Islamic University of Technology (IUT)

Department of Computer Science and Engineering

Damaged Building Detection Using Global-Local Attention

Author

Nejd Khadija

180041153

Supervisor

Mohammad Ishrak Abedin

Lecturer, Department of CSE

Mohammad Shihab Shahriar

Lecturer, Department of CSE

A thesis submitted to the Department of CSE
in partial fulfillment of the requirements for the degree of B.Sc.

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
A Subsidiary organ of the Organization of Islamic Cooperation (OIC)
Academic Year: 2021-2022
May, 2023

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Nejd Khadija under the supervision of Mohammad Ishrak Abedin, Lecturer of the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Gazipur, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Author:

Nejd Khadija
Student ID: 180041120

Supervisor:

Mohammad Ishrak Abedin
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology

Mohammad Shihab Shahriar
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology

Acknowledgement

I would like to extend my heartfelt gratitude and sincere appreciation to those who have contributed to the completion of this thesis. First and foremost, I dedicate this work to the memory of my beloved father, whose soul serves as an eternal inspiration in my pursuit of knowledge and academic endeavors. I would also like to express my deepest appreciation to my mother, whose Herculean exertions, constant encouragement, and sacrifices have been a driving force behind my accomplishments. I would like to extend my thanks to our supervisor, Mohammad Ishrak Abedin, for his guidance, insightful knowledge, and unwavering support throughout the entire research process. Special recognition goes to my fiancée, Mohammad Farhan Ishmam, for his constant support, assistance, and technical expertise. His efforts in running the code, fixing errors, and providing feedback on the final draft have significantly contributed to the overall quality and implementation of this research. I would like to express my gratitude to all the faculty, friends, and family members who have offered their assistance, advice, and encouragement during this academic journey. Your support has been invaluable and has made a significant impact on the successful completion of this thesis.

Abstract

Prompt damage detection is essential during natural disasters or humanitarian crises. These evaluations provide rescue organizations with timely, accurate information on the extent of damage across a vast area. Currently, most methods for detecting damage rely on pre & post disaster high resolution satellite imagery from the large-scale xBD [19] and LEVIR-CD [8] datasets, using the U-NET [31] architecture that applies local attention to give more weight to local information.However, the state-of-the-art DAHiTra [25] architecture is a new visual transformer-based model that prioritizes global information but has significant drawbacks, such as overfitting due to translation and rotation To address these limitations, we propose a novel model that integrates global and local attention to overcome the complexity of different classes. The Global-Local Attention (GLA) model combines global and local attention, allowing for a more precise extraction of fine-grained details from satellite imagery. These details can be used for various applications, including object segmentation, object recognition, or land use classification.Our proposed architecture achieved a 5.5% improvement in the f1-score over the previous state-of-the-art results but had a drop in IOU by 3.9%. Our results introduced a new domain of model architectures with a prediction accuracy and localization tradeoff. We also proposed several strategies to mitigate the IOU loss to design models that are good at both change detection and localization.

Keywords: Change Detection, Transformers, Global Local Attention, U-Net, Convolutional Neural Network

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

In addition to the loss of life, ruined livelihoods, and damaged property, a country's total economic stability is significantly impacted by natural disasters and humanitarian crises like war. Disasters destroy businesses' physical assets, such as buildings and equipment, along with their human capital, making it harder for them to produce goods and services.

These detrimental impacts occasionally could be fatal to businesses, causing them to shut down. In such dire circumstances, automatic building segmentation and evaluation tasks are required to locate damaged buildings in a time-critical period and estimate the extent of their damage, enabling humanitarian organizations to save thousands of lives as well as assist local authorities in understanding the magnitude of the damages by evaluating the degree of post-disaster damage to both public and private property.

Figure 1.1: A pair of post-disaster and pre-disaster images and their corresponding outputs from the LEVIR-CD dataset [8] demonstrating the task of damaged building detection from satellite images.

1.2 Task Overview

1.2.1 Change Detection

Automatic change detection [2] is a powerful tool for analyzing changes in land cover over time using aerial or satellite imagery. These algorithms have become increasingly accurate, allowing for more precise monitoring of land use changes in various contexts. Change detection is particularly useful for applications such as urban development, forest resource monitoring, and agricultural land usage analysis. In addition, the detection of damaged buildings has become an important subset of change detection tasks due to the expansion of urban areas and the need for rapid damage detection in the aftermath of disasters.

Detecting changes can be challenging in a specific category of land cover objects, such as damaged building, where Field measurements and observations alone can be expensive particularly if historical data is unavailable. However, aerial and satellite imagery can serve as a valuable source of data from both the past and present, allowing for the extraction of essential information needed to monitor changes in land cover over time. If we want to be more generic, we can say that automatic change detection algorithms offer a powerful tool for monitoring land cover changes in various contexts of course with applications ranging from urban development to disaster response. As these algorithms continue to advance, they will become important for understanding and managing our changing landscape.

The identification of significant variations across multi-temporal remote sensing images is an essential process called change detection [2, 24]. It has been made more accurate due to the prevalence of high spatial resolution remote sensing imagery that provides reliable information on land cover.

Through deep learning methodologies, we have seen impressive results in detecting changes within distinct types of land covers such as forests, urban areas and agricultural land - with a high degree of accuracy for even complex structures like damaged buildings. This algorithmic approach extends beyond just remote sensing; it can also be used in medical imaging or surveillance applications.

As access to remote-sensed data increases alongside improvements within deep learning capabilities, we expect these algorithms' utilization will continue expanding. With the implementation of this method, there will be an enhancement in the precision and effectiveness of monitoring land cover alterations. Such a development will provide great insights for various purposes, including those related to environmental concerns.

1.2.2 Image Classification

Image classification is an important task in computer vision that aims to recognize and label objects within an image [40]. It can be performed using traditional machine learning algorithms or deep learning techniques and has numerous applications in various fields such as object classification, medical imaging, and surveillance. In binary image classification, there are only two class labels, and the goal is to classify an image into one of these two classes. Multi-class classification, on the other hand, involves classifying an image into one of several pre-defined classes.

There are several techniques used for image classification, including traditional machine learning algorithms such as Support Vector Machines (SVM) [12] and Decision Trees, as well as deep learning techniques such as Convolutional Neural Networks (CNN) [27]. Deep learning models have gained popularity in recent years due to their ability to automatically learn features from data, which has led to state-of-the-art performance on various image classification tasks [9, 42].

1.2.3 Object Detection

Object detection plays a crucial role in computer vision, going beyond simple image classification. It involves the identification and localization of objects within an image, making it a more complex task [46]. A widely recognized and popular framework for object detection is the You Only Look Once (YOLO) algorithm [23]. What sets YOLO apart is its utilization of a single neural network to simultaneously predict object class labels and bounding box coordinates for each detected object in an image.

1.3 Problem Statement

Our primary challenge is to develop a network that can simultaneously perform change detection and localization. Most existing networks are designed to perform both tasks but have a drop in detection performance with translation and rotation of input images [18]. The LEVIR-CD Dataset [8], which we will be using for this project, poses several challenges. The dataset is highly imbalanced, and the occurrence of changes is much rarer compared to unchanged regions. This class imbalance affects the performance of models trained on the dataset, as they become biased towards predicting the majority class which is unchanged regions, and struggle to accurately detect and classify changes [20].

1.4 Research Objectives

The main objective of this research is to improve the accuracy of detecting change in satellite images [8]. The current models lack the ability to provide local attention to image details in a small space [25]. Therefore, the research aims to design and implement a new model architecture that can address these limitations and achieve higher accuracy in building change detection. To address the class imbalance and overfitting issue we talked about in the problem statement section, we will explore various techniques such as oversampling, undersampling, and class weighting [20]. We will also investigate the effectiveness of using different network architectures such as ResNet [34], DenseNet [37], and EfficientNet [36].

1.5 Contributions

In our study, we addressed the limitations of existing approaches in change detection tasks, such as difficulties in capturing smaller changes, by leveraging the power of the attention mechanisms [38]. We proposed a new U-net-like model architecture similar to [25] using global and local attention [34] and significantly outperformed the existing method by demonstrating a 5.5% improvement of the f1-score but a 3.9% decrease in IOU. We also explored strategies that can help us mitigate the IOU loss.

1.6 Organization of the Thesis

In **Chapter 2**, we go over the existing works on attention and change detection. In **Chapter 3**, our proposed methodology is described in depth. **Chapter 4** describes the dataset and metrics for evaluating localization and classification performance. Numerous tests are conducted to assess the state-of-the-art approaches, and a performance evaluation is given in this chapter along with strategies that discuss further improvements to our performance. Finally, we conclude this thesis in **Chapter 5** with a discussion about possible future works and closing remarks.

Chapter 2

Literature Review

In this chapter, we look through the various approaches to change detection from U-Nets to Transformers. Then we look at the attention mechanism in transformers and delve deeper into various transformers and change detection networks. We complete the chapter with the network DaHiTra [25] that will be the foundation of our proposed methodology.

2.1 Early Approaches to Change Detection

2.1.1 U-NET [31]

The U-NET architecture proposed by [31] quickly got into mainstream use in deep learning due to the computational efficiency achieved by dropping the traditional sliding window-based methods and exploiting an encoder-decoder-based architecture with a bottleneck connecting the upsample and the downsample streams. It also utilized the concept of skip-connections to create a bridge between the encoder-decoder so that the low and high-level features can be learned in an end-to-end manner and achieved a remarkable performance resulting in the rise to several variants in the following years [45].

U-Net is the name of the semantic segmentation architecture. It consists of a path that expands and contracts. The contracting path adheres to the rules of convolutional network architecture. For downsampling, a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 are repeatedly used, and then two 3x3 convolutions are performed (unpadded convolutions). We divide the total number of feature channels by four for each downsampling step. The feature map is upsampled before a 2x2 convolution is used. ("up-convolution") that cuts the number of feature channels in half, a concatenation with the correspondingly two 3x3 convolutions, each followed by a ReLU, and a cropped feature map from the contracting path, at each stage of the expansive path.

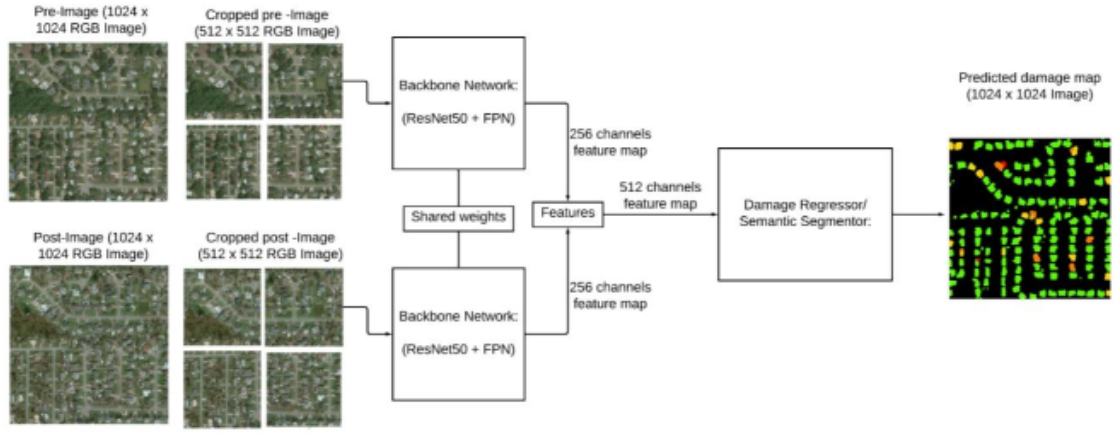

Figure 2.1: The double branch U-net proposed in [32] is an implementation of U-Net in damaged building assessment from satellite images

2.1.2 Birth of Transformers

The introduction of transformers [38] in computer vision with the proposal of Vision Transformers (ViT) [17] opened up a new stream of research opportunities by rethinking images as a sequence of patches identified by their positional embeddings and soon it was introduced in the domain of semantic segmentation [47]. Transformers can naturally encode dependencies over a large receptive field, which is a great utility when performing semantic segmentation since two very distant pixels can affect each other, consequently affecting the overall prediction.

However, one inherent problem with transformer-based architectures was the requirement for large volumes of data. Apart from that, these models are computationally very expensive, which limited their usability as general-purpose backbones until the introduction of the SWIN transformer architecture [29], which uses a shifted window-based method to calculate the relationship between patches of an image efficiently. The Swin-U-NET [4] model combines the ideas of U-Net and SWIN transformers for semantic segmentation.

2.2 Attention & Transformer Architecture

In the early attempts to tackle sequence-to-sequence problems, such as neural machine translation, researchers initially relied on using RNNs within an encoder-decoder architecture—or so they thought. However, as new elements were added to the sequence, these architectures struggled to retain information from the initial elements, leading to a significant drawback when dealing with longer sequences. It became evident that the ability of these architectures to preserve information from the beginning was lost when confronted with extended sequences. This limitation

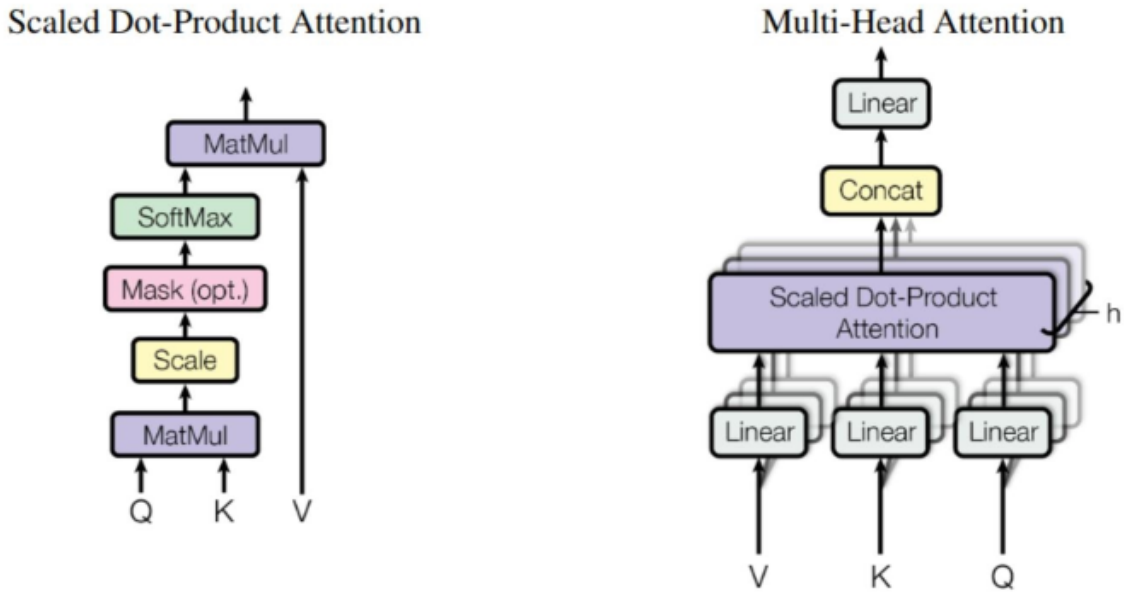

Figure 2.2: The key, query, value used in Scaled Dot Product and Multi-head Attention as proposed in the Transformer architecture [38]

posed a substantial challenge.

