(-)$ normalizes each column of $Q'_T K' \in R^{dhw \times dhw}$ with the inverse of the tensor spatial size $1/dhw$. Specifically, the normalization function is defined as follows:

$$Norm(K'^T Q') = \frac{K'^T Q'}{dhv} = \frac{1}{dhv} K'^T Q' \in R^{c \times dhw} \tag{3.11}$$

Afterwards, Y is folded back to a tensor $Y' \in R^{c \times d \times h \times w}$. The output of the size-preserving GVTO is the sum between Q and Y', that is a residual connection between the query tensor Q and the output of the attention operator Y'. Sizes are therefore preserved.

3.3.2 Down-sampling and Up-sampling GVTOs

The difference between down-sampling/up-sampling GVTOs and size-preserving GVTOs are the following.

In the first step of down-sampling GVTOs, we use a $3 \times 3 \times 3$

convolution with stride 2 to obtain the tensor query $QR^{d/2\times h/2\times w/2\times 2c}$, and two independent $1 \times 1 \times 1$ to generate the tensors key $K \in R^{d\times h\times w\times 2c}$ and value $V \in R^{d\times h\times w\times 2c}$. The three tensors are then unfolded along the channel dimensions to obtain $Q' \in R^{2c\times dhw/8}$, $K' \in R^{2c\times dhw}$, and $V' \in R^{2c\times dhw}$. The tensors are then fed into the same attention module, which output the tensor $Y \in R^{2c\times dhw}$, folding it back to a tensor $R^{d/2\times h/2\times w/2\times 2c}$.

As opposed to size-preserving GVTO the output tensor Y' has shape $d/2 \times h/2 \times w/2 \times 2c$, thus performing a down-sampling operation. Up-sampling GVTOs are very similar to down-sampling operator. To obtain Q, K, V transpose convolutions are applied instead of convolutions (all other parameters are maintained). Hence, the three convolutions generate $Q' \in R^{2d\times 2h\times 2w\times c/2}$, $K' \in R^{d\times h\times w\times c/2}$, and $V' \in R^{d\times h\times w\times c/2}$. The attention module and the residual connections are kept the same. The output of a up-sampling GVTO is $Y' \in R^{2d\times 2h\times 2w\times c/2}$, doubling the spatial size but halving the channel dimension.

3.3.3 Positioning GVTOs inside BDNet

Z. Wang [4] et al. states that GVTOs can easily replace convolution operators in order to increase generalizability. Specifically, they demonstrated that a basic GVT-Net improves performances over a basic U-Net architecture in label-free prediction of 3D fluorescence images from transmitted-light microscopy. The basic GVT-Net differs from the U-Net only at the bottom level, where a size-preserving GVTO is applied instead of a convolution. As the first step for applying GVTOs to BDA, inspired by Z. Wang et. al, we also want to investigate whether the replacement of the bottom convolution of BDNet with a size-preserving GVTO is improving performances.

3.3.4 Experimental settings

The first experiment that we run with GVTOs is a comparison between a basic BDNet architecture with the ResNet encoder, and a GVT-BDNet architecture with ResNet encoder and a size-preserving GVTO as the bottom operator. As Table 3.2 shows, the comparison has been done with a subset of the xBD dataset. More specifically, we exploited a 90% of Tier1-slice as training set, and the remaining 10% as validation set. Then we used Test-slice as the test set. Results are shown in Section 4.3.

As the second experiment with GVTOs, we compared GVT-BDNet performances with other state-of-the-art BDA models to have a better overview

of the power of GVTOs compared to current BDA solutions. At this time, we trained and tested on the entire Tier1 folder (90% for train, 10% for validation) and the entire xBD test dataset, respectively. However, as we discuss in Section 3.1, in order to compare consistently GVT-BDNet with the state-of-the-art, we tested it also on 10% of the Tier1 folder. Results are shown in Section 4.4.

3.3.5 Evaluation and Metrics

The experimental results of all trained methods are reported and compared to other models by using three different metrics defined within the xView2 Challenge. By doing so, we can compare our models with state-of-the-art models in a clear manner. The first metric is the F1-score ($F1_b$) for building segmentation. The second metric is the harmonic mean of class-wise damage classification F1 ($F1_d$), which defines the model's overall performance for damage assessment. The third and final metric describes the model's overall performance for both tasks ($F1_o$). Specifically, metrics are defined as follows:

$$F1_b = \frac{2TP}{2TP + FP + FN} \tag{3.12}$$

$$F1_d = \frac{n}{\sum_{i=1}^{n} 1/F1_{C_i}} \tag{3.13}$$

$$F1_o = 0.3 \times F1_b + 0.7 \times F1_d \tag{3.14}$$

In Equation 3.12, TP, FP, and FN are the number of true postive, false positive, and false negative of building segmentation results, respectively,. In Equation 3.13, $F1_{C_i}$ denotes the F1-score of each damage level for damage assessment and has a definition similar to $F1_b$. C_i denotes the damage level. By using $F1_d$ as the damage assessment metric we can compare models that have a different division of the damage classes, as we take their harmonic mean. In Equation 3.14, the model's overall performance is largely influenced by its performance on the damage assessment task.

3.4 Testing GVTNet on other tasks with highly skewed datasets

The effectiveness of neural network architectures can also be described on how well they perform for similar tasks with different scenarios. An excellent example of a good segmentation model is the U-Net architecture, which proved

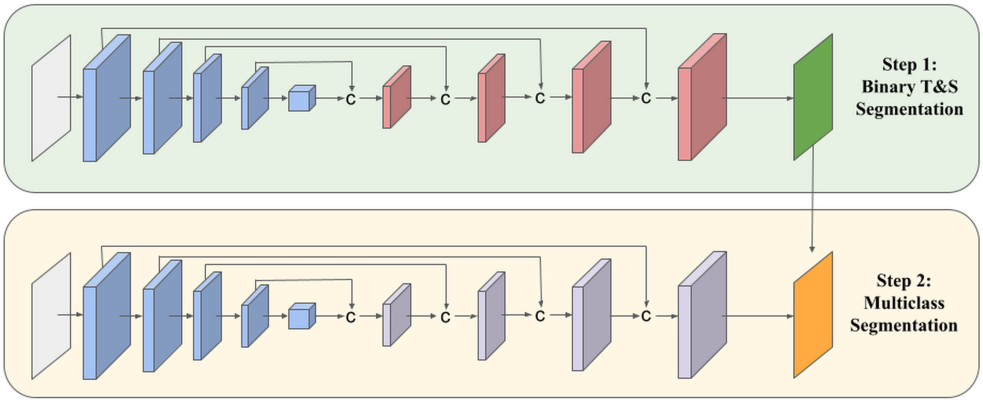

Figure 3.4: *TSNet architecture. Both steps explot the same architecture. They differ only for the output layer. In the 1st step a 1 channel convolution and the sigmoid activation function is used. In the 2nd step a 3 channel convolution and a softmax activation function is used.*

to outperform state-of-the-art performances for cell segmentation, cancer segmentation, and many other segmentation problems.

In our study, we want to investigate whether GVT-BDNet does perform well on other multi-class segmentation problems with highly skewed datasets. More Specifically, we decided to test GVT-BDNet performances with Tree and Shadow segmentation (T&S).

For this task we used the Spacept Dataset as the training set (90% as proper training set, 10% as validation data) and the testing set. The dataset contains more than 11000 thousands 1024x1024 satellite images of cities and countrisides divided into 8 subfolders, which categorized by the country and the time that the satellite images have been retrieved. Images are labelled at pixel-level. There are three different classes: trees, shadows, and background. Moreover, the Spacept dataset is highly biased too, even if at a lower scale compared to the xBD dataset. Specifically, in the Spacept Dataset there are 95% of background pixels, 3.7% of tree pixels, and 1.3% of tree shadows pixels. Figure 3.5 shows an example of satellite imagery from the Spacept Dataset.