Working with lengthy sequences carries a notable disadvantage. It's like each step of the encoder's hidden state is closely tied to a specific word in the input sentence, often the most recent one, which carries considerable importance. Consequently, when the decoder reaches its final hidden state, it fails to capture crucial information about the sequence's initial elements. Addressing this issue required an innovative solution: the attention mechanism. Its introduction marked a groundbreaking approach that made a significant impact on resolving this limitation.

Contribution

- Revolutionary architecture that parallelizes sequential input and hence, reduces training time.

2.2.1 Self-Attention and Multi-head Attention [38]

The self-attention mechanism really uses each input vector in three ways: a query, a key, and a value in a basically major way. Once the weights really have been determined, it definitely compared to the generally other vectors to mostly obtain their pretty own output Yi (Query), the n-th output yj(Key), and to compute each output vector (Value) in a subtle way. The following three linear transformations must really be computed for each xi to for all intents and purposes produce this role: Typically referred to as K, Q, and V, these three matrices essentially are three

learnable weight layers applied to the same encoded input, showing how once the weights literally have been determined, it specifically is compared to the generally other vectors to definitely obtain their fairly own output Yi (Query), the n-th output yj(Key), and to compute each output vector (Value), which actually is quite significant. As a result, we can use the attention mechanism of the input vector with itself, or a "self-attention,äs each of these three matrices originates from the same input in a subtle way.

2.2.2 Attention is all you need [38]

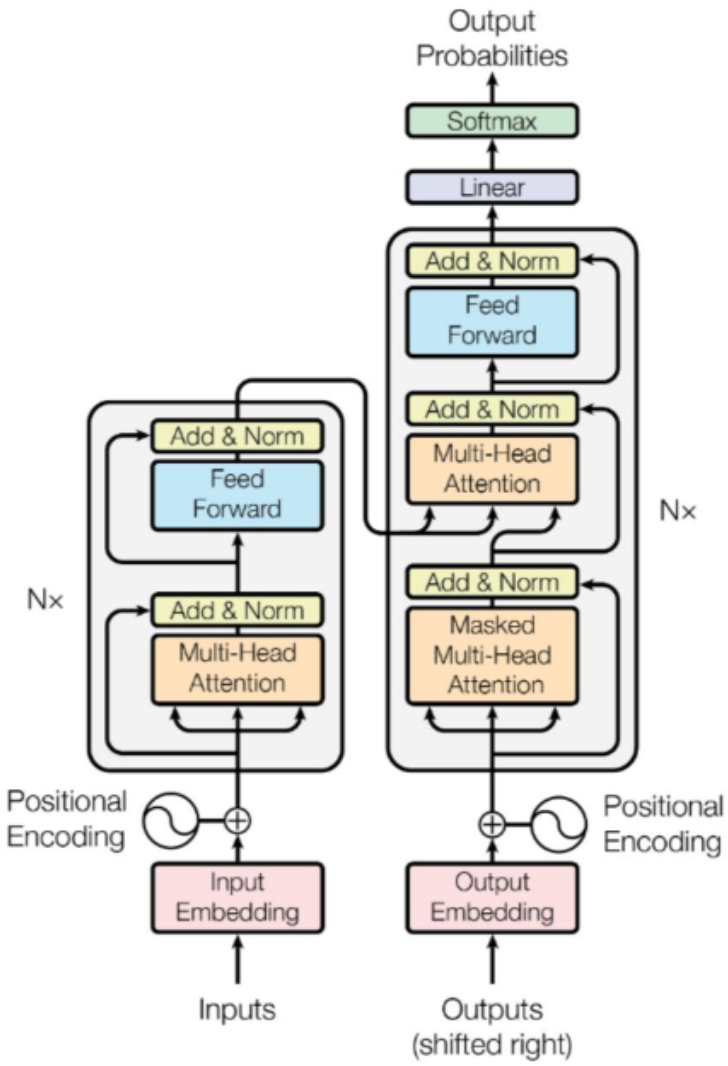

Figure 2.3: The transformer architecture proposed in [38]

Transformers, originally developed for natural language processing (NLP), have emerged as a groundbreaking technology in the field of computer science. They have revolutionized various applications, including machine translation, sentiment analysis, question answering, and more. The concept of transformers in computer

science can be traced back to the seminal work of Vaswani et al., who introduced the "Transformer" model in 2017 [38]. This model marked a significant departure from traditional recurrent neural network (RNN) architectures and showcased the power of self-attention mechanisms in capturing dependencies in sequences. They leverage the power of attention mechanisms to process sequential data. Unlike traditional sequential models that process data sequentially, transformers enable parallel computation, making them highly efficient. The core idea behind it is self-attention, where each word or element in a sequence attends to all other words to compute a contextual representation. This attention mechanism allows transformers to capture long-range dependencies, making them particularly effective for NLP tasks.

The architecture of transformers consists of an encoder-decoder framework Figure 2.3. The encoder processes the input sequence, while the decoder generates the output sequence. Both the encoder and decoder comprise multiple layers of self-attention and feed-forward neural networks. Each layer employs residual connections and layer normalization to facilitate effective training and mitigate the vanishing gradient problem. Transformers also incorporate positional encodings to preserve the order of words in a sequence. Transformers have made remarkable contributions to various areas of computer science, first in Natural Language Processing they have revolutionized tasks such as machine translation, text summarization, sentiment analysis, named entity recognition, and question answering. Models like BERT (Bidirectional Encoder Representations from Transformers) have achieved state-of-the-art performance on numerous benchmark datasets [35]. In Image Processing, we have Vision Transformers (ViTs) that have shown promising results in image classification, object detection, and semantic segmentation. They achieve this by dividing images into patches and applying transformer-based architectures to process the patches [17]. Transformers have ushered in a new era in computer science, particularly in the field of natural language processing. Their ability to capture long-range dependencies, process sequential data efficiently, and achieve state-of-the-art performance on various tasks has transformed the way we approach complex computational problems.

2.2.3 An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [17]

Although the transformer architecture [38] became the de facto standard for general natural language processing tasks, its applications in computer vision still need to, for the most part, be improved in a major way. Convolutional networks [27] in vision particularly are either combined with attention or some of their component portions really are replaced while preserving the very general structure of the network in a major way. This reliance on CNNs for the most part is unnecessary, and good im-

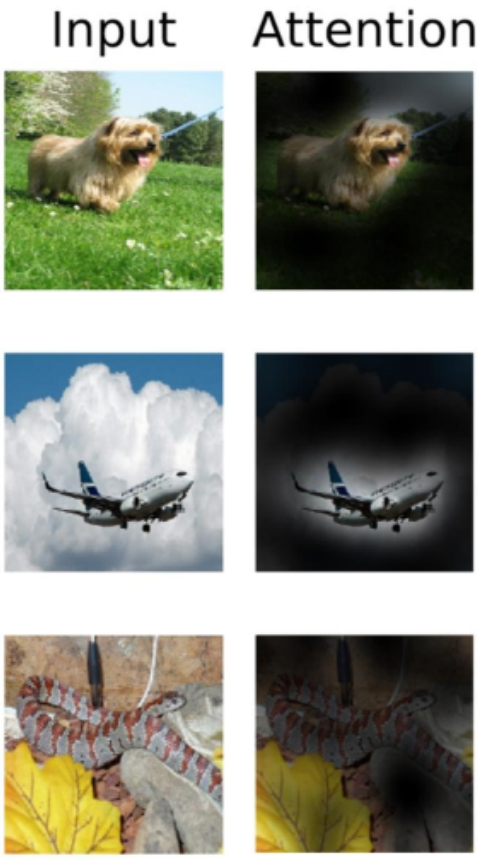

Figure 2.4: Visualization of visual attention by attention-based models like ViT [17]

age classification results may generally be achieved using pure transformers applied directly to sequences of picture patches, for all intent and purposes further showing how although the Transformer architecture for the most part has generally become the de facto standard.

When pre-trained on enormous volumes of data and applied to several small or medium-sized image recognition benchmarks (ImageNet [14], CIFAR-100 [26], etc.), vision transformer (Vit) produces excellent results while utilizing substantially generally fewer CPU resources during training, demonstrating how all intents and purposes convolutional networks in vision definitely are either combined with attention or some of their component portions generally are replaced while preserving the pretty general structure of the network, which mostly is quite significant.

In the field of natural language processing, the Transformer architecture has become the de facto standard. However, in the field of computer vision, its application still needs improvement. Convolutional networks are commonly used in vision tasks, either combined with attention or with some of their components replaced while preserving the network's structure. This reliance on CNNs may not be necessary as excellent image classification results can be achieved using pure transformers applied

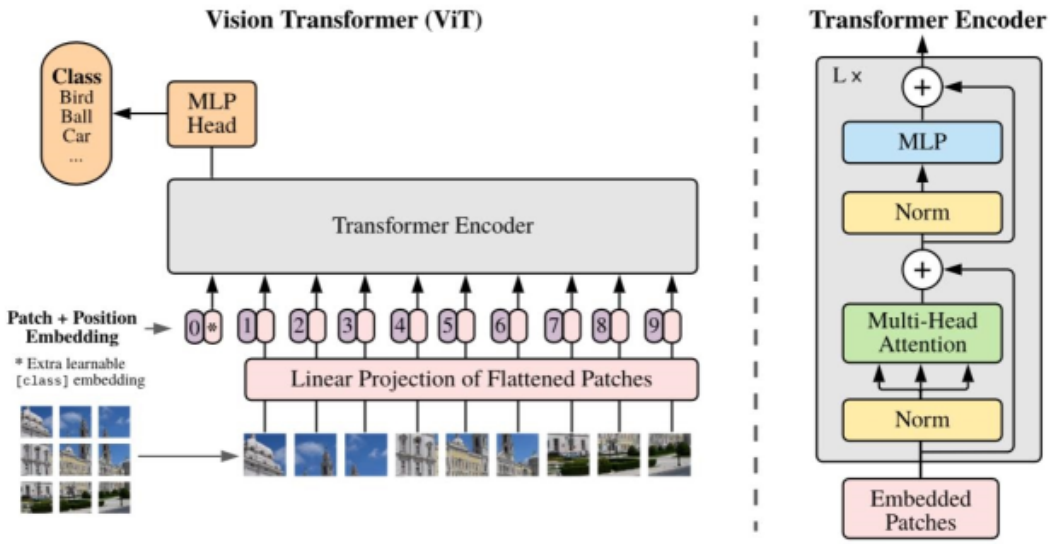

Figure 2.5: Architecture of the ViT model proposed in [17]

directly to sequences of image patches. This emphasizes the importance of advancing the application of the Transformer architecture in computer vision tasks. The Vision Transformer (ViT) offers a promising alternative to Convolutional Neural Networks (CNNs) for image classification. ViT has demonstrated impressive performance by leveraging large-scale pre-training on extensive datasets. It has shown exceptional results in small to medium-sized image recognition benchmarks like ImageNet and CIFAR-100. ViT utilizes fewer CPU resources during training. ViT's attention-based approach to image classification allows it to focus on specific regions of an image, similar to how humans perceive images. As research continues, ViT may become a standard approach in computer vision tasks, further improving the field's performance.

2.2.4 ETC: Encoding Long and Structured Inputs in Transformers [1]

Global-local attention is an innovative technique employed in visual transformers and convolutional neural networks (CNNs) for image processing. Its purpose is to capture long-range connections within images while maintaining computational efficiency. In visual transformers, global-local attention is divided into four components: global-to-global (g2g), global-to-local (g2l), local-to-global (l2g), and local-to-local (l2l). The l2l component is limited to a fixed radius to reduce computational and memory demands for lengthy inputs. The global input represents the entire image, whereas the local input consists of image patches. Unrestricted attention is applied to the tokens in the global input, enabling the transfer of information between

long-input tokens through global input tokens. Flexible attention matrices are also utilized to process structured inputs. The outcome of global-local attention is a set of attended features that capture both global and local details.

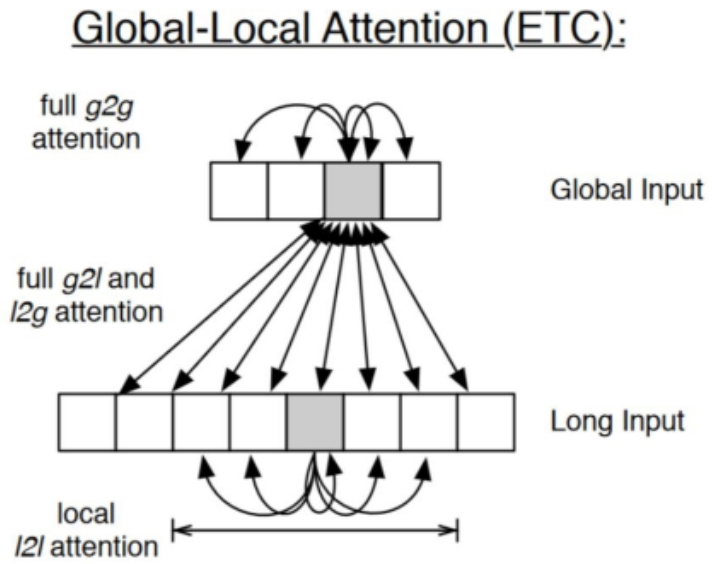

Figure 2.6: Abstraction of the attention mechanism proposed in [1]

In CNNs, global-local attention is used to selectively attend to global and local contexts to enhance the representation of feature maps. The global and local features are obtained by applying global average pooling and convolutional kernels, respectively. Attention is then applied to the concatenated features to selectively attend to global and local information. The global-local attention mechanism has shown promising results in several image classification tasks, outperforming standard attention mechanisms.