3.4.1 GVTNet architecture for T&S segmentation

The architecture of GVT-BDNet used for BDA is strictly related to change detection, particularly the one used in the 2nd and final step. That is, the neural network leverage pre-disaster and post-disaster image features to estimate a

building damage map, as we want to predict the damage that happened between the two temporal stages. In T&S segmentation, we do not have paired images as input, as the task is not related to change detection. Instead, we estimate whether each pixel is either a tree, a shadow, or the background, thus having a more general multi-class segmentation problem.

When translating the GVT-BDNet architecture to T&S segmentation, one could merge the two-steps training into a single multi-class segmentation training step with a single CNN branch. Nevertheless, we wanted to maintain the GVT-BDNet structure and training process as similar as possible to the one used for BDA. Therefore, we decided to keep the two-steps training and define the two steps as follow:

- Step 1: Identical to GVT-BDNet 1st step (see Table 3.3). In this case the output is a binary T&S segmentation map, where trees and shadows are defined as a single positive class, and the background is defined as the negative class.
- Step 2: Similar to GVT-BDNet 2nd step. The difference is that we are no-longer using a double-branch CNN, as we do not need a comparison between two images. Instead, in the T&S 2nd step, we modify the same neural network architecture used in the 1st step by replacing the output layer with a multi-class segmentation layer. As for BDA, 2nd step initial weights are loaded from 1st step weights.

Figure 3.4 shows the architecture used for T&S segmentation. We named the network Tree Shadow Neural Network (TSNet).

To obtain the final output, we apply the 1st step T&S segmentation map as a binary mask to the 2nd step multi-class segmentation mask. The threshold value used for the 1st step T&S segmentation map is chosen from the best-performing threshold value from 0.01 to 0.99, similarly to BDA. Furthermore, to be time-efficient, we did not benchmark the four balancing loss functions described previously for the T&S task. Instead, given the similarity between the two tasks, we decided to only train TSNet with the loss function that would have achieved the best performance within the BDA task. Similar to BDA, the T&S experiment results are evaluated by using the $F1_t$ and $F1_s$ metrics, which are defined as Equation 3.12.

3.5 Implementation Details

We implemented all network architectures using Tensorflow and Keras. Final experiments with the entire dataset have been conducted with AWS (Amazon Web Services) Notebooks with four Tesla A100 GPUs. Intermediate experiments have been conducted with dataset-slices via Google Cloud and Google Colab, with a single Tesla P100 GPU.

1024x1024 Images were cropped into four 512x512 images for training and testing. We applied only basic data augmentations such as flip, rotation,

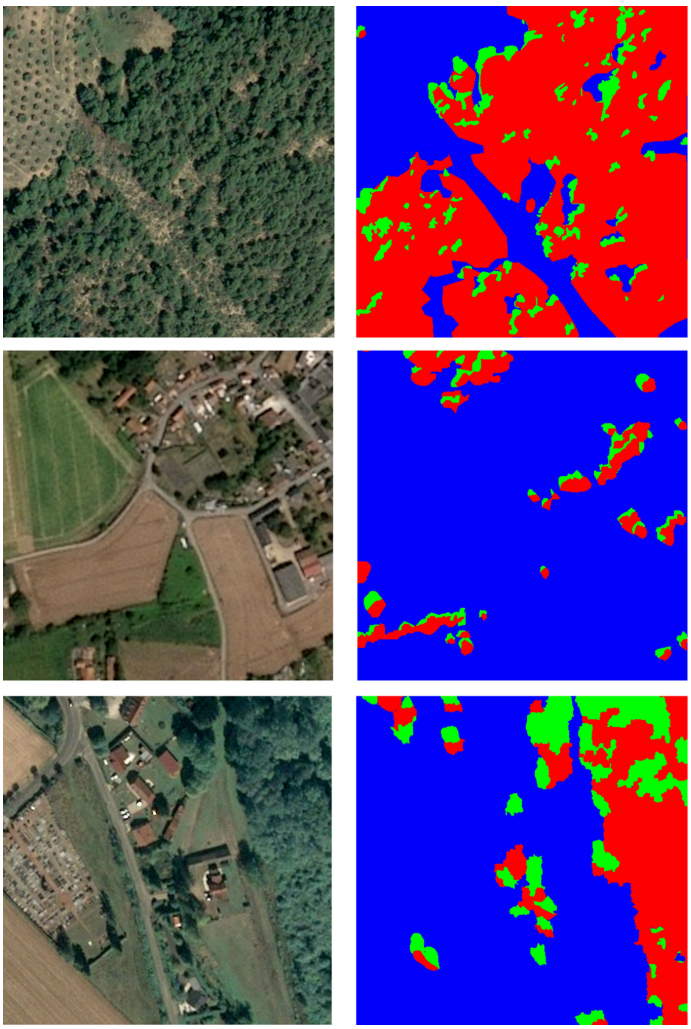

Figure 3.5: *Visualization of the Spacept Dataset. Image pixels are labelled into three different classes: color blue represents the background, color red represents trees, and color green represents tree shadows*

and random brightness increase/decreased. We used different loss functions depending on the experiment. Moreover, we used ResNet-50 with pre-trained weights loaded from the Keras library as the backbone for all experiments except the first one. The number of convolutional channels is equal between each 1st step (BDNet, GVT-BDNet, TSNet, GVT-TSNet). The 2nd step of BDNet, TSNet, and GVT-BDNet has equal convolutional parameters, while the 2nd step of GVT-TSNet has the same parameter as the 1st step. The last convolutional block has 1 convolutional channel during the 1st steps and 4 channels during the 2nd steps.

The optimization method is Adam [50]. In the 1st step of BDA and T&S segmentation, the learning rate used is 0.00015 and the initial chosen number of epochs was 120 (our assumption is based on the training parameters chosen by Y.Shen at al. [7]). During the final training, however, the 1st step converged after 90 epochs for BDA, and 65 epochs for T&S segmentation. On the other hand, in the 2nd step of BDA and T&S segmentation, the learning rate used is 0.0002 and the initial chosen number of epochs was 25. In this case, during the final training, BDA converged after 24 epochs, and T&S segmentation converged after 15 epochs.

Chapter 4

Results and Analysis

This section reports all the results obtained by the experiments described in Section 3 and divides them into four main subsections. In Subsection 4.1, we analyze how Transfer Learning affects the performance of our baseline neural network (BDNet) on the xBD dataset. In Subsection 4.2, we compare different balancing loss functions to identify the most suitable one for BDNet and the xBD dataset. In Subsection 4.3, we discuss the performance of a size-preserving GVTO applied to the BDNet and compare the results with the state-of-the-art BDA neural networks. In Section 4.4, we analyze the performance of TSNet, a neural network based on the BDNet for T&S segmentation.

4.1 Experiment 1: Transfer Learning

As the first step of our analysis, we investigated the performance gain that Transfer Learning can boost for glsBDA. Particularly, we compared two neural network architectures using the same training set (90% of Tier1-slice) and testing set (10% of Tier1-slice) but with different encoder initialization strategies. The first neural network architecture, named vanilla BDNet, is a simple BDNet architecture with model parameters described in Table 3.3, and is initiated with random weights. The second neural network architecture, named BDNet (with ResNet), is also a simple BDNet architecture but using a ResNet-50 module with convolutional blocks [64, 256, 512, 1024, 2048]. Moreover, it starts with ImageNet pre-trained weights. Each neural network has been trained in two steps, one focusing on building segmentation and the other focusing on damage classification, as described in Section 3.

As we can see from the results obtained (Table 4.1), Transfer Learning improves overall performance using BDNet for BDA as expected. Specifically,

Model	$F1_b$	*Und. Build.*	*Dam. Build.*	$F1_d$	$F1_o$
Vanilla BDNet	0.82	0.73	0.32	0.44	0.55
BDNet (with ResNet)	**0.85**	**0.79**	**0.58**	**0.67**	**0.73**

Table 4.1: *Quantitative results of Experiment 1. BDNet with a pre-trained ResNet-50 module used as encoder outperformed the vanilla BDNet architecture. With the best results in Bold*

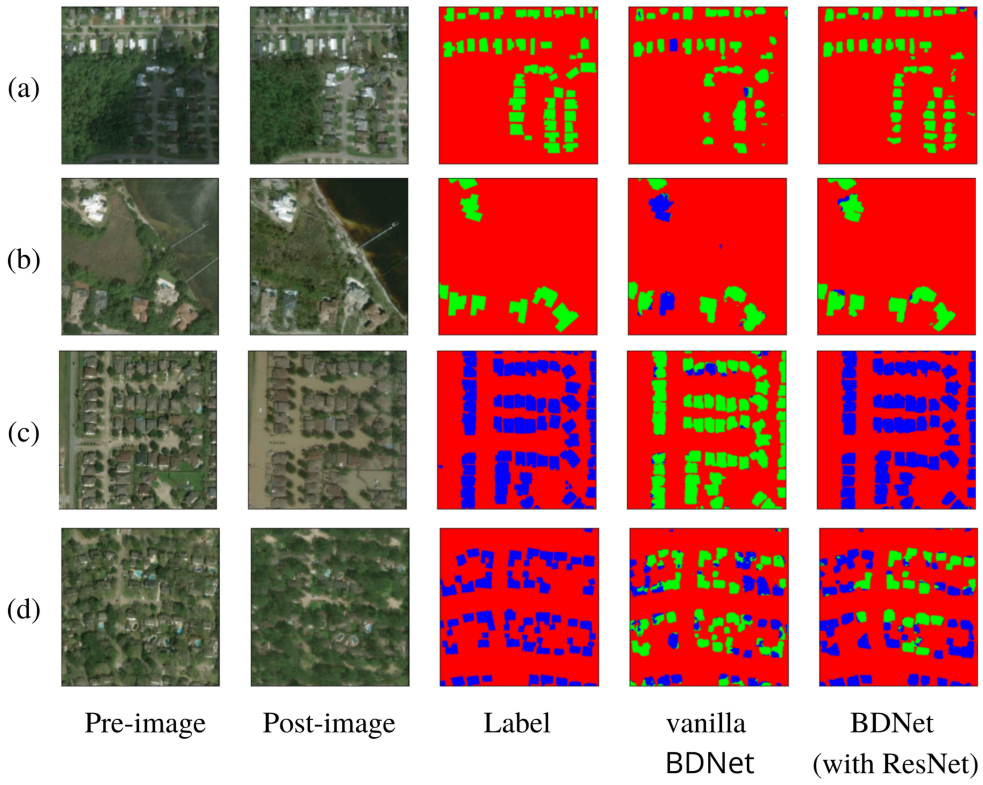

Figure 4.1: *Comparison between vanilla BDNet predictions and BDNet (with ResNet) predictions. Pre-disaster and post-disaster images are visualized alongside their respective label and the predictions of the two models.*

the vanilla BDNet module achieved an F1-score of 0.82 in building segmentation while BDNet-ResNet achieved an F1-score of 0.85, where there is an improvement of 0.03 for the binary segmentation task of step 1. Moving to step 2 (damage assessment), we can see that the BDNet experienced a boost in performance for undamaged buildings and damaged building segmentation when using the pre-trained ResNet-50 module. Particularly, we observe an improvement of 0.06 in terms of the F1-score metric for undamaged buildings

segmentation, as well as an improvement of 0.26 F1-score for damaged buildings. These results highlight the effectiveness of Transfer Learning to improve overall performance including the generalization performance of a multi-class segmentation model (BDA in this case), especially for hard classes (e.g. undamaged and damaged buildings).

Such claims are also supported by qualitative results. As Figure 4.1 shows, we can see that buildings were better segmented from the BDNet with a ResNet backbone (Figure 4.1(a)) and damaged buildings were also more accurately detected. A clear example is the case of damage caused by floods, which were missed by the Vanilla BDNet, but have been detected by the BDNet with the pre-trained ResNet encoder (Figure 4.1(c)-(d)). Overall, both quantitative and qualitative results showed a clear improvement in the prediction of damaged buildings by BDNet when Transfer Learning were applied.

4.2 Experiment 2: Loss Function Benchmark

Results showed that Transfer Learning is needed for better generalizability. However, to further increase the performance on a highly skewed dataset like the xBD dataset, one needs to assess the class imbalance of the dataset. Therefore, we decided to benchmark some of the most promising balancing loss functions referred by recent BDA (Weighted Cross Entropy) and Computer Vision literature in general (Dice Loss, Focal Loss, J Regularization). The goal here is to compare Weighted Cross Entropy performace, which is the loss function most adopted by the BDA literature. A banchmark is also needed as the Weighted Cross Entropy loss function is most of the time adopted without motivation and/or not specifying the reason why it is better than other loss functions.

For this experiment, we used a slice of Tier1 and a slice of the xBD test dataset for training and testing, respectively. The adopted network is the BDNet architecture with a pre-trained ResNet-50 encoder. Quantitative results are showed in Table 4.2. Each column represents the chosen model trained with either the Weighted Cross Entropy (WCE), the Focal Loss (FL), the Dice Loss (DL), and the J regularization technique (JL). From the results, we can see that JL achieved the highest F1-score for building segmentation with an F1-score of 0.87, followed by the WCE Loss (0.86 F1-score), the Dice Loss (0.85 F1-score), and the Focal Loss (0.84 F1-score). Overall, in the 1st step performance were quite similar, which demonstrate that the WCE loss

Model	$F1_b$	*Und. Build.*	*Dam. Build.*	$F1_d$	$F1_o$
BDNet (with ResNet) + WCE	0.86	0.78	0.53	0.63	0.70
BDNet (with ResNet) + FL	0.84	0.75	0.57	0.65	0.71
BDNet (with ResNet) + DL	0.85	**0.79**	0.58	0.67	0.73
BDNet (with ResNet) + JL	**0.87**	0.78	**0.6**	**0.68**	**0.74**

Table 4.2: *Comparison of BDNet performance when trained with different loss functions. They are Weighted Cross-Entropy (WCE), Focal Loss (FL), Dice Loss (DL), and J regularization (JL). Best results are visualized in Bold*

function has generally equal performance to other balancing loss functions for building segmentation. However, we cannot say the same for the 2nd step. From the results, we can see that the best two loss functions that achieved the highest F1-score for damage assessment were the DL (0.67 F1-score) and the JL (0.68 F1-score), while slightly lower performance were achieved by the FL (0.65 F1-score) and the WCE (0.63 F1-score). Here we can see that loss functions that are not specifically designed for segmentation tasks (WCE, FL) are outperformed by loss functions specifically tailored to the tasks (JL, DL). Overall, when performance of both steps were taken into account, JL was the best loss function with an overall F1-score of 0.74, followed by DL (0.73 overall F1-score), FL (0.71 overall F1-score), and WCE (0.7 overall F1-score).