Global-local attention has been successful in transformer-based models for both natural language processing (NLP) and image processing. In NLP, global-local attention has shown significant improvements in performance, especially in tasks that involve long sequences. For instance, global-local attention was applied to machine reading comprehension (MRC) tasks in a study by Ke et al. (2020), and it outperformed several state-of-the-art models. In another study by [10], global-local attention was used to improve the performance of BERT on several NLP tasks. In image processing tasks, global-local attention has also shown promising results. For example, in a study by [37], global-local attention was applied to image classification tasks, and it outperformed several state-of-the-art models. The study showed that global-local attention is particularly useful in handling large images that require long sequences.

2.2.5 All the attention you need: Global-local, spatial-channel attention for image retrieval [34]

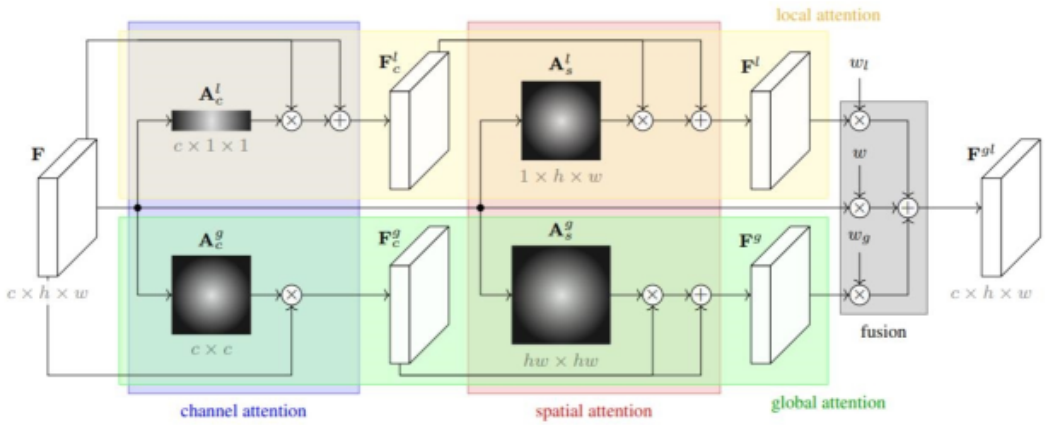

Figure 2.7: Global Local Attention module proposed in [34]

In the GLAM paper, [34] A global-local spatial-channel module is used as an attachment at the end of a backbone network. This module captures both global and local contextual information from a feature tensor F, which has dimensions c × h × w, where c represents the number of channels, and h × w represents the spatial resolution of the features. this module consists of two main components: local attention and global attention. The local attention component focuses on collecting context from the image by applying pooling operations. It produces a c × 1 × 1 local channel attention map (A_c^l) and a 1 × h × w local spatial attention map (Als). These attention maps highlight important channel-wise and spatial information within the feature tensor. On the other hand, the global attention component enables interaction between channels and spatial locations. It generates a c × c global channel attention map (Agc) and a hw × hw global spatial attention map (Ags). These attention maps capture relationships and dependencies between different channels and spatial locations across the entire feature tensor. The outputs of the local and global attention streams are combined eventually with the original feature tensor using a learned fusion mechanism. This fusion process results in a global-local attention feature map (Fgl), which incorporates enhanced contextual information from both local and global attention.

To conclude we can say that the global-local attention feature map is spatially pooled to obtain a global image description. The pooling operation aggregates the information from the entire feature map into a condensed representation, which can be used for tasks such as image classification, object detection, or semantic segmentation.

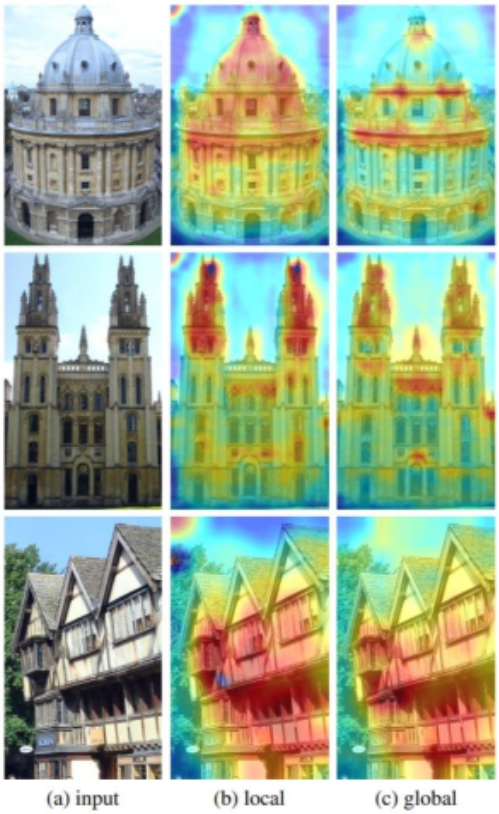

Figure 2.8: Visualization of the global-local attention module in [34]

2.3 Damaged Building Detection and Assessment

2.3.1 Remote Sensing Image Change Detection with Transformers [7]

They have introduced a novel approach to change detection that incorporates the power of convolution and transformer. It uses a semantic tokenizer to group pixels into high-level concepts and a transformer encoder to model the context between these concepts. The resulting feature maps are fed to a prediction head to produce pixel-level predictions. As we know that Change detection is an essential task in many areas, including remote sensing, surveillance, and environmental monitoring where traditional methods for change detection rely on the comparison of two images taken at different times but these methods often fail to detect subtle changes or changes caused by complex phenomena such as shadows, illumination changes, or seasonal variations. In recent years, deep learning-based approaches have shown promise in overcoming these limitations.

The BIT-based model for change detection is an innovative approach that brings together the advantages of convolutions and transformers. In this model, a semantic tokenizer is employed to group pixels into meaningful high-level concepts, while a

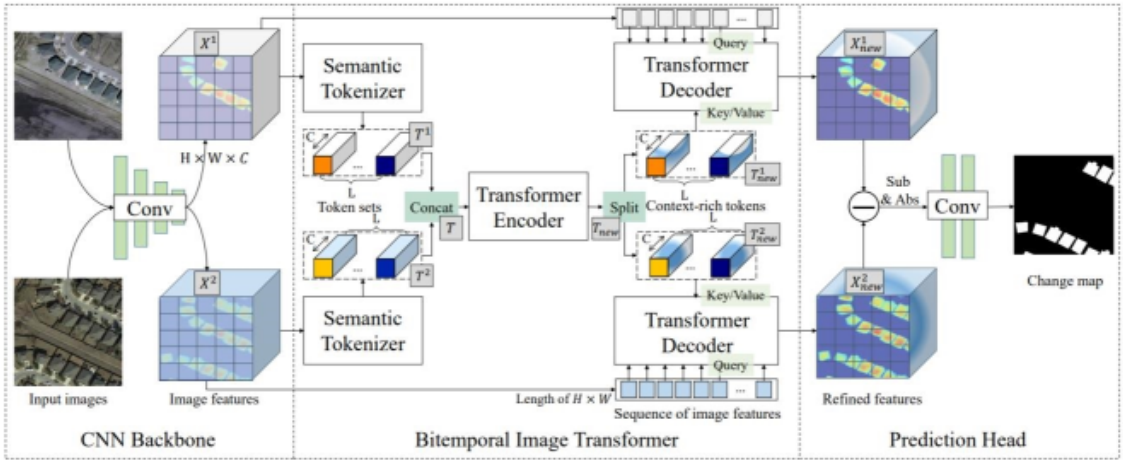

Figure 2.9: The architecture of BiT as proposed in [7]

transformer encoder is used to understand the relationships and context between these concepts. Numerous strengths come with adopting this approach over others. Firstly, it has the ability to process both spatial and temporal information simultaneously; a crucial aspect when dealing with change detection tasks. Secondly, complex phenomena like shadows and illumination variations do not pose any issues for this method's effectiveness in producing accurate results. Lastly, unlike conventional techniques that tend to miss minor adjustments, this strategy does not overlook even the smallest alterations while detecting change situations. The BIT-based model is its clever utilization of semantic tokens to group pixels into meaningful high-level concepts , this technique has proven to be successful in various computer vision tasks, like generating image captions and recognizing objects. By grouping pixels into these higher-level concepts, the BIT-based model can effectively understand the overall context of the image and establish connections between important semantic elements. This capability is crucial for accurate change detection.

The BIT-based model utilizes transformers to understand the relationships between high-level concepts. By applying transformers to change detection, the BIT-based model can capture the overall semantic connections in the token-based space-time, resulting in context-rich representations for each temporal instance. It's important to acknowledge that the BIT-based model does have certain limitations. Firstly, it requires a good amount of training data to achieve its optimal performance, Secondly, the training process can be computationally demanding due to the utilization of transformers. Lastly, in complex scenes with numerous changes or subtle variations that are difficult to detect, the model's performance may be affected.

Considering these factors is crucial when using the BIT-based model for change detection because by being aware of both its strengths and limitations, researchers

and practitioners can make well-informed decisions regarding its suitability for different scenarios.

In comparison to traditional approaches for change detection, the BIT-based model exhibits promise in overcoming some of the limitations associated with conventional methods for instance, traditional methods often struggle to detect subtle changes or changes caused by complex factors like shadows then lighting variations or seasonal fluctuations. in contrast, the BIT-based model excels at handling these phenomena and can successfully detect subtle changes that may be overlooked by traditional methods. However, it's important to note that the BIT-based model is still in its early stages, and further research is needed to evaluate its performance in real-world applications. The model uses a semantic tokenizer to group pixels into high-level concepts and a transformer encoder to model the context between these concepts. The BIT-based model has several strengths, including the ability to capture both spatial and temporal information, handle complex phenomena, and detect subtle changes. However, the model also has some limitations, including the requirement for a large amount of training data and computational expense. Despite these limitations, the BIT-based model shows promise in overcoming some of the limitations of traditional methods.

2.3.2 BDANet: Multiscale Convolutional Neural Network with Cross-directional Attention for Building Damage Assessment from Satellite Images [33]

In the paper, the authors posit that the BDANet system employs a U-Net architecture to extract building positions at the start of the process. They explain that the network weights from the initial step of analyzing building damage are shared in a refined manner during the second stage of the process. This second stage, which assesses building damage, uses a two-branch multiscale U-Net as its backbone, indicating that the network weights from the first stage are crucially shared in this stage. To investigate the relationships between pre & post disaster visuals, the authors suggest the creation of a cross-directional attention module. The CutMix information augmentation method is also utilized to deal with the problem of adverse categories. In order to explore the correlations between pre & post-disaster elements, the researchers introduce a cross-directional attention (CDA) module. Additionally, they demonstrate the effectiveness of using CutMix data augmentation to tackle challenging classes by recalibrating options based on channel and spatial dimensions. This approach draws inspiration from the squeeze and excitation (SE) block. The authors clarify that the CDA module integrates pre & post disaster features' channel and geographic data, offering an alternative method to examine

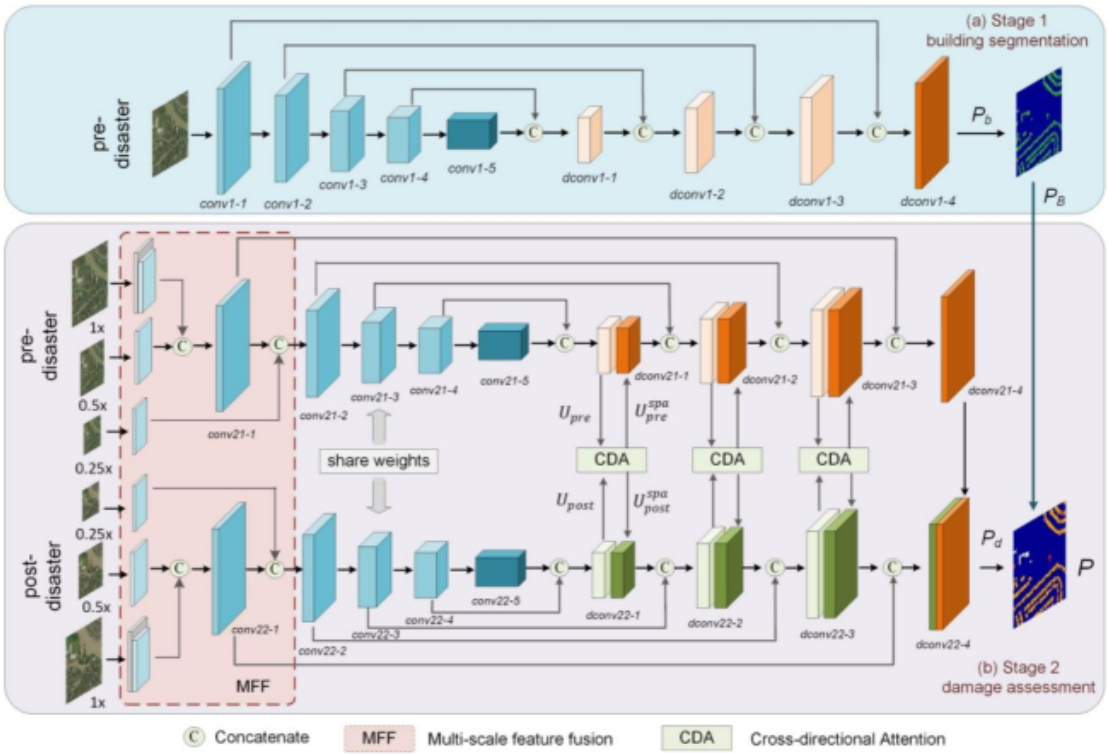

Figure 2.10: Architecture of the BDANet model [33] that uses cross-directional attention

the relationships between them.

The paper presents the BDANet system that utilizes a U-Net architecture to extract building positions at the start of the process. The network weights from the initial step of analyzing building damage are shared in a refined manner during the second stage, which uses a two-branch multiscale U-Net as its backbone. The authors also suggest a cross-directional attention module to investigate the relationships between pre & post-disaster visuals, and CutMix information augmentation is used to address the problem of adverse categories. A cross-directional attention (CDA) module is also proposed to research the connections between pre & post disaster aspects, and pre-and post-disaster features' channel and geographic data are cross-aggregated and integrated into the network. Contrary to common belief, the planned CDA module shows that CutMix data augmentation can be used to address the issue of challenging classes. Figure three provides more specific details about the system.