As we expected, loss functions built for imbalance segmentation tasks (as mentioned in Section 3), outperformed the WCE loss function in BDA. Therefore, we can state that both DL and JL should be favored to adoption of the BDNet network when one is dealing with such a task. Due to time constraints, we were not able to test the next experiments with both JL and DL, thus we tested on DL only. However, we mention that the next experiments should be tested on JL as future works, as they achieved the best overall performance when tested on a slice of the xBD dataset.

4.3 Experiment 3: GVTOs

To reach the state-of-the-art performance, we added GVTOs to the BDNet architecture. Particularly, we replaced the topmost convolutional layer of the encoder (2048 feature channel dimension) with a size-preserving GVTO, to boost generalization and therefore the segmentation of hard classes. From literature review, we already know that GVTOs and a singular size-preserving GVTO in particular, have already been tested for augmented microscopy tasks and outperformed the state-of-the-art U-Net architecture [4].

Model	$F1_b$	*Und. Build.*	*Dam. Build.*	$F1_d$	$F1_o$
BDNet (with ResNet)	0.85	0.79	0.58	0.67	0.73
GVT-BDNet (with ResNet)	**0.86**	**0.81**	**0.69**	**0.75**	**0.78**

Table 4.3: *Results obtained from the comparison between GVT-BDNet and BDNet. We can see that the size-preserving GVTO located at the bottom of the encoder improves the overall performance of GVT-BDNet*

Therefore, as the first step, we compared the proposed network architecture, named GVT-BDNet with the BDNet architecture adopted from the previous experiments. As we mentioned in the previous subsection, we used the Dice Loss as the primary loss function. Particularly, we used a Binary Dice Loss function for step 1, as building segmentation is a binary segmentation task, and a Multi-Class Dice Loss function for step 2, as building damage segmentation is a multi-class segmentation task. Moreover, we trained on Tier1-slice (90% for training, 10% for validation), and tested on Test-slice.

Results showed (Table 4.3) that GVT-BDNet achieved an F1-score of 0.86 in building segmentation and an F1-score of 0.78 in damage assessment, while BDNet achieved lower results in both tasks (0.85 of F1-score in building segmentation and 0.69 in damage assessment). We notice there is significant improvement in damaged building segmentation. The F1-score increased from a value of 0.58 to a value of 0.69, improving by a factor of 0.11. Overall, the GVT-BDNet architecture outperformed the basic BDNet architecture by a factor of 0.05, achieving an overall F1-score $F1_o$ of 0.78.

Figure 4.2 and Figure 4.3 show the qualitative results of the comparison between GVT-BDNet and BDNet. Particularly, from Figure 4.2 we can see that GVT-BDNet has better segmentation of undamaged buildings with more solid border. Moreover, from Figure 4.3 we can see that GVT-BDNet improves also the segmentation for both undamaged and damaged buildings.

Overall, both quantitative and qualitative results of GVTO-based models yield higher performance on the classification of hard classes as well as an improvement in building segmentation.

This experiment showed the power of a single size-preserving GVTO, which has consistently improved the performance of a basic U-Net-like state-of-the-art neural network architecture for BDA. We hypothesized that it is due to the ability of GVTOs to aggregate global information, as opposed to local operators like convolutions. With the shared size-preserving GVTO at the bottom of the ResNet-50 module, more information can be exploited to improve features representation of pre-disaster and post-disaster images,

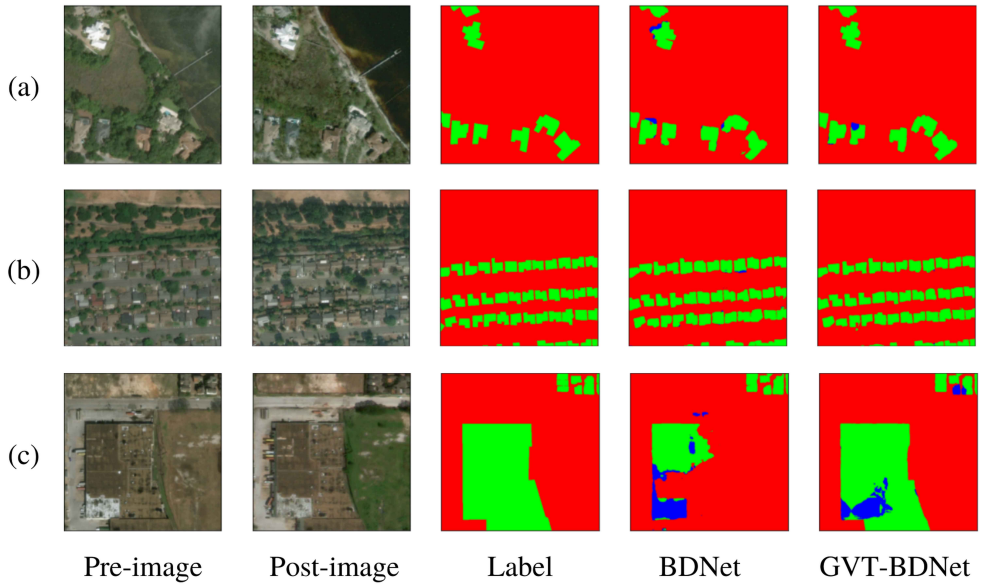

Figure 4.2: *Comparison between BDNet and GVT-BDNet predictions. In these examples we can clearly see an improvement in building segmentation for the GVT-BDNet model.*

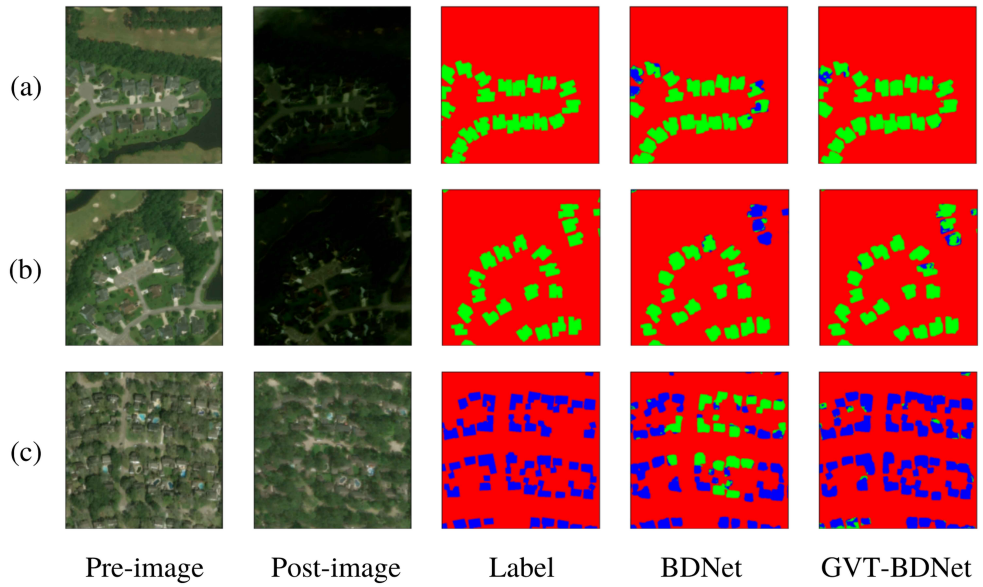

Figure 4.3: *Comparison between BDNet and GVT-BDNet predictions. In these examples we can clearly see an improvement in damage assessment for the GVT-BDNet model, even with high-contrast images (examples (a)-(b))*

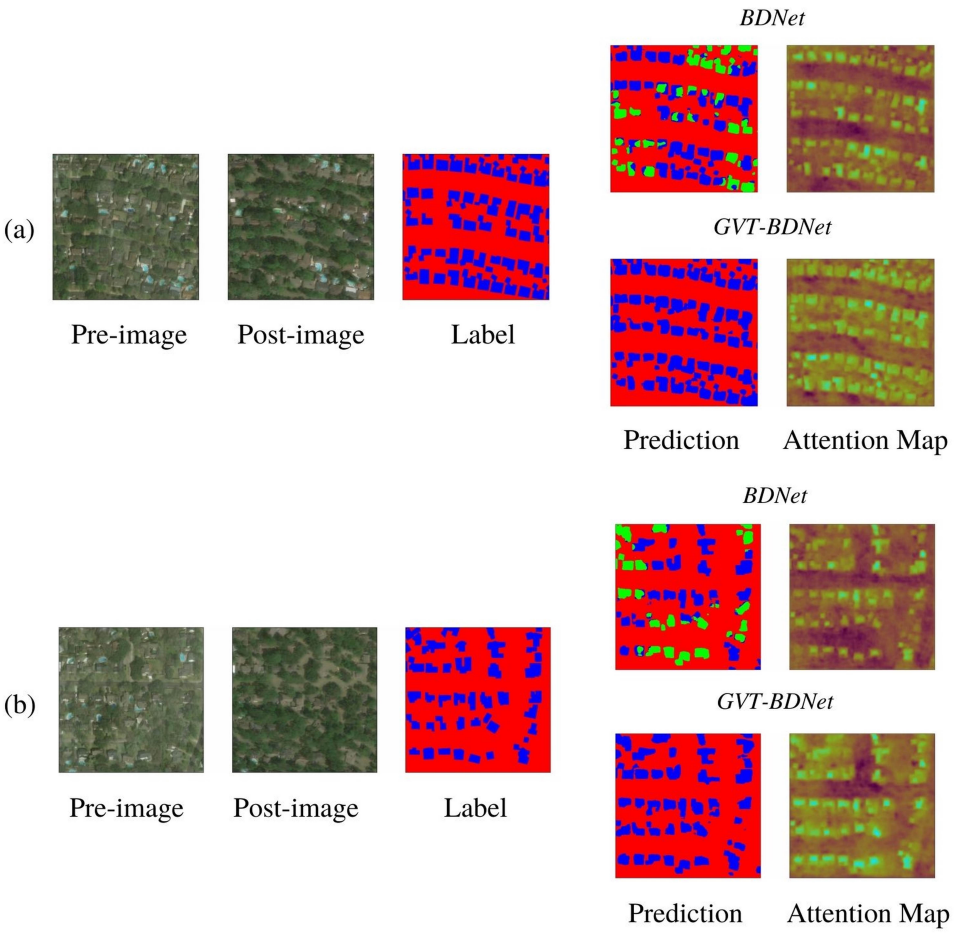

Figure 4.4: *Visualization of BDNet and GVT-BDNet prediction and relative attention map given a paired input images.*

thus improving the damage detection. We can see that this is particularly true as xBD damage labels value the damage score not only by the damages on the building itself but also at its surrounding (water surrounding or near the house, burned objects surrounding the house, encroaching volcanic flow in the nearby). We hypothesized that convolutional operations might have difficulties detecting those types of damage, as a fixed receptive field size could prevent them from spotting long-range dependencies. That limit might be overcame with attention operators, as they allow a global receptive field and thus facilitate the learning of long-distance damage features.

To support our hypothesis, we used attention maps. They are heatmaps representing the relative importance of layer activations with respect to the target task (in this case, BDA). When the neural network outputs a prediction,

one can visualize the attention map of the last convolution to see which part of the image has contributed more to the output. Specifically, the brightest the color, the most that part of the image has contributed to the prediction.

Figure 4.4 shows predictions and relative attention maps of BDNet and GVT-BDNet given a pair of input images. Pre-disaster and post-disaster images describe the damage suffered by buildings after a hurricane that caused extensive flooding (water surrounding buildings).

By looking at predictions and relative attention maps, we can see that BDNet struggled to capture damages in both examples. BDNet's attention maps are overall less bright in the surrounding of a building, meaning that little information has been used to retrieve that particular prediction. On the other hand, GVT-BDNet achieved higher damage segmentation performance. GVT-BDNet's attention maps are brighter than the BDNet ones within the surrounding of a building, meaning that more information has been used to retrieve the prediction. Overall, this analysis validates our initial hypothesis. BDNet's architecture seems to be less sensitive than the GVT-BDNet's one to the surroundings of buildings and therefore cannot precisely predict damages caused by floodings. Nevertheless, we encourage future research to build upon our results and contribute to the analysis.

4.3.1 Comparison with the state-of-the-art

To evaluate the final performance of our model, we compared GVT-BDNet with the state-of-the-art BDA models. As discussed in Section 3.1, we divided the comparison into two tables due to different data splits used in the evaluation phase by the BDA literature.

The first comparison is between state-of-the-art models that trained on 90% of the Tier1 folder and tested on 10% of the Tier1 folder. We trained from scratch and tested our model with this split to conduct the comparison. Table 4.4 shows the performance of BDNet and GVT-BDNet along with other state-of-the-art models. Results showed that our baseline neural network (BDNet) achieved and set a new state-of-the-art overall performance for building segmentation, with an F1-score $F1_b$ of 0.87. Similarly, GVT-BDNet improved and set a new state-of-the-art for damage assessment, with an F1-score $F1_d$ of 0.78. The overall performance $F1_o$ also improved with an F1-score of 0.8. Note that we did only use basic data augmentations with no advanced data re-sampling techniques, which are commonly adopted by most of the state-of-the-art models.

The second comparison comprehends state-of-the-art models trained on

Model	$F1_b$	$F1_d$	$F1_o$
Siam-U-Net (concatenation) [6]	0.73	0.69	0.70
Siam-U-Net (difference) [6]	0.73	0.70	0.70
RescueNet [46]	0.84	0.74	0.77
BDNet (with ResNet) [Ours]	**0.87**	0.73	0.77
GVT-BDNet (with ResNet) [Ours]	0.86	**0.78**	**0.8**

Table 4.4: *First comparison with state-of-the art BDA models. In this case all models have been trained on 90% of the Tier1 folder and tested on 10% of the Tier1 folder. BDNet and GVT-BDNet improved over the state-of-the-art for both building segmentation and damage assessment.*

Model	$F1_b$	$F1_d$	$F1_o$
Weber E. et al [19]	0.84	0.70	0.74
Shen Y. et al. [7]	**0.86**	**0.78**	**0.80**
GVT-BDNet (with ResNet) [Ours]	**0.86**	0.76	0.79

Table 4.5: *Second comparison with state-of-the art BDA models. In this case all models have been trained on the entire Tier1 folder and tested on the xBD holdout dataset. GVT-BDNet reached state-of-the-art performances for building segmentation and parallel results for damage assessment.*

the entire Tier1 folder and tested on the xBD test dataset. At this time, we trained from scratch and tested our neural network according to this split. Table 4.5 shows the performance of GVT-BDNet along with the other state-of-the-art models. Unlike the previous comparisons, GVT-BDNet did not improved, but reached, state-of-the-art performances. Our proposal neural network achieved an F1-score of 0.86 (in line with Shen Y. et al. model [7]) for building segmentation ($F1_b$), and parallel results for damage assessment ($F1_d$) and overall performance ($F1_o$).