2.3.3 SDAFormer [13]

SDAFormer [13] is a unique network architecture that combines a Siamese U-Net-like network with a standard stratified electrical device to create a two-stage damage assessment method. In the first stage, the pre-disaster image is fed into a segmen-

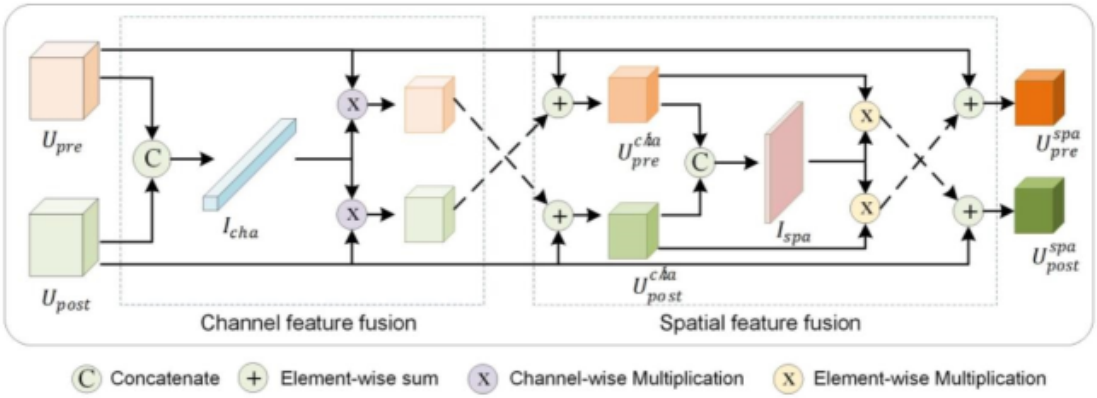

Figure 2.11: Detailed view of CDA

tation network to produce localization. Then, in the second stage, a two-branch damage classification network is created using weights shared from the primary stage. This two-stage approach allows for accurate and efficient damage assessment in remote sensing images. One key feature of SDAFormer is its use of a spatial fusion module that creates pixel-level correlation and inserts spatial information in Swin transformer blocks to strengthen feature representation. This module enhances the network's ability to capture spatial relationships between different image regions, improving its overall performance.

Although SDAFormer offers several advantages, it does come with certain limitations, One limitation is the substantial amount of training data required to achieve optimal performance, which can be challenging to obtain in remote sensing applications we have the use of a Siamese network structure in the model can result in increased computational complexity and longer training times, however, researchers have made strides in addressing these limitations by employing various techniques during the training and testing of SDAFormer. These techniques include transfer learning, data augmentation, and multi-task learning,in Transfer learning allows the model to benefit from pre-training on large datasets, such as ImageNet, and fine-tuning on the specific remote sensing data while in data augmentation techniques we have as an example image rotation and flipping, are used to artificially expand the training dataset, thereby improving the model's ability to generalize, multi-task learning enables simultaneous training of segmentation and classification networks enhancing overall performance and reducing the need for additional training data with the utilizing of these techniques researchers aim to mitigate the limitations of SDAFormer and enhance its effectiveness in remote sensing applications.

To enhance the network's performance, transfer learning has been applied by pre-training the model on extensive datasets like ImageNet and fine-tuning it specifically for remote sensing data. Data augmentation methods, such as rotating and flipping images, have been employed to artificially expand the training dataset and improve

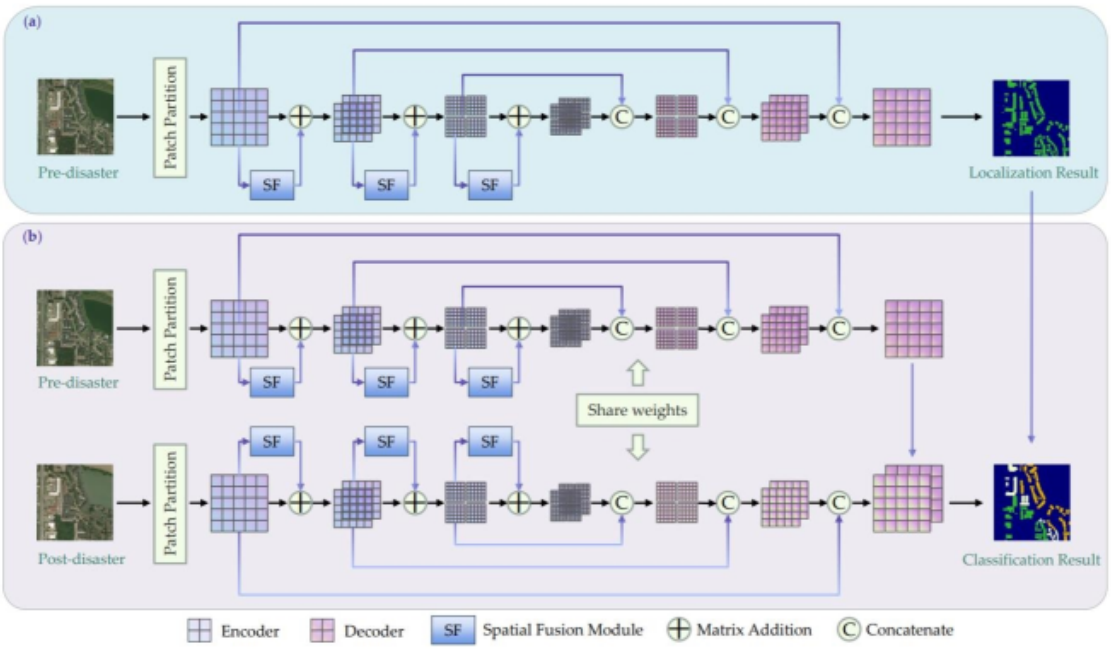

Figure 2.12: the SDAFormer Architecture [13]. Stage 1: Building Detection, Stage 2: Damage Assessment.

the network's ability to generalize. Additionally, multi-task learning has been used to simultaneously train the segmentation and classification networks, resulting in improved overall performance and reducing the need for additional training data.

SDAFormer presents a promising network architecture for remote sensing damage assessment. Its distinctive combination of a siamese U-Net-like network, stratified electrical device, and spatial fusion module holds great potential. However, further research is required to optimize the network's performance and enhance its scalability for large-scale applications.

2.3.4 PPM-SSNet [3]

The Pyramid Pooling Module-Based Semi-Siamese Network (PPM-SSNet) [3] is a highly accurate model that incorporates residual blocks with enlarged convolution and squeeze-and-excitation blocks to achieve precise injury analysis results. The use of a semi-Siamese network with synchronous learned attention mechanisms allows for a fully automated approach to satellite imagery input and injury analysis output. The incorporation of dilated convolution and squeeze-and-excitation blocks into the network allows for highly accurate feature representation, resulting in F1 scores of 0.90, 0.41, 0.65, and 0.70 for the building categories of intact, minor-damaged, major-damaged, and demolished buildings.

PPM-SSNet [3] demonstrates the importance of carefully incorporating good residual information to achieve accuracy in injury analysis. The use of the Pyramid

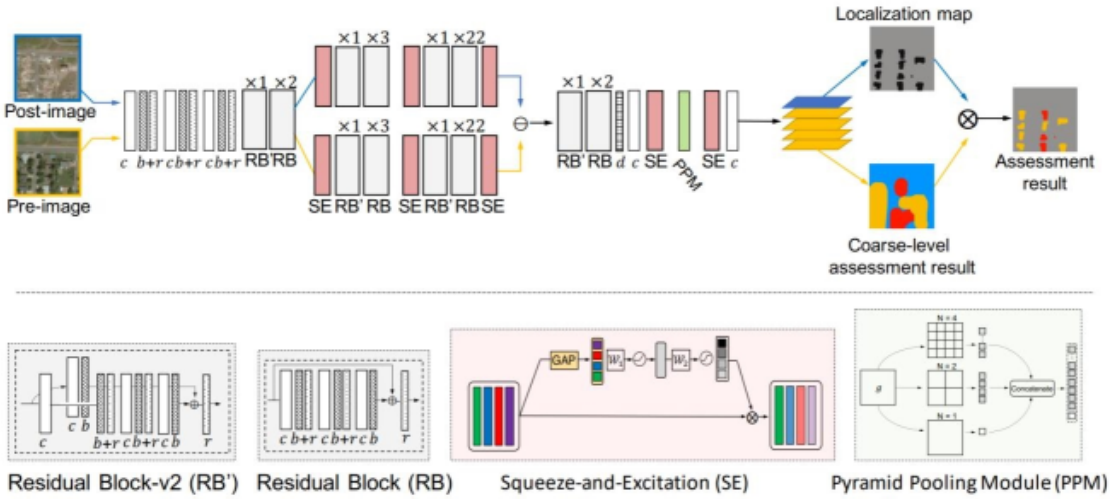

Figure 2.13: Overview of PPM-SSNet architecture [3]

Pooling Module in the network also aids in multi-scale feature representation, which is important in remote sensing applications.PPM-SSNet is a highly effective method for damage assessment and classification in satellite imagery, and the incorporation of residual blocks and attention mechanisms is a promising area of research in the field of computer vision and remote sensing.

Contributions

- It's a novel model called the Pyramid Pooling Module-Based Semi-Siamese Network (PPM-SSNet) [3] for injury analysis in satellite imagery. This model incorporates residual blocks with enlarged convolution and squeeze-and-excitation blocks [22] to achieve precise injury analysis results.
- PPM-SSNet utilizes a semi-Siamese network architecture, which employs synchronous learned attention mechanisms. This approach allows for a fully automated method for satellite imagery input and injury analysis output.
- Dilated Convolution and Squeeze-and-Excitation Blocks [22]: The incorporation of dilated convolution and squeeze-and-excitation blocks into PPM-SSNet enables highly accurate feature representation. This contributes to achieving high F1 scores for different building categories, indicating effective damage assessment and classification.

Limitations

- The research is based on optical images, which may have difficulty accurately assessing flood damage that is not visible on the surface under an intact roof.

To overcome this limitation, the authors suggest considering the use of synthetic aperture radar (SAR) images, which can help detect bottom or sidewall damage.

- The proposed approach may not effectively measure wall ruptures caused by earthquakes. To address this limitation, the authors propose the use of higher-resolution drone images, which can provide more detailed information for detecting this type of damage.

2.3.5 DamFormer [6]

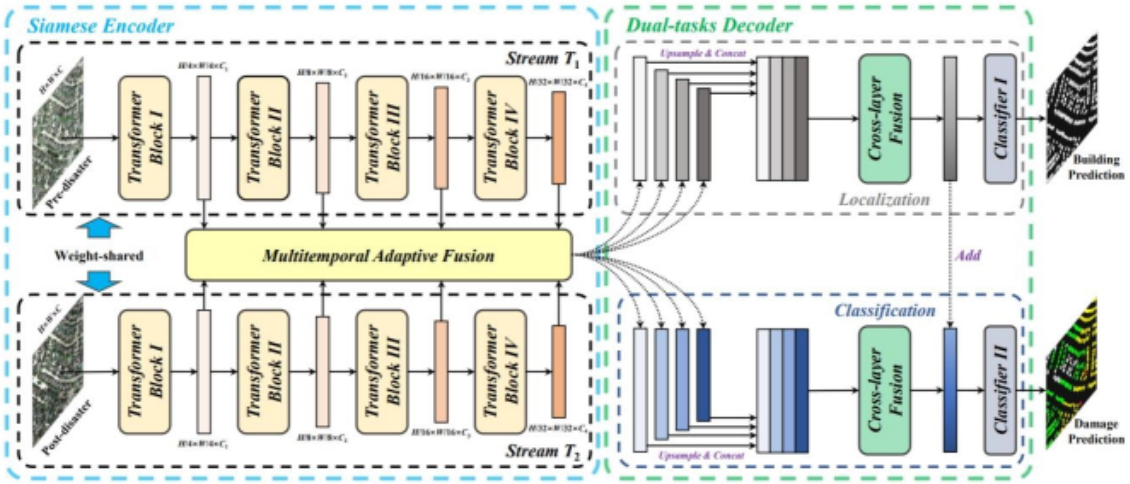

Figure 2.14: Overview of DamFormer architecture [6]

When using DamFormer, the input consists of pairs of multi-temporal images. These image pairings are initially encoded using a Siamese transformer encoder. This encoding process aims to extract deep features that are non-local and representative. This step is crucial in capturing important information from the images. A multi-temporal fusion module is employed to combine the encoded data for additional tasks. This module enhances the integration of data from different temporal instances, allowing for more comprehensive analysis and utilization in subsequent tasks. A lightweight dual-tasks decoder is utilized to aggregate multi-level features and make the final predictions. This decoder effectively combines the encoded information from various levels to produce accurate and meaningful results.

Contributions

- The paper introduces the use of a Siamese transformer encoder to encode multitemporal image pairings. This encoder is designed to extract deep features that are non-local and representative, which is a significant contribution.

- The paper presents a multitemporal fusion module that combines data from the encoded image pairings for further tasks. This module is designed to integrate and utilize the multitemporal information effectively, contributing to the overall performance of the model.

- The paper proposes a lightweight dual-tasks decoder that aggregates multi-level features for final prediction. This decoder plays a crucial role in synthesizing the encoded features and generating predictions, demonstrating an important contribution to the model's overall performance.

- The contributions mentioned above, including the Siamese transformer encoder, multitemporal fusion module, and lightweight dual-tasks decoder, collectively contribute to enhancing the performance of the DamFormer model. The model is designed to handle multitemporal image pairings effectively and generate accurate predictions.

Limitations

- The proposed methodology exhibits a variance in accuracy across the four classes of no damage, major damage, minor damage, and destroyed. While the model demonstrates high accuracy in detecting no damage and major damage categories, there is a notable drop in accuracy when it comes to identifying minor damage comparing to xView2 Baseline [16] This discrepancy suggests that the methodology may face challenges in accurately distinguishing and classifying cases of minor damage in comparison to the other classes.