Let us now analyse these comparison results deeper. Unlike other state-of-the-art BDA models, GVT-BDNet tackles class imbalance with a balancing loss function specifically designed for image segmentation without using strong data augmentation. Specifically, we adopted the Dice Loss as a loss function, which improved the overall F1-score performance of building damage segmentation upon the Weighted Cross-Entropy loss function. Moreover, the size-preserving GVTO demonstrated to be helpful for the segmentation of hard classes, and improved the model's ability to distinguish between undamaged buildings and damaged ones. Moreover, it even improved the performance that surpass the state-of-the-art, proving that

attention modules are a great solution to help classifiers at learning more complex and global information, and thus a solution for class imbalance. We also stress that, as the first step, we decided to only test the effectiveness of substituting the bottom convolutional layer of the encoder with a single size-preserving GVTO. Considering the state-of-the-art results already gotten by GVT-BDNet in certain scenarios, we believe more evaluations with different GVTO variants can be very interesting and promising, as down-sampling GVTO and up-sampling GVTO are yet to be investigated. As a follow-up to our analysis, one could carry on the investigation of GVTOs, and examine whether down-sampling GVTOs and up-sampling GVTOs could improve even more BDA performance. Another interesting and simple follow-up is to apply strong data augmentations to GVT-BDNet, which we avoided due to time constraints, but could be the key to achieve higher performance on hard classes.

4.4 Experiment 4: Generalizability evaluation on the T&S segmentation task

To better evaluate the flexibility of BDNet and GVT-BDNet, we decided to test the previously mentioned neural network architectures on a brand new task: T&S. The goal here is to understand if BDNet and the two-step training, could be translated to other imbalanced multi-class segmentation tasks while keep good results. However, since the T&S task is logically different from the BDA task, we implemented a variation of BDNet, named TSNet (Figure 3.4), and re-defined the two-step protocol (as discussed in Section 3.4.1). The main difference between TSNet and BDNet architectures is that the latter present a single-branch CNN for building segmentation (1st step) and a double-branch CNN with paired input images for building damage segmentation (2nd step); on the other hand, TSNet presents a single-branch with single input images for both steps (the description of TSNet steps can be found is section 3.4.1).

Table 4.6 shows the performance of TSNet and GVT-TSNet trained and tested on the Spacept dataset. Similar to GVT-BDNet, GVT-TSNet is a TSNet variant, which uses a size-preserving GVTO at the bottom of the pretrained ResNet-50 encoder. Therefore, the comparison shares the same spirits with the case of TSNet and GVT-BDNet for T&S segmentation. Results show that TSNet achieves an F1-score of 0.85 for tree segmentation ($F1_t$) and an F1-score of 0.73 for shadow segmentation ($F1_s$), with an overall F1-score of 0.79 ($F1_o$). On the other hand, GVT-TSNet achieves equal performance to TSNet,

Model	$F1_t$	$F1_s$	$F1_o$
TSNet	**0.85**	**0.73**	**0.79**
GVT-TSNet	0.84	**0.73**	0.785

Table 4.6: *performance of TSNet and GVT-TSNet with the Spacept data. In this case, 90% of the dataset has been used as train set, and 10% of the dataset has been used as test set. We can see that TSNet and GVT-TSNet have overall similar performance*

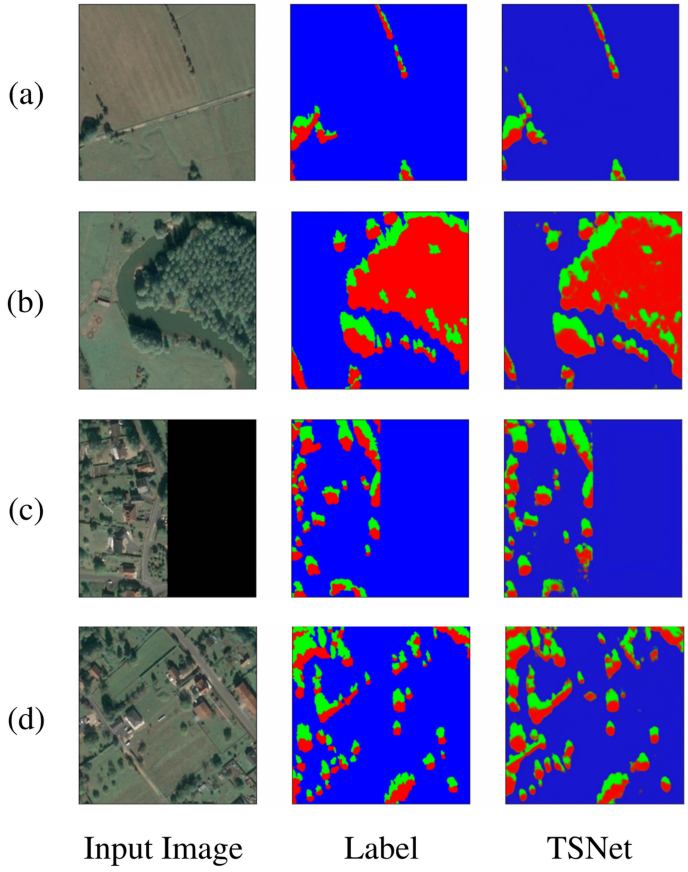

Figure 4.5: *Visualization of TSNet prediction for T&S segmentation. In this case, trees are colored in red, shadows in green, and the background in blue. We can see that overall predictions have high segmentation performance.*

with slightly lower results in tree segmentation and, consequently, in overall performance.

It is interesting to see that, unlike what one would have expected, the size-preserving GVTO is not improving the performance, but seems to have zero impact. This fact suggests that GVTOs, and attention operators in

general, might not be effective when no additional global view is needed to increase the performance and thus, their performance could be strictly related to the different tasks. Apart from that, the neural network architectures, and TSNet particularly, achieve good overall performance, which validates our initial assumption. Figure 4.5 shows the qualitative results of TSNet for T&S segmentation. We can see that, generally, the model achieves good performance on tree segmentation and shadow segmentation. We decided not to show the predictions of GVT-BDNet since they are very similar to the ones of TSNet and GVT-TSNet, and meaningful differences between the two cannot be extracted. Nevertheless, this is a validation of the flexibility and generalization performance of BDNet architecture.

Chapter 5

Conclusions and Future Works

This chapter presents a final discussion and the conclusions of our research. Moreover, we discuss limitations and future works that could be investigated as a follow-up to our research.

5.1 Conclusions

In our research, we proposed a flexible attention-operator, named Global Voxel Transformer Operator (GVTO), to improve performance and generalizability within the scope of ML-based BDA. GVTOs have been initially proposed for Augmented Microscopy, where they demonstrated their superiority over the U-Net architecture on several publicly available datasets. These attention operators replace convolutional up-sampling, down-sampling, and size-preserving operators, improving the extraction of global information and long-range dependencies. Moreover, unlike the other state-of-the-art attention operators for BDA, they are flexible, as they can be applied to any CNN architecture without being tied to a specific task or neural network architecture.

As the starting point of the investigation of GVTOs for BDA, we proposed and tested the BDNet, a double-branch U-Net with the bottom encoder convolution replaced by a size-preserving GVTO. We named the resultant model as GVT-BDNet. Results showed that GVT-BDNet achieves state-of-the-art performances on the xBD dataset, which is the largest and most diverse publicly available dataset for BDA. Specifically, GVT-BDNet improved hard-classes segmentation (damaged buildings) and increased generalizability.

To test the effectiveness of BDNet and GVT-BDNet towards generalizability, we implemented two of their variants, TSNet and GVT-TSNet, for Tree & Shadow segmentation. At this time, results showed that

the baseline TSNet and the GVTO variant, GVT-TSNet, achieves comparable results, which made us re-consider the power of GVTO and attention operators in general. From our perspective, attention operators and GVTO are a great opportunity to improve deep neural network performances in terms of generalizability and hard-classes accuracy. However, easy tasks as Tree & Shadow segmentation might not be suitable for such operators. In those cases, the aggregate global information collected by GVTOs and attention operators could be inefficient, and in some cases might be even counterproductive. However, both neural networks (TSNet and GVT-TSNet) still achieved high performances for the previously mentioned task, demonstrating the power of our baseline CNN.

Throughout our research, we also performed minor experiments, which are useful for future explorations for BDA. Most importantly, due to the class-imbalance nature of the xBD dataset, we benchmarked the most promising balancing loss function specifically for BDNet, our CNN baseline. We discovered that when loss functions specifically designed for image segmentation tasks are adopted (J regularization technique, Dice Loss), performances usually increase. Specifically, these loss functions outperformed the Weighted Cross Entropy, which is nowadays widely adopted by the BDA literature.

Moreover, we proposed a new guideline for the xBD benchmark, as we found the comparisons between state-of-the-art models are sometimes confusing and not straightforward. That means different evaluation metrics and datasets were used, making the comparison difficult to build. With our suggestions, we want to define a new protocol, which is going to facilitate and make the comparisons more accessible, accelerating future BDA development. We encourage the use of this proposed benchmark in future academic works. The new benchmark is described in detail in Section 3.1 and is now available at PapersWithCode's website [48].

5.2 Limitations

Some of our experiments were limited by time constraints and available computational resources. Specifically, for the first three experiments, we decided to train with a representative subset of the dataset used for the fourth experiment, as increased GPU power was not available, and because we wanted to keep the time costs for each training under 24h. On the other hand, for the last two and most important experiments, we trained with multi-GPU machines, utilizing the entire dataset. Each training was still under 24h.

Nevertheless, even if during the first experiments we trained with a representative subset of the data, we believe that results are reliable. Results from GVT-BDNet, trained with both Tier1-slice (Experiment 3) and the entire Tier1 folder (Experiment 4), did not present significant changes, thus increasing the statistical reliability of the first three experiments.

5.3 Future works

Some future works that could be built upon our research are:

- *Divide damage classes into three more sub-classes*: The xBD dataset defines three different damage classes to describe the type of damage suffered from a building. Those are minor damages, major damages, and destroyed. As a first step, we decided to relabel those three classes as a single class, but future works could investigate whether performances change when the damage class is divided into minor damages, major damages, and destroyed (and if so, to what extend).

- *Apply advanced data augmentation to BDNet and GVT-BDNet*: In our research, we decided not to use any advanced data augmentation. However, as we see from BDA literature, a benchmark with a set of more diversed augmentation techniques could be crucial to improve the performances of BDNet/GVT-BDNet, and the segmentation of hard classes even more. At the same time, it can also investigate the effectiveness of the state-of-the-art data augmentation techniques for BDA.

- *Feature Subtraction VS Feature Concatenation*: As we mentioned in the literature review, Google researchers discovered that double branch CNN models where pre-disaster and post-disaster features are subtracted before the convolutional encoder achieves overall better results than the ones that concatenate them [5]. It could be intuitive to test performances of BDNet and GVT-Net with a element-wise subtraction operation before the convolutional decoder instead of the currently adopted concatenation.

References

[1] Emergency event database of the centre for research on the epidemiology of disasters. [Online]. Available: https://public.emdat.be/

[2] J. Jeong, T. Yoon, and J. Park, "Towards a meaningful 3d map using a 3d lidar and a camera," vol. 18, p. 2571. doi: 10.3390/s18082571

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation." [Online]. Available: http://arxiv.org/abs/1505.04597

[4] Z. Wang, Y. Xie, and S. Ji, "Global voxel transformer networks for augmented microscopy." [Online]. Available: http://arxiv.org/abs/2008.02340

[5] J. Z. Xu, W. Lu, Z. Li, P. Khaitan, and V. Zaytseva, "Building damage detection in satellite imagery using convolutional neural networks." [Online]. Available: http://arxiv.org/abs/1910.06444

[6] H. Hao, S. Baireddy, E. R. Bartusiak, L. Konz, K. LaTourette, M. Gribbons, M. Chan, M. L. Comer, and E. J. Delp, "An Attention-Based System for Damage Assessment Using Satellite Imagery," *arXiv:2004.06643 [cs]*, Apr. 2020, arXiv: 2004.06643. [Online]. Available: http://arxiv.org/abs/2004.06643

[7] Y. Shen, S. Zhu, T. Yang, and C. Chen, "Cross-directional feature fusion network for building damage assessment from satellite imagery." [Online]. Available: http://arxiv.org/abs/2010.14014

[8] R. Gupta, R. Hosfelt, S. Sajeev, N. Patel, B. Goodman, J. Doshi, E. Heim, H. Choset, and M. Gaston, "xbd: A dataset for assessing building damage from satellite imagery," 2019.

[9] "Climate change: How hot cities could be in 2050." [Online]. Available: https://www.bbc.com/news/newsbeat-48947573

[10] Environmental health in emergencies. [Online]. Available: https://www.who.int/teams/environment-climate-change-and-health/emergencies

[11] A. Suppasri, S. Koshimura, M. Matsuoka, H. Gokon, and D. Kamthonkiat, "Application of remote sensing for tsunami disaster," in *Remote Sensing of Planet Earth*, Y. Chemin, Ed. InTech. ISBN 978-953-307-919-6. [Online]. Available: http://www.intechopen.com/books/remote-sensing-of-planet-earth/application-of-remote-sensing-for-tsunami-disaster

[12] H. Gokon and S. Koshimura, "Mapping of building damage of the 2011 tohoku earthquake tsunami in miyagi prefecture," vol. 54, no. 1, pp. 1 250 006–1–1 250 006–12. doi: 10.1142/S0578563412500064. [Online]. Available: https://www.tandfonline.com/doi/full/10.1142/S0578563412500064

[13] G. Finnveden, "On the limitations of life cycle assessment and environmental systems analysis tools in general," *The International Journal of Life Cycle Assessment*, vol. 5, no. 4, p. 229, Jul. 2000. doi: 10.1007/BF02979365. [Online]. Available: https://doi.org/10.1007/BF02979365

[14] D. Vuković. Why the first 72 hours after a disaster are critical. [Online]. Available: https://www.primalsurvivor.net/why-the-first-72-hours-after-a-disaster-are-critical/

[15] M. P. CNN. The critical 72 hours after nepal earthquake. [Online]. Available: https://www.cnn.com/2015/04/25/asia/nepal-earthquake-challenges/index.html

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," pp. 815–823. doi: 10.1109/CVPR.2015.7298682. [Online]. Available: http://arxiv.org/abs/1503.03832

[17] H.-R. Chou, J.-H. Lee, Y.-M. Chan, and C.-S. Chen, "Data-specific adaptive threshold for face recognition and authentication," version: 1. [Online]. Available: http://arxiv.org/abs/1810.11160

[18] M. Yaqub, J. Feng, M. S. Zia, K. Arshid, K. Jia, Z. U. Rehman, and A. Mehmood, "State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images," vol. 10, no. 7, p. 427. doi: 10.3390/brainsci10070427 Number: 7 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-3425/10/7/427

[19] E. Weber and H. Kané, "Building disaster damage assessment in satellite imagery with multi-temporal fusion." [Online]. Available: http://arxiv.org/abs/2004.05525

[20] F. Nex, D. Duarte, F. G. Tonolo, and N. Kerle, "Structural building damage detection with deep learning: Assessment of a state-of-the-art CNN in operational conditions," vol. 11, no. 23, p. 2765. doi: 10.3390/rs11232765 Number: 23 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2072-4292/11/23/2765

[21] L.-A. Tran and M.-H. Le, "Robust u-net-based road lane markings detection for autonomous driving," in *2019 International Conference on System Science and Engineering (ICSSE)*. doi: 10.1109/ICSSE.2019.8823532 pp. 62–66, ISSN: 2325-0925.