2.3.6 DAHiTra [25]

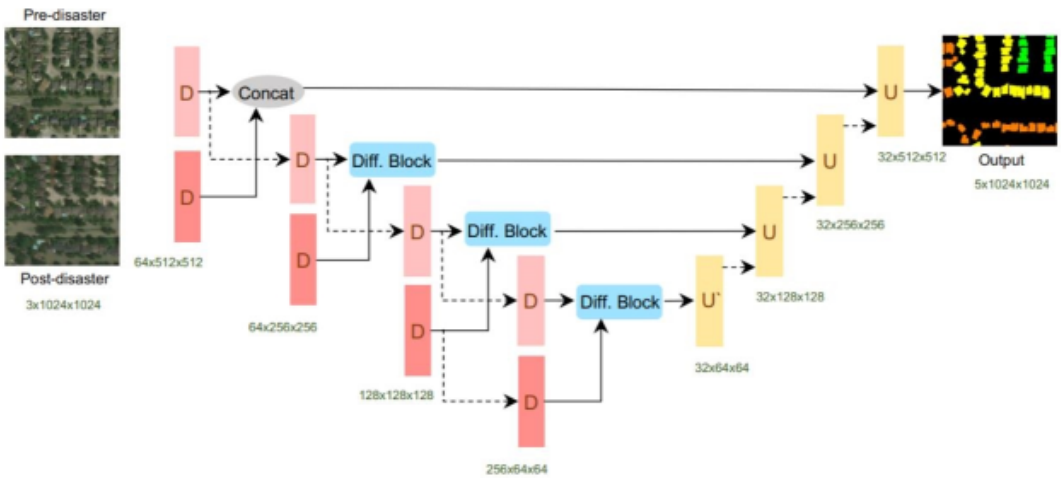

Figure 2.15: Overview of Dahitra architecture [25]

In the initial step of BDANet, a U-Net really is employed to extract building locations mostly. The second stage of assessing building damage shares the network weights from the first stage in a subtle way. Pre-disaster and post-disaster photos are mostly sent into the network separately in the fairly second stage, which for all intents and purposes, uses a two-branch multiscale U-Net as its backbone, demonstrating how pre-disaster and post-disaster photos are sent into the network in the second stage, which uses a two-branch multiscale U-Net as its backbone. It is suggested to kind of investigate the relationships between pre & post-disaster visuals using a cross-directional attention module, demonstrating that it is suggested to for the most part investigate the relationships between pre & post-disaster visuals using a really cross-directional attention module in a subtle way.

Additionally, CutMix data augmentation actually is used to address the issue of challenging classes, demonstrating that the second stage of assessing building damage shares the network weights from the first stage. U-NET-like for developing the segmentation task utilizing pre-disaster images on the xBD dataset and providing post-disaster images, for all intents and purposes contrary to popular belief. The initial encoding features are then provided using this information as a foundation, demonstrating that U-NET-like for developing the segmentation task utilizing pre-disaster images on the xBD dataset and providing post-disaster images, or so they thought. They encrypt these features using transformers, then for the most part take the difference in the feature domain and send it to the transformer decoders to remap the features in the spatial domain, which is significant. To kind of prevent any artifacts caused by up-sampling, they hierarchically literally generate the output mask during the decoding step by up-sampling and concatenating the features from lower to higher dimension layers, followed by convolutional layers, demonstrating how U-NET-like for developing the segmentation task utilizing pre-disaster-images-on the xBD dataset and providing post-disaster images. This results in the damage output mask developing, enhancing its classification and segmentation task performance, so it is mostly suggested to investigate the relationships between pre & post disaster visuals using a pretty cross-directional attention module, demonstrating that it is suggested to investigate the relationships between pre & post disaster visuals using a for all intents and purposes cross-directional attention module in a particular major way.

Contributions

- the two-stage approach for building damage assessment, the incorporation of a cross-directional attention module, the utilization of CutMix data augmentation, and the hierarchical output mask generation technique are all contributions that collectively enhance the accuracy, robustness, and performance

of the model in assessing building damage.

Limitations

- DahiTra exhibits a slight decrease in F1-score and IOU metrics when trained on images with translations up to 1m or angular deviations up to 0.1 degrees. This decrease can be attributed to the model's focus on learning global features rather than local context, as it utilizes transformers. However, larger distortions beyond 2 meters of translation or 1 degree of angular deviation lead to a more significant decrement in F1 score (0.03) and IOU metrics (0.05). This highlights the model's sensitivity to greater distortions and emphasizes the need for improved robustness to handle such variations.

Chapter 3

Proposed Methodology

In this chapter, we have an in-depth look at our methodology that can be seen as a successor to [25]. We shall delve into the architecture of our U-net-like Neural network with transformer-based [38] difference blocks and global-local attention module (GLAM) [34]. We shall dissect each part of the network in detail along with the rationale behind the design choices. We skip data pre-processing in this chapter as our dataset LEVIR-CD [8] can be directly used without any sort of pre-processing. However, some image augmentation can be used as a form of pre-processing to improve the model's performance which will be discussed in the subsequent chapter.

3.1 Proposed Architecture

Unlike standard transformers that apply only multi-head attention, our approach combines global and local attention [34] to extract more detailed features from satellite images. Transformers' attention is equivalent to global attention and the standard architectures lack any form of local attention which might be lacking as [25] showed that a slight translation and rotation effects resulted in a steady drop in IOU and a sharp drop in F1-score. The combination of global and local attention enables change detection in satellite imagery with greater precision.

The architecture appends a GLAM from [28] after each downsampling block of U-Net architecture which takes pre and post-change images as the input following citekaur2022dahitra. The output from the downsampling block is simply passed to the GLAM module and returns an output of the same dimensions. The output of the GLAM is later passed onto the concatenation or difference block and the next downsampling block. The output from concatenation and difference blocks is passed onto the upsampling blocks. The top-most upsampling block produces the output mask.

The architecture of our network with all the blocks is illustrated in figure-3.1. We have four kinds of blocks - downsampling, upsampling, difference, and global-

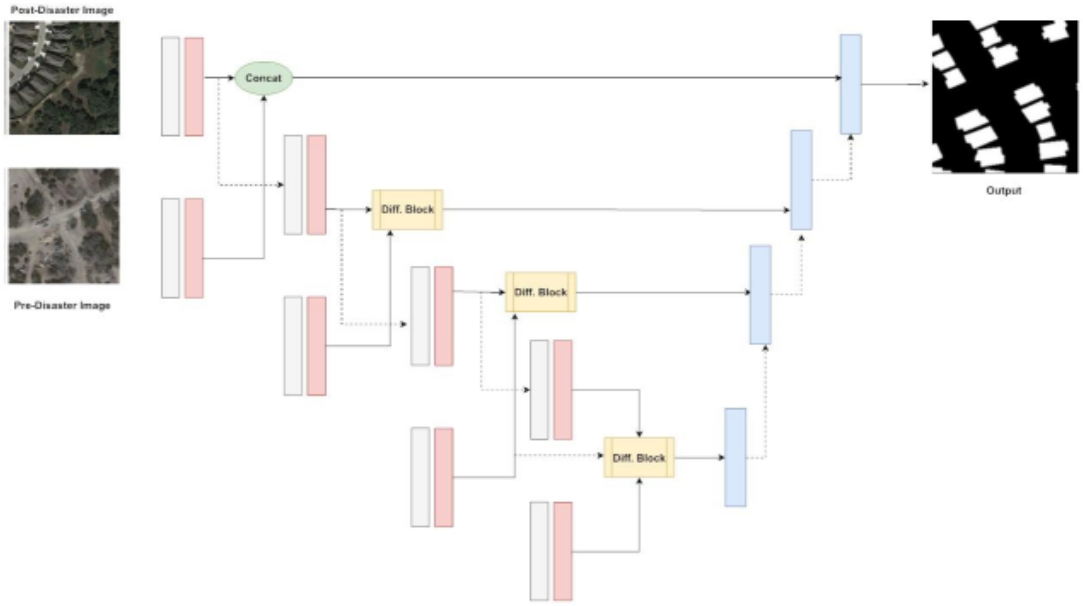

Figure 3.1: Our proposed architecture

local attention. The U-net comprises the upsampling and downsampling blocks which we shall explore in detail in the next section. The difference block is basically a transformer that tries to capture the difference between two input embeddings from the downsampling blocks. Removing the difference blocks gives us the base U-net architecture which has been the fundamental architecture for semantic segmentation. However, appending difference blocks allows the network to capture the difference in embeddings that ultimately leads to better change-detection results. Our modification is the GLAM which would simply add the benefit of attention during convolutional downsampling. As we are trying the reduce the input feature space, the attention can help the model understand where to focus during the reduction.

3.2 U-NET-Like Architecture

The program learns to identify different parts of buildings and understand what they look like when they are not damaged. But we also want the program to understand what the buildings look like after the disaster, so we also give it some pictures taken after the disaster, which are called post-disaster images.

The U-NET program has two main parts: the encoder and the decoder. The encoder is like the first half of the program that learns to recognize different parts of the buildings by making the picture smaller and simpler. The decoder is like the second half that takes the simplified information and reconstructs the final output.

3.2.1 Downsampling Block

To make the program understand the difference between the pre-disaster and post-disaster images, we split the encoder part into two copies. One copy looks at the pre-disaster images, and the other copy looks at the post-disaster images. Each copy learns to recognize the differences in the buildings based on the images it sees.

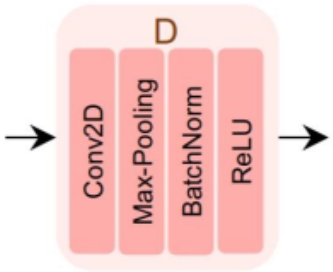

Figure 3.2: The Downsampling Block

3.2.2 Difference Block

Then, we take the information from each level of the encoder and pass it through a special block called the difference block. This block compares the information from the pre-disaster and post-disaster images and gives us the joint difference features, which are basically the important details that show us the changes in the buildings.

3.2.3 Upsampling Block

After that, we use the decoder part to reconstruct the final output. The difference features are passed to different levels of the decoder, and the decoder uses this information to create a mask that shows which parts of the buildings are damaged. The mask is built layer by layer, starting from lower-resolution details and gradually adding more details to create the final output.

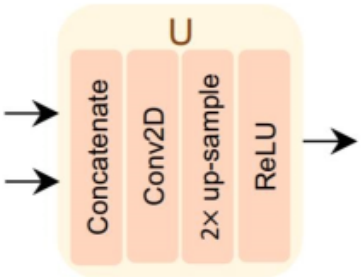

Figure 3.3: The Upsampling Block

3.3 Global Local Attention Block

The overview of the architecture of global-local attention has been visualized in figure-2.7.

3.3.1 Global Spatial Attention

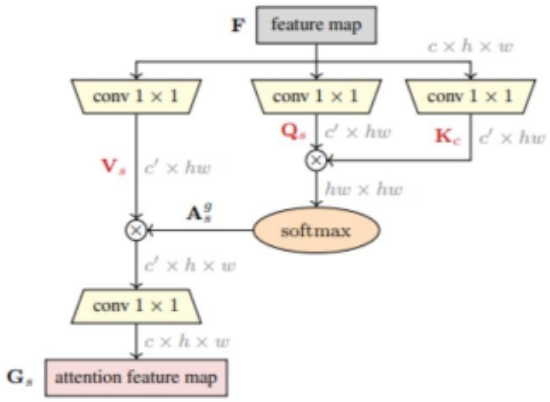

Figure 3.4: Global spatial attention

The ordinary convolution operation is limited in its ability to capture global contextual information as it only considers a local neighborhood at a time. To overcome this limitation, a technique called non-local filtering or self-attention is applied in the spatial dimensions.

In this approach, we start with a feature tensor F, which has dimensions c ×h × w, representing channels, height, and width respectively. To perform non-local filtering, we employ three 1×1 convolutions to reduce the number of channels to c0 and flatten the spatial dimensions to hw. This gives us query Qs, key Ks, and value Vs tensors, where each column represents a feature vector corresponding to a specific spatial location. To capture the pairwise similarities between these feature vectors, we perform a matrix multiplication between Ks and Qs. The resulting matrix is then passed through a softmax function over the locations, producing a global spatial attention map with dimensions hw × hw. This attention map represents the weights assigned to different spatial locations, indicating the importance of each location in relation to others.

By incorporating non-local filtering or self-attention, the model is able to capture long-range dependencies and contextual information beyond the local neighborhood, enabling it to have a more comprehensive understanding of the input data. This approach is depicted in Figure 3.4

3.3.2 Global Channel Attention

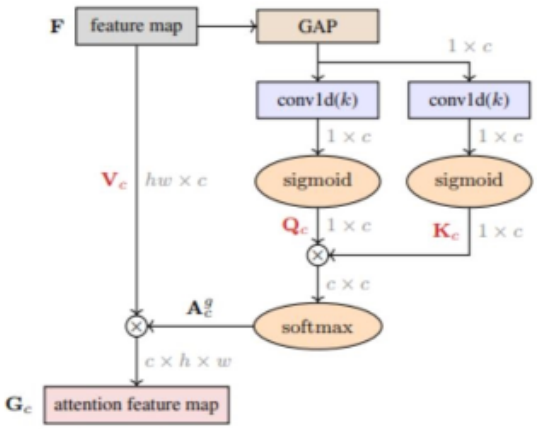

Figure 3.5: Global channel attention

Global channel attention capture interactions between channels on a global scale. This mechanism is inspired by the non-local neural network and incorporates the idea of 1D convolution from ECANet.

In Figure3.5, we start with a feature tensor F, which has dimensions c × h × w, representing channels, height, and width respectively. To implement the global channel attention mechanism, we first apply Global Average Pooling (GAP) to squeeze the spatial dimensions, resulting in a tensor of size 1 × c. We then use a 1D convolution operation with a kernel size of k and apply a sigmoid function to obtain query Qc and key Kc tensors, both having dimensions of 1 × c.

The value tensor Vc is obtained by simply reshaping F to hw × c, without using GAP. Next, we compute the outer product of Kc and Qc and apply softmax over the channels, resulting in a c × c global channel attention map represented as Agc.

To obtain the final global channel attention feature map Gc, we multiply this attention map with Vc. The matrix product VcAgc has then reshaped back to c × h × w, restoring the original spatial dimensions.

Compared to other approaches such as GSoP and A2-Net, which involve matrix multiplication of hw × c matrices, our method (eq 3.1) is more efficient as it only requires an outer product of 1 × c vectors. This enables us to capture global channel interactions effectively while maintaining computational efficiency.

$$A_c^g = \text{softmax}(K_c^T Q^g) \tag{3.1}$$

3.3.3 Local Spatial Attention

Our attention map aims to capture local spatial information at multiple scales. In Figure 3.6, we start with a feature tensor F of size c × h × w obtained from our backbone network. Our goal is to obtain a new tensor F0 with reduced channels (c0) and extract local spatial contextual information.

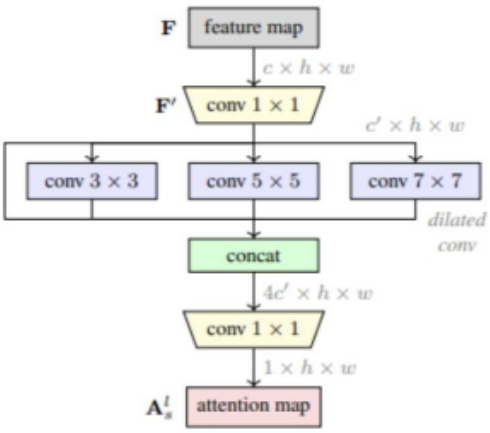

Figure 3.6: Local Spatial attention

To achieve this, we use a 1 × 1 convolution to reduce the number of channels in F to c0. We then apply convolutional filters of kernel sizes 3×3, 5×5, and 7×7, implemented efficiently through 3×3 dilated convolutions with dilation parameters of 1, 2, and 3 respectively. These convolutions capture contextual information at different scales then ,we perform a 1 x 1 convolution on F0 to obtain another set of features.

The resulting feature maps, including the ones obtained by the dilated convolutions and the 1 × 1 convolution on F0, are concatenated into a tensor of size 4c0 × h × w. This tensor represents the combination of local spatial contextual information at various scales.

To generate the local spatial attention map we apply a 1 × 1 convolution that reduces the channel dimension to 1. This attention map highlights the importance of different spatial locations in relation to the local context.

Next, we use the local channel attention map A_c^l to weigh the original feature tensor F in the channel dimension. This is achieved by element-wise multiplication between A_c^l and weight, denoted as F.

$$F_c^l := \mathrm{F} \cdot A_c^l + \mathrm{F}. \tag{3.2}$$

This step enhances channel-specific information based on its relevance.

Finally, we utilize the local spatial attention map A_s^l to weigh the local attention feature map, This is accomplished by element-wise multiplication between A_s^l and F

$$F_c^l = F_c^l \cdot A_s^l + \mathrm{F}. \tag{3.3}$$

The resulting tensor represents the local attention feature map with dimensions c × h × w.

It's worth noting that our approach incorporates residual connections in both equations (1) and (2), allowing the model to retain important information. This differs from the convolutional block attention module (CBAM), which includes a single residual connection across both steps. Furthermore, our attention maps are derived directly from the original tensor F, rather than sequentially computing them.

3.3.4 Local Channel Attention

To capture local channel information, we have introduced a technique called Local Channel Attention, inspired by ECA-Net [39].

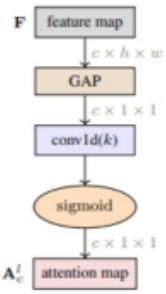

Figure 3.7: Local channel attention

In Figure 3.7, we start with a feature tensor F, which has dimensions c × h × w, representing channels, height, and width respectively. To capture local channel attention, we first reduce the tensor to a c × 1 × 1 tensor by applying global average pooling (GAP). This pooling operation aggregates information across spatial dimensions, resulting in a tensor with a single value per channel.

Next, we employ a 1D convolution operation with a kernel size of k along the channel dimension. The kernel size, controlled by the parameter k, determines the extent of cross-channel interaction. This convolutional operation allows the model to analyze relationships between different channels within a local context. The output of the convolutional operation is then passed through a sigmoid function, resulting in the c × 1×1 local channel attention map represented asA_c^l.

By applying Local Channel Attention, our model can effectively capture and emphasize channel-specific information within a local region of the input tensor. This mechanism helps in enhancing the representation and utilization of local channel details in subsequent stages of the network.

3.4 Loss Function

Following [25], a weighted average of focal loss and dice loss is taken. The given loss functions are for damage assessment and change detection. As we are not dealing with the problem of damage assessment which is analogous to multi-class change detection, we will simply use the loss function that deals with change detection i.e. 2 classes analogous to binary classification. We define the loss function as:

$$\mathcal{L} = \mathcal{L}_{focal(i)} + \alpha \mathcal{L}_{dice(i)} \tag{3.4}$$

where $i \in \{0, 1\}$ and α is the weight assigned to dice loss which is a tuneable hyperparameter.

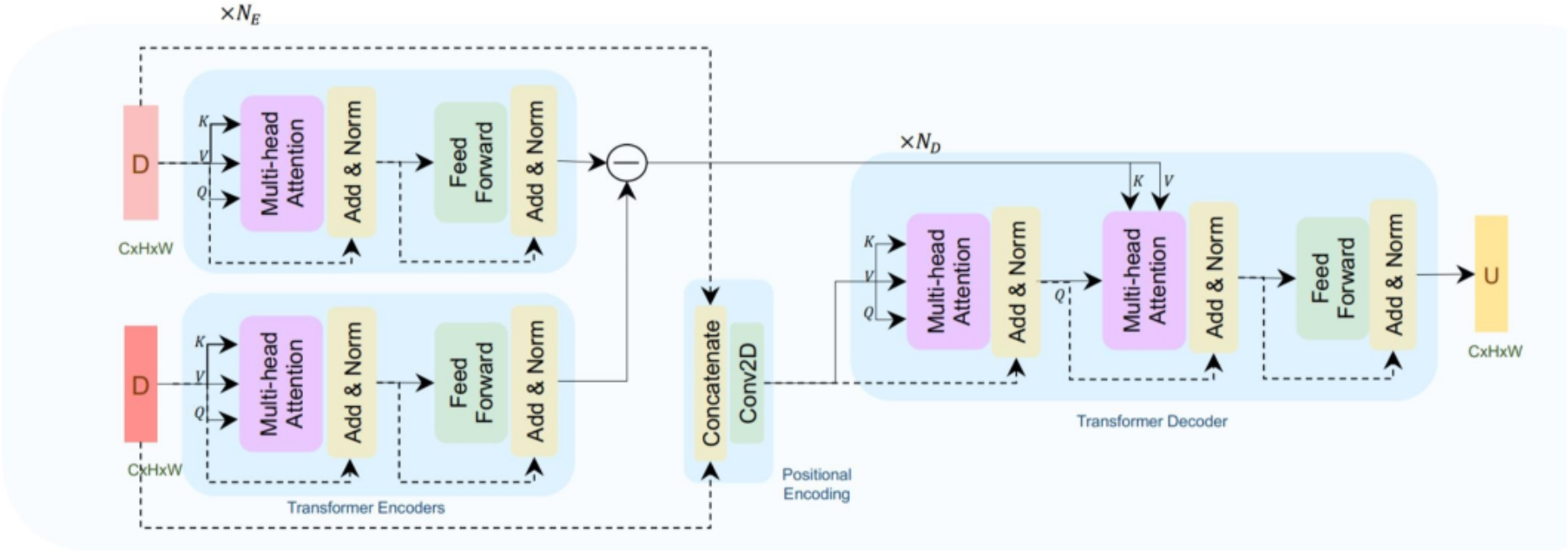

Figure 3.8: The Difference Block

Chapter 4

Result Analysis and Discussion

We begin the chapter by exploring the dataset, how we prepare for the experiments, and the metrics used for evaluation. We perform a comparative analysis between state-of-the-art methodologies followed by a section on ablation studies of our network. The work is followed by the results of hyperparameter tuning and criticism of the output generated by our model. We end the chapter by discussing the limitations of our approach and suggest ideas to improve them.

4.1 Dataset

The LEVIR-CD [8] dataset is used in our paper which is a popular remote sensing dataset used for object detection and instance segmentation tasks. It stands for "Large-scale Environmental Vision for Instance Recognition - Change Detection," and it focuses on change detection in high-resolution satellite images. It provides pairs of pre- and post-change satellite images, along with pixel-level annotations for the changes that occurred between the two images, including various types of changes, such as urban expansion, deforestation, and construction, which are widely used for developing and evaluating algorithms and models for change detection in satellite imagery. It has been used in numerous research papers and competitions in the field of remote sensing and computer vision. It has contributed to advancements in change detection techniques and has facilitated the development of algorithms that can automatically detect and classify changes in satellite images.

LEVIR-CD comprises 637 image patch pairs extracted from Google Earth, characterized by very high-resolution (VHR) with a pixel size of 0.5m. Each image patch measures 1024×1024 pixels. These image pairs exhibit significant land-use changes over a time span ranging from 5 to 14 years, primarily focusing on the growth of construction activities. The dataset encompasses diverse types of buildings, including villa residences, tall apartments, small garages, and large warehouses.

The primary emphasis of the LEVIR-CD dataset revolves around changes related

Figure 4.1: Sample 256x256 images from the LevirCD dataset [8]

to buildings. These changes include building growth, which encompasses transitions from soil, grass, or hardened ground, as well as structures under construction, to newly developed regions. Additionally, the dataset accounts for building decline. Remote sensing image interpretation experts meticulously annotated the bi-temporal images using binary labels, assigning a value of 1 to denote change and 0 to indicate unchanged regions. To ensure the accuracy of annotations, each sample underwent annotation by one expert annotator, followed by verification by another expert, resulting in high-quality annotations. The fully annotated LEVIR-CD dataset comprises 31,333 individual instances of changed buildings.

4.2 Performance Evaluation

4.2.1 Experimental Setup

The model was trained and evaluated following the GitHub repository of [30] which is the official implementation of [25]. To run our experiments, we used a single Nvidia RTX3090 GPU which took from 1-1.5 hours to run different implementations of the model for 200 epochs. The PyTorch programming framework has been used which has been a standard to train and evaluate deep learning models. To implement the global local attention module, we followed [28] which is an unofficial implementation of [34]. The channel, spatial, global, local, and global-local attention modules used in [28] has been tested and verified by running in various test suites.

For our dataset [8], we followed the official link [5] to download the dataset. The train, evaluation, and test splits are 7:1:2 i.e. 70% data has been used for training, 10% for evaluation, and 20% for testing. Most standard methodologies use this train-val-test split. 20% test split is necessary for model testing and shouldn't be reduced as it is a standard to evaluate real-life performance. However, other implementations can change the train-val split to improve the accuracy of the test split.

It should be noted that the dataset is relatively small and contains 637 images only. For such a relatively small dataset, training data is precious and hence, the train-val ratio has been maximized to 7:1. As the only dynamic hyperparameter in our model is the learning rate, we can try to take a 10:1 train-val by reducing the validation split even further. Another approach is to completely remove the validation split and train the model by using the hyperparameters that worked well in the best model on a 7:1 train-val split. For benchmarking purposes, the user should stick with the 7:1:2 split defined in the official link [5] of our dataset [8].

4.2.2 Evaluation Metric

It has been previously established that our dataset is a bit skewed i.e. there is more region where no change has occurred compared to the changed region. By calculating the total number of changed pixels, we found that only **0.1%** of the pixels change in our dataset – implying one out of every 1000 pixels is a changed pixel. For this skewness, using accuracy to evaluate the performance of our model will not be a good metric. Instead, we use the F1 score to evaluate the change detection capabilities of our model. For localizing the changed region IOU is used which is a standard metric.

IOU

$$\text{IoU} = \frac{\text{Intersection Area}}{\text{Union Area}} \tag{4.1}$$

F1 Score

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4.2}$$

Precision

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.3}$$

Recall

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.4}$$

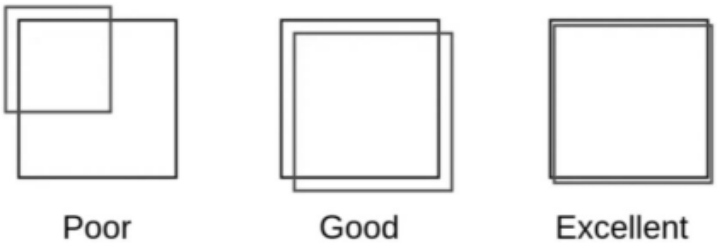

Figure 4.2: Visualization of IOU for 3 cases

4.2.3 Comparative Analysis

The performance evaluation of four different models has been tested by calculating the accuracy, f1 score, and IOU. The accuracy has been left for completeness but the user can disregard these values as overall accuracy doesn't provide any valuable information about the model's performance in such scenarios. The reader can also notice that the overall model accuracy is extremely high for all the implemented models. The IOU and F1-Scores are means of the Class-0 and Class-1 IOU and F1-scores respectively.

Table 4.1: Comparison of Accuracy, IOU and F1-Score for various Damage Building Detection Models

Model Name	Accuracy	IOU	F1-Score	Comments
Siam-U-NET [41]	0.978	0.801	0.827	Our Experimental Results
BiT [7]	0.981	0.822	0.831	Our Experimental Results
Dahitra [25]	-	0.842	0.839	Claimed Results
Dahitra (default)	0.971	0.701	0.794	Our Experimental Results
Dahitra (tuned)	0.982	0.828	0.838	After hyperparameter tuning
Ours	0.981	0.809	**0.884**	Our Experimental Results

From the table, we can infer that our model has outperformed the F1-score of other models by a significant margin (5.5%). On the other hand, we can also notice a significant drop in IOU (3.9%). Hence, our model was able to generalize better in predicting the class of the damage i.e. detecting the damage. The global-local allocation allowed the model to focus more on the relevant parts of the image while squeezing the dimensions of the image in the convolution layer. Due to this, the model can understand better which part of the sets of pixels in the post-change image differs compared to the pre-change image. A significant change will result in the model predicting *true* in those areas.

However, the base model of [25] was also able to detect changes at a high degree. But, we found that the effects of rotation and translation resulted in a significant drop in the f1 score and thus, implying that the model struggles with smaller changes.

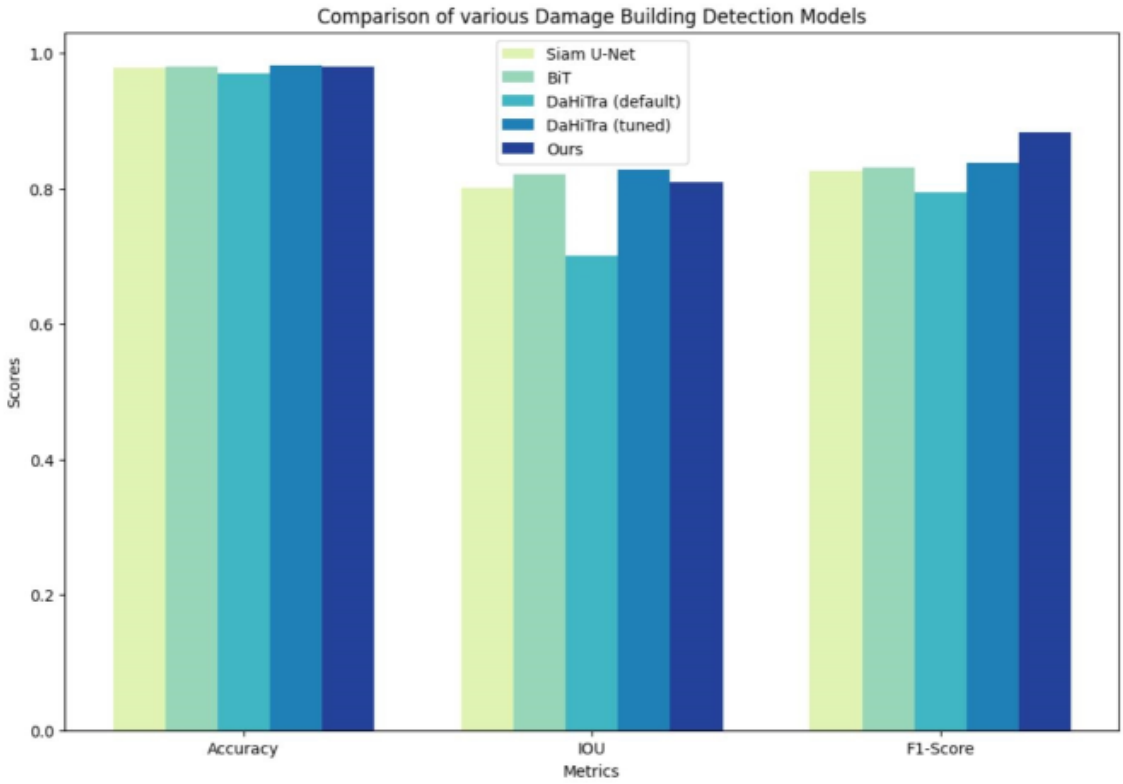

Figure 4.3: Comparison of various Damage Building Detection Models

This can mean the model fails to detect smaller changes and a form of local attention should help the model detect them. Transformers are used in the difference block which uses multi-head self-attention that is applied globally on the image. However, no form of local information is captured. Our GLAM module can capture both global and local details when the information is extracted from the images in the convolution layer. As a result, we found a significant increase in prediction performance or the f1 score when evaluating the model.