[22] A. Rakhlin, A. Davydow, and S. Nikolenko, "Land cover classification from satellite imagery with u-net and lovász-softmax loss," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. doi: 10.1109/CVPRW.2018.00048. ISBN 978-1-5386-6100-0 pp. 257–2574. [Online]. Available: https://ieeexplore.ieee.org/document/8575508/

[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" [Online]. Available: http://arxiv.org/abs/1411.1792

[24] J. Xie, "Transfer learning with deep neural networks for computer vision," accepted: 2019-05-02T23:18:30Z. [Online]. Available: https://digital.lib.washington.edu:443/researchworks/handle/1773/43663

[25] Z. Alyafeai, M. S. AlShaibani, and I. Ahmad, "A survey on transfer learning in natural language processing." [Online]. Available: http://arxiv.org/abs/2007.04239

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." [Online]. Available: http://arxiv.org/abs/1512.03385

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, Dec. 2017, arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762

[28] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local Neural Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018. doi: 10.1109/CVPR.2018.00813 pp. 7794–7803, iSSN: 2575-7075.

[29] User guides - sentinel-2 MSI - overview - sentinel online - sentinel. [Online]. Available: https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-2-msi/overview

[30] B. Kalantar, N. Ueda, H. A. H. Al-Najjar, and A. A. Halin, "Assessment of convolutional neural network architectures for earthquake-induced building damage detection based on pre- and post-event orthophoto images," vol. 12, no. 21, p. 3529. doi: 10.3390/rs12213529 Number: 21 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2072-4292/12/21/3529

[31] Z. Chen, M. Wagner, J. Das, R. K. Doe, and R. S. Cerveny, "Data-driven approaches for tornado damage estimation with unpiloted aerial systems," vol. 13, no. 9, p. 1669. doi: 10.3390/rs13091669 Number: 9 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2072-4292/13/9/1669

[32] Q. D. Cao and Y. Choe, "Post-hurricane damage assessment using satellite imagery and geolocation features." [Online]. Available: http://arxiv.org/abs/2012.08624

[33] xView2. [Online]. Available: https://xview2.org/

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection." [Online]. Available: http://arxiv.org/abs/1708.02002

[35] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation." [Online]. Available: http://arxiv.org/abs/1606.04797

[36] N. United, "THE 17 GOALS | Sustainable Development." [Online]. Available: https://sdgs.un.org/goals

[37] F. A. G. Peña, P. D. M. Fernandez, P. T. Tarr, T. I. Ren, E. M. Meyerowitz, and A. Cunha, "J regularization improves imbalanced multiclass segmentation." [Online]. Available: http://arxiv.org/abs/1910.09783

[38] S. Koshimura and N. Shuto, "Response to the 2011 great east japan earthquake and tsunami disaster," vol. 373, no. 2053, p. 20140373. doi: 10.1098/rsta.2014.0373 Publisher: Royal Society. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rsta.2014.0373

[39] E. Mas, J. Bricker, S. Kure, B. Adriano, C. Yi, A. Suppasri, and S. Koshimura, "Field survey report and satellite image interpretation of the 2013 super typhoon haiyan in the philippines," vol. 15, no. 4, pp. 805–816. doi: 10.5194/nhess-15-805-2015 Publisher: Copernicus GmbH. [Online]. Available: https://nhess.copernicus.org/articles/15/805/2015/

[40] N. Mori, T. Takahashi, and THE 2011 TOHOKU EARTHQUAKE TSUNAMI JOINT SURVEY GROUP, "Nationwide post event survey and analysis of the 2011 tohoku earthquake tsunami," vol. 54, no. 1, pp. 1 250 001–1–1 250 001–27. doi: 10.1142/S0578563412500015. [Online]. Available: https://www.tandfonline.com/doi/full/10.1142/S0578563412500015

[41] F. Dell'Acqua and P. Gamba, "Remote Sensing and Earthquake Damage Assessment: Experiences, Limits, and Perspectives," *Proceedings of the IEEE*, vol. 100, no. 10, pp. 2876–2890, Oct. 2012. doi: 10.1109/JPROC.2012.2196404 Conference Name: Proceedings of the IEEE.

[42] R. T. Eguchi, C. K. Huyck, S. Ghosh, B. J. Adams, and A. McMillan, "Utilizing New Technologies in Managing Hazards and Disasters," in *Geospatial Techniques in Urban Hazard and Disaster Analysis*, ser. Geotechnologies and the Environment, P. S. Showalter and Y. Lu, Eds. Dordrecht: Springer Netherlands, 2010, pp. 295–323. ISBN 978-90-481-2238-7. [Online]. Available: https://doi.org/10.1007/978-90-481-2238-7_15

[43] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp.

1527–1554, Jul. 2006. doi: 10.1162/neco.2006.18.7.1527. [Online]. Available: https://doi.org/10.1162/neco.2006.18.7.1527

[44] M. Ji, L. Liu, and M. Buchroithner, "Identifying Collapsed Buildings Using Post-Earthquake Satellite Imagery and Convolutional Neural Networks: A Case Study of the 2010 Haiti Earthquake," *Remote Sensing*, vol. 10, no. 11, p. 1689, Nov. 2018. doi: 10.3390/rs10111689 Number: 11 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2072-4292/10/11/1689

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017. doi: 10.1145/3065386. [Online]. Available: https://dl.acm.org/doi/10.1145/3065386

[46] R. Gupta and M. Shah, "RescueNet: Joint Building Segmentation and Damage Assessment from Satellite Imagery," *arXiv:2004.07312 [cs, eess]*, Apr. 2020, arXiv: 2004.07312. [Online]. Available: http://arxiv.org/abs/2004.07312

[47] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features," *arXiv:1905.04899 [cs]*, Aug. 2019, arXiv: 1905.04899. [Online]. Available: http://arxiv.org/abs/1905.04899

[48] "Papers with Code - xBD Benchmark (2D Semantic Segmentation)." [Online]. Available: https://paperswithcode.com/sota/2d-semantic-segmentation-on-xbd

[49] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950. doi: 10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3. [Online]. Available: https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.1002/1097-0142%281950%293%3A1%3C32%3A%3AAID-CNCR2820030106%3E3.0.CO%3B2-3

[50] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017, arXiv: 1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

For DIVA

{
"Author1": {
"Last name": "Remondini",
"First name": "Leonardo",
"Local User Id": "19970128-T512",
"E-mail": "lre@kth.se",
"organisation": {"L1": "School of Electrical Engineering and Computer Science ",
}
},
"Degree": {"Educational program": "Master's Programme, ICT Innovation, 120 credits"},
"Title": {
"Main title": "GVT-BDNet: Convolutional Neural Network with Global Voxel Transformer Operators for Building Damage Assessment",
"Language": "eng" },
"Alternative title": {
"Main title": "GVT-BDNet: Convolutional Neural Network med Global Voxel Transformer Operators för Building Damage Assessment",
"Language": "swe"
},
"Supervisor1": {
"Last name": "Hu",
"First name": "Hao",
"Local User Id": "u100003",
"E-mail": "haohu@kth.se",
"organisation": {"L1": "School of Electrical Engineering and Computer Science ",
"L2": "Computer Science" }
},
"Supervisor2": {
"Last name": "Iliev",
"First name": "Sergiu Petre",
"E-mail": "sergiu@iliev.us",
},
"Examiner1": {
"Last name": "Aristides",
"First name": "Gionis",
"Local User Id": "u100004",
"E-mail": "argioni@kth.se",
"organisation": {"L1": "School of Electrical Engineering and Computer Science ",
"L2": "Computer Science" }
},
"Cooperation": { "Partner_name": "Spacept"},
"Other information": {
"Year": "2021", "Number of pages": "xv,65"}
}

TRITA -EECS-EX