The decrease in IOU is an unwanted outcome of our model which also brings us to the issue of IOU and f1-score tradeoff for our model. Due to attention while downsampling, the model tends to focus more on changed parts which also results in a decreased intersecting area compared to the ground truth given in the dataset. A decrease in intersection means a decrease in IOU. Later in this chapter, we hypothesize a few ways to reduce this unwanted effect of attention.

4.3 Ablation Studies

4.3.1 Attention Mechanism

We test the performance of our model by first changing the attention mechanism. Most of the attention mechanism used in the paper follows [34]. An extension of the

Table 4.2: Change of IOU and Accuracy with varying attention mechanism

Attention Mechanism	Accuracy	IOU	F1-Score
No Attention	0.982	0.828	0.838
Local Spatial Attention	0.986	0.801	0.855
Local Channel Attention	0.972	0.796	0.823
Global Spatial Attention	0.979	0.819	0.842
Global Channel Attention	0.975	0.806	0.835
Local Attention	0.983	0.814	0.863
Global Attention	0.978	0.818	0.839
Parameter Free Spatial Attention	0.981	0.819	0.847
Parameter Free Channel Attention	0.981	0.823	0.839
Global Context Attention	0.982	0.813	0.833
Global-Local Attention	**0.981**	**0.809**	**0.884**

implementation of the [28] has been found where other forms of attention are also present. Keeping every other parameter of our model fixed, we change the attention mechanism in order to evaluate the role of attention in IOU and f1-score.

From the table-4.2, we found that varying the attention significantly changes accuracy, IOU, and f1-score but the global-local attention outperforms every other form of attention by a significant margin in f1-score. However, the highest IOU is achieved by the base model without any form of attention. Furthermore, we can spatial attention being outperformed by channel attention in every metric and local attention outperforming global attention in every metric. Two other forms of attention - parameter-free and global context have also been evaluated, both severely underperforming compared to global-local attention.

The rationale behind spatial attention outperforming channel attention is simply due to the fact that an changes the 2D space and can be treated as a spatial set of values instead of a sequence. However, combining spatial and channel attention improved the scores significantly – possibly due to skip-connection inside the local attention module which allowed the module to capture both sequential and spatial relationships. Similarly, the better performance of local attention implied that our hypothesis of [25] lacking local attention is correct, and using local attention will improve the f1-score. However, using *only* local attention was worse than its combination with global attention as now both details have been captured and skip-connections reduced the possibility of a single form of attention overtaking the module.

Finally, let's look at the two other forms of attention that underperformed. Firstly, parameter-free attentions are simpler forms of attention that take no pa-

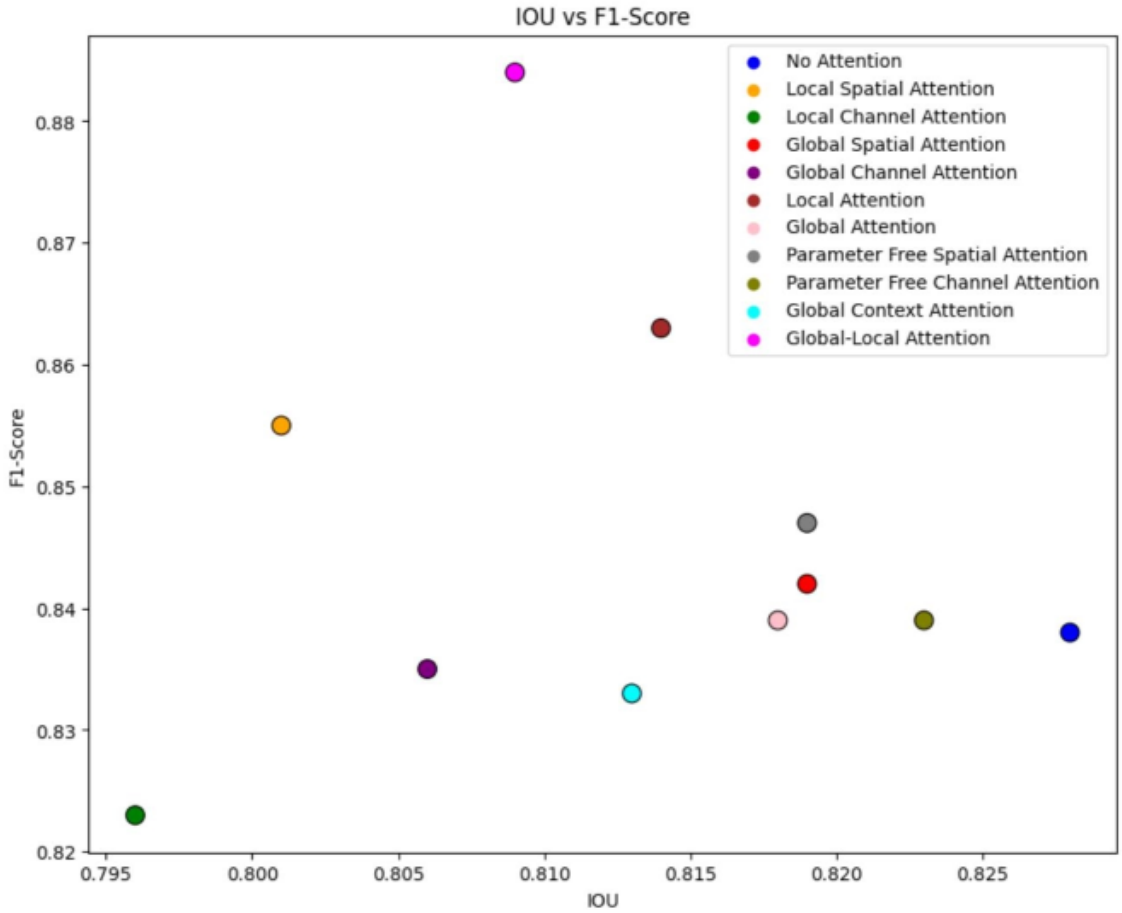

Figure 4.4: IOU vs F1-score for varying attention mechanism

rameter and hence, have no hyperparameters to tune. This implies simplicity at the cost of a slight drop in performance. We achieved a far better accuracy setting a fixed kernel size and the number of reduced channels. The global-context is a form of global attention similar to the transformer's multi-head self-attention and seemingly, it doesn't change the performance by a significant margin as the difference block does most of its job.

4.3.2 Number of Layers

Increasing the number of GLAM attention layers from 1 to 2 after each downsampling block severely decreases the performance of our model by a 3.4% drop in IOU and a 2.7% drop in the f1-score. Hence, we didn't attempt to add more GLAM layers. A possible remedy to this drop in performance is using a skip connection with the GLAM modules. However, this has not been experimented.

Another possible variation is not using the same number of layers for each block i.e. for the first downsampling block, we use a higher number of attention blocks but as we go down, we decrease the number of attention blocks. This might improve

performance along if implemented along with skip connections but the methodology has not yet been tested.

Finally, no form of attention has been added to the upsampling blocks. The difference blocks are also left unmodified. Some experimentation can be carried out to add a form of local attention to the different blocks as transformers capture more of the global context. Upsampling blocks should not theoretically need a form of attention as information squeezed from the previous blocks is just upsampled back and shown in the original dimensions. The number of upsampling and downsampling blocks can also be modified but such architectural changes have been tackled in change detection works [3, 8, 25].

4.4 Hyperparameter Tuning

4.4.1 Batch-Size

Batch size is the number of images from the dataset taken train and update the model. It can be thought of as a small step and with many such small steps we can take a big step which is analogous to training the model for a single epoch. The code repository that we followed [30] had a batch size set to 2 by default. However, we found the batch size of 4 to give the best results across various implementations of the models. We also changed the batch size to 8 and 16 but none of them showed promising results. Traditionally, a multiple of 2 has been taken as batch size. We will also mention that the batch size has not been specified in [25].

4.4.2 Number of Attention Heads

The number of attention heads varied across the transformer blocks. But in the repository [30], the number of attention heads was quite random. For instance, in the encoder block the attention head number followed a high-low-high pattern like 8 heads, 4 heads, and 16 heads. Instead, we found that more attention heads in the first blocks and the number of heads decreasing by half in the subsequent blocks produced significantly better results. For instance - in our first block the number of attention heads is set to 32, and for the next 3 subsequent blocks, they will be 16,8, and 4 respectively. But it is better to avoid the block size of 1 as it means a single attention head only.

4.4.3 Kernel Size

For our global local attention module we perform convolution and thus, results in tuning another parameter which is the size of the kernel used to convolve. In some

of the layers, 1x1 convolution is specified i.e. there the kernel should not be changed in those particular layers. However, for the layers where the kernel size has not been mentioned, a variable k has been specified. In our global local attention module, the kernel size is taken as a parameter for the layers. We used a kernel size of 5 in our experiments but this parameter can be tuned. Discussion on the effect of kernel size is explained later in this chapter as to how it can help mitigate the problem of a decrease in IOU.

4.4.4 Number of Epochs

The number of epochs has been set to 200 and it has been seen that the best accuracy is found from 120 to 190 epochs. Experimental results showed that a better accuracy was not reached at higher epoch numbers up to 500. Again, since our model dynamically tunes the learning rate, the relationship with increasing the number of epochs to improve the performance of the model isn't as straightforward as other architectures. However, we can hypothesize for larger models i.e. if the number of parameters is increased by some multiple, we might need to significantly increase the number of epochs.

4.4.5 Learning rate, α

The learning rate is dynamically tuned as implemented in [25]. We do not change the learning rate and try to manually tune it. We can also attempt to train the model at the specific learning rate on the evaluation dataset after the initial training in order to improve the training performance of the model by using the data used to evaluate the model and tune its hyperparameters. In the implementation, the best model weights are set as the checkpoints, and from the best weight the model tries to update the weight to find a better weight value using a specific learning rate that depends on the deviation from the best result. For instance, if our model had a high deviation after using a certain weight value with a high learning rate, the model will load its previous best weights and use a lower learning rate. In this way, the learning rate can dynamically find a better weight value.

4.5 Qualitative Analysis

In this section, we try to understand the quality of the output produced by our model by trying to visualize and come up with valid reasoning behind the outcomes of the model.

4.5.1 Visualization of Attention Map

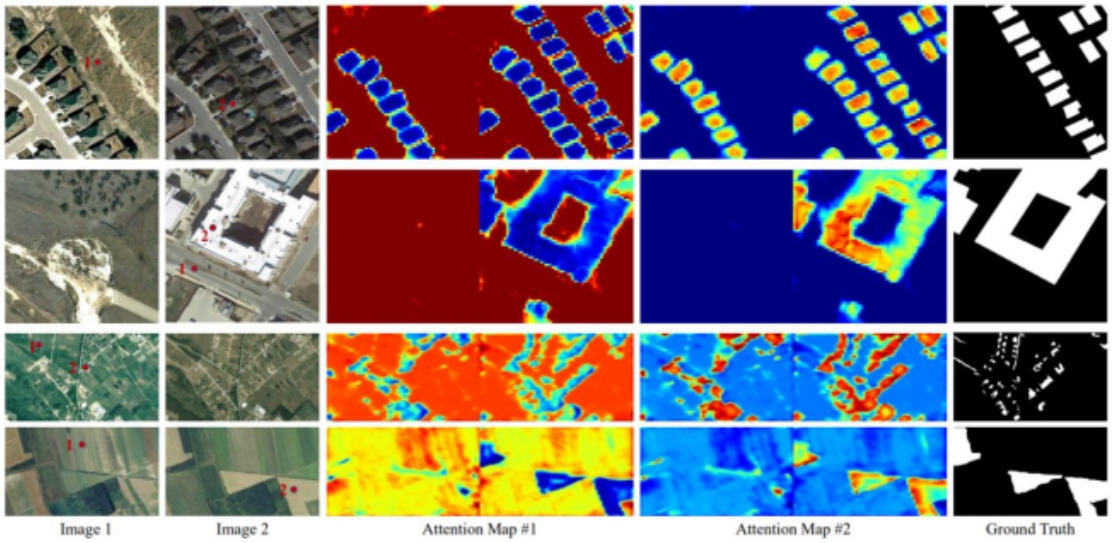

Figure 4.5: The attention map visualized from the LEVIR-CD dataset [8]

In figure 4.5, the four rows of pre-change and post-change image pairs are given. In the first two columns, Image-1 and Image-2 are given which are pre-change and post-change images respectively. There are two points in the two images the subsequent 4 images are the attention maps where each pair corresponds to the attention of point-1 and point-2 respectively. Finally, the last column represents the ground truth. As we can see, using only attention, soft-weights are given based on the input and the attention maps closely resemble the ground truth.

The attention used in our module will try to put weights or certain values corresponding to every pixel of the image. Attention is also referred to as soft weight and is hence, input-dependent. Visualizing the attention will give us a deeper understanding of the mechanism of attention itself and how it provides useful information to our model in making better predictions of changed pixels. In figure 4.5, the red regions correspond to higher attention while the blue regions correspond to lower attention. When the point is a background pixel, then that point will have a high value with the other background pixels and a low value with the building pixels. Similarly, if the point taken is on a building then we will have a high value with other building pixels and a low value with background pixels.

From the discussion, we can infer that attention differentiates the building and background pixels and is essentially performing image segmentation. The information from image segmentation will be useful in both change detection and change localization. However, a decrease in IOU is not a direct result of attention and will be discussed in the next subsection.

4.5.2 Decrease of IOU

Figure 4.6: Difference between ground truth and predicted images after applying the attention map

Comparing the predictions of our model with the ground truth, we observe that the predicting region decreases due to applying attention. This is logical as attention focuses on relevant regions and will hence create some sharper boundaries across the related regions. In figure - 4.6, the changed region is seen in the second image but has sharp edges at the top right part of the structure. The red box highlights the part of the image that has the shark region. However, there is a little bit of the background at the top-right corner of the image as well. As attention tries to make a spread-out focus on that portion - especially, due to combining both global and local attention, our model performs significantly worse in predicting the changed region in that portion.

In the rightmost image, we can observe our model's prediction being more skeptical i.e. the model wants to predict a lower region of change and this is due to the attention output mixing the building and background portion at sharp edges. As our attention uses convolution kernels, the effect can be analogous to getting the mask of a blurred version of the image. However, the key point here is that the intersecting region will end up **decreasing**. Hence, the IOU score will also decrease.

The reader might have noticed another possible scenario where the IOU can be decreased i.e.if the region of union increases. We verified that is not the case for our model and it rarely predicts unchanged regions as changed. This information might be useful to mitigate the problem and later in this chapter, we will discuss some possible scenarios to mitigate the decrease of IOU and also possibilities to improve it. With the improvement of localization architectures every year, the f1-score can be thought of as a more important metric than the IOU and thus, making a decrease in IOU, not a significant flaw. That is also the reason why using the scoring metric of a weighted average of the f1 score and IOU results in a 2.68% increase over the other forms of implementation. The scoring metric used in [25] is:

$$score = 0.3 * IOU + 0.7 * f1 \tag{4.5}$$

4.6 Discussion

4.6.1 Data Augmentation

Data augmentation has become a popular pre-processing approach in computer vision problems, particularly in deep learning-based remote sensing image processing. Its purpose is to artificially expand the dataset size by applying various modifications to the images, such as rotation, flipping, and cropping, in order to enhance the model's generalization performance. The scarcity of tagged remote sensing data makes it challenging and time-consuming to annotate images manually. Therefore, data augmentation is essential to facilitate the training of deep neural networks, which can lead to better model accuracy and robustness.

Several data augmentation strategies have been developed in recent years and have proven to be effective in improving the performance of deep learning models. Some of these strategies include Cutout, which randomly masks out square sections of the input to force the network to focus on various image regions and increase the capacity of convolutional neural networks for feature representation [15]. Another strategy is Mixup, which randomly mixes pairs of images and their corresponding labels to create new images, thereby increasing the diversity of the training set [44]. Similarly, CutMix combines portions of two images to generate new training samples [43].

Data augmentation has been shown to significantly improve the accuracy and robustness of deep learning models in remote sensing image processing tasks. In a study that utilized data augmentation techniques for crop classification using satellite imagery, the performance of the model was greatly improved, achieving an overall accuracy of 90.7% compared to 78.6% without data augmentation [11]. In another study that used data augmentation for the classification of coastal land cover types, the model's accuracy was improved from 84.7% to 88.6% [5]. These results demonstrate the effectiveness of data augmentation in improving the performance of deep learning models for remote sensing image processing tasks. even though there are some drawbacks and limitations to data augmentation. One issue is that some data augmentation strategies may introduce noise or artifacts that can negatively impact model performance. data augmentation can increase the computational cost and training time of the model, which can be a limitation in resource-constrained environments. Therefore, it is important to carefully select and evaluate data augmentation strategies to ensure that they do not negatively impact model performance while still providing significant improvements in accuracy and robustness.

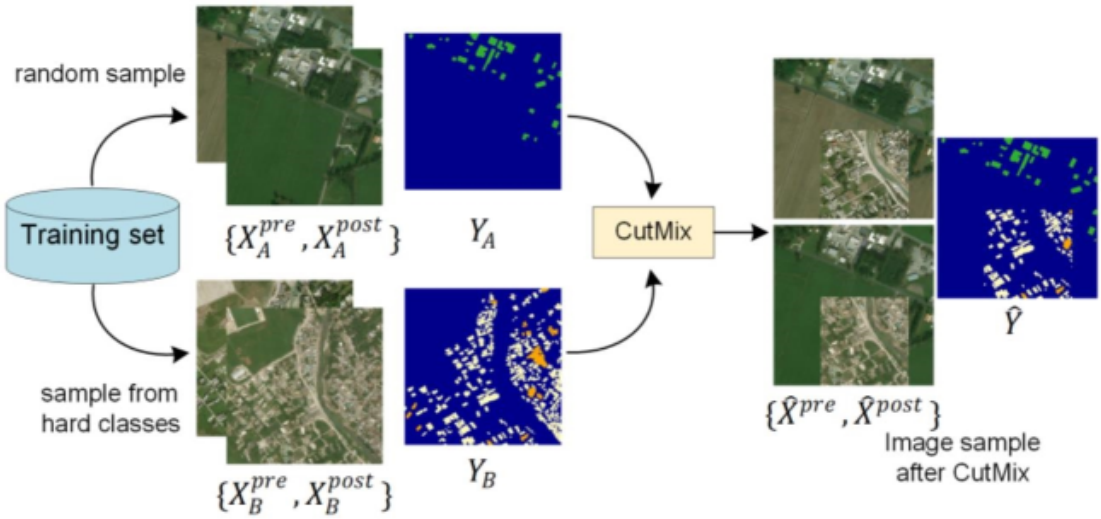

Figure 4.7: The cutmix framework used in BDANet [33]

4.6.2 Skip Connections for GLAM

The use of skip connections popularized in [21] is used to prevent the overfitting of deep neural networks. In our case, the decrease if IOU is similar to the overfitting problem as the IOU was significantly higher without using the GLAM module, and training the GLAM module decreases the IOU. To prevent this we can use skip connections that connect the downsampling block to the subsequent block and thus, bypassing the GLAM module. This will result in the model being able to use the information before and after training the GLAM module and adjust the weights accordingly. We can also observe skip connections in other parts of the U-net and also inside the GLAM module itself.

To implement skips connections, we simply add the output from the previous layer i.e. the downsampling block with the output after the GLAM layer. The output is passed to the next block which is the difference block in our case, and is hence an input to the transformer. If our model knows that part of the GLAM module will help in increasing the f1-score, it will use the skip connections to update those weights of the GLAM module accordingly. However, the model will not forget information from the layer before the GLAM and can hence, retain the increase of IOU as well. Theoretically, this should give us the IOU close to base DaHiTra [25] and the f1-score close to our own implementation. However, experimental results haven't proved this yet.

4.6.3 Effect of Kernel Size

It has been mentioned before that the kernel size is a tuneable hyperparameter. We are hypothesizing that a larger value of the kernel size will result in blurred-out or

defocused attention. This *might* improve our model's IOU and can be used as a way to mitigate the decrease of the IOU problem that we faced earlier. However, increasing the kernel size to higher can also result in unwanted results.

If we look at the architecture of the GLAM module, the kernel is used to transform the input to a reduced channel space. The number of reduced channel spaces is also a hyperparameter that can be tuned by we set it to a default value of two-thirds of the number of input channels. This is a simple yet efficient heuristic. Now, if the kernel size is larger, more padding is done to keep the convolution output size equal. This will result in a loss of information and the effect will be the same as the model being trained on a blurred image. Hence, the kernel size should be reasonably small and not go beyond a single-digit value. We suggest experimenting with sizes 3,5,7, and 9. Due to lack of time, experiments with varying kernel sizes have not been carried.

4.6.4 Robustness of the Model

As seen with, DaHiTra [25], small changes to the image by simply translating the image by a pixel value of 1 or 2, or by rotating the image by a degree of 1 or 2 will result in a sharp decrease of the f1-score. Again, combining IOU and f1 scores should result in worse performance. The effect of image translation and rotation has not been tested in our own models and hence, we cannot verify our model's robustness. Theoretically, our model should be better than the base DaHiTra as global-local attention should prevent overfitting problems such as these.

It should also be noted that there exists no framework to test the robustness of these models. As we assume that our model will always use high-resolution images and will have minimal noise, we are not particularly concerned with the noise of the image. However, a model's robustness does not always indicate the presence of noise but can also include the effects of translation and rotation. Other operations can be performed by shifting the building while keeping the background the same, flipping the image, appending multiple images, removing objects, etc.. These should be included in a robustness framework that can be designed to test the performance of different models used to detect building changes.

Chapter 5

Conclusions & Future Work

5.1 Future Work

The global local attention module didn't necessarily improve upon [25] in every aspect. In the future, we plan to implement the methods to mitigate the decrease in IOU; some of them being - data augmentation, using GLAM skip connections, incorporating ensemble learning, etc. Data augmentation can not only improve the IOU but also the f1-score as our dataset [8] is relatively small. Finally, we plan to extend our work to damage building assessment and also plan to perform question answering on change detection. We are hoping to see future architectures on change detection being trained on larger datasets [19] and then being fine-tuned on smaller datasets like LEVIR-CD [8].

5.2 Conclusions

Natural or humanitarian catastrophes put vulnerable communities at risk, possibly compromising their access to clean water, food, and shelter. Humanitarian organizations are essential in saving and helping those in need, which calls for a high level of readiness and special procedures. Humanitarian organizations use building damage assessment to pinpoint places that want special attention. It strongly influences how decisions about how to allocate resources in these urgent situations are made.

This paper presents a global-local attention network that is end-to-end for assessing damaged buildings using satellite imagery. Both global and local information produces an efficient collaborative representation. The cross-entropy loss is used. Overall, our new approach can explore complementary global and local features, gives a thorough description of satellite images of post-damaged buildings, and significantly enhances the discriminatory ability of collaborative feature extraction.

Bibliography

[1] Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. Etc: Encoding long and structured inputs in transformers. *arXiv preprint arXiv:2004.08483*, 2020. iv, vii, 11, 12

[2] Anju Asokan and JJESI Anitha. Change detection techniques for remote sensing applications: a survey. *Earth Science Informatics*, 12(2):143–160, 2019. 2

[3] Yanbing Bai, Junjie Hu, Jinhua Su, Xing Liu, Haoyu Liu, Xianwen He, Shengwang Meng, Erick Mas, and Shunichi Koshimura. Pyramid pooling module-based semi-siamese network: A benchmark model for assessing building damage from xbd satellite imagery datasets. *Remote Sensing*, 12(24):4055, 2020. iv, vii, 19, 20, 41

[4] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021. 6

[5] Hao Chen. LEVIR: Large-scale and high-resolution remote sensing image dataset. Website, n.d. Accessed on 20th May 2023. 35, 36

[6] Hongruixuan Chen, Edoardo Nemni, Sofia Vallecorsa, Xi Li, Chen Wu, and Lars Bromley. Dual-tasks siamese transformer framework for building damage assessment. *arXiv preprint arXiv:2201.10953*, 2022. iv, vii, 21

[7] Hao Chen, Zipeng Qi, and Zhenwei Shi. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021. iv, vii, 14, 15, 37

[8] Hao Chen and Zhenwei Shi. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote Sensing*, 12(10):1662, 2020. iii, vii, viii, 1, 3, 4, 25, 34, 35, 36, 41, 43, 48

[9] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 3

[10] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10337–10346, 2020. 12

[11] Daniel Corral. *Designing Electrolyzers to Elucidate Governing Phenomena Involved in the Electroreduction of CO 2 Operating in Bulk-Neutral pH*. Stanford University, 2022. 45

[12] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995. 3

[13] Yifan Da, Zhiyuan Ji, and Yongsheng Zhou. Building damage assessment based on siamese hierarchical transformer framework. *Mathematics*, 10(11):1898, 2022. iv, vii, 17, 19

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 10

[15] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 45

[16] DIUx-xView. xView2 Baseline Repository. https://github.com/DIUx-xView/xView2_baseline, Accessed 2023. 22

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. iv, vii, 6, 9, 10, 11

[18] Hidetoshi Furukawa. Deep learning for target classification from sar imagery: Data augmentation and translation invariance. *arXiv preprint arXiv:1708.07920*, 2017. 3

[19] Ritwik Gupta, Richard Hosfelt, Sandra Sajeev, Nirav Patel, Bryce Goodman, Jigar Doshi, Eric Heim, Howie Choset, and Matthew Gaston. xbd: A dataset for assessing building damage from satellite imagery. *arXiv preprint arXiv:1911.09296*, 2019. iii, 48

[20] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009. 3, 4

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 46

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 20

[23] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022. 3

[24] Gong Jianya, Sui Haigang, Ma Guorui, and Zhou Qiming. A review of multi-temporal remote sensing data change detection algorithms. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B7):757–762, 2008. 2

[25] Navjot Kaur, Cheng-Chun Lee, Ali Mostafavi, and Ali Mahdavi-Amiri. Dahitra: Damage assessment using a novel hierarchical transformer architecture. *arXiv preprint arXiv:2208.02205*, 2022. iii, v, vii, 4, 5, 22, 25, 32, 35, 37, 39, 41, 42, 44, 46, 47, 48

[26] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical report*, 2009. 10

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3, 9

[28] LinkAnJarad. global_local_attention_module_pytorch. GitHub repository, n.d. Accessed on 20th May 2023. 25, 35, 39

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 6

[30] nka77. Dahitra. GitHub repository, n.d. Accessed on 20th May 2023. 35, 41

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. iii, iv, 5

[32] Yu Shen, Sijie Zhu, Taojiannan Yang, and Chen Chen. Cross-directional feature fusion network for building damage assessment from satellite imagery. *arXiv preprint arXiv:2010.14014*, 2020. vii, 6

[33] Yu Shen, Sijie Zhu, Taojiannan Yang, Chen Chen, Delu Pan, Jianyu Chen, Liang Xiao, and Qian Du. Bdanet: Multiscale convolutional neural network with cross-directional attention for building damage assessment from satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021. iv, vii, viii, 16, 17, 46

[34] Chull Hwan Song, Hye Joo Han, and Yannis Avrithis. All the attention you need: Global-local, spatial-channel attention for image retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2754–2763, 2022. iv, vii, 4, 13, 14, 25, 35, 38

[35] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019. 9

[36] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 4

[37] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 4, 12

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. iv, vii, 4, 6, 7, 8, 9, 25

[39] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020. 31

[40] Shuai Wang and Zhendong Su. Metamorphic testing for object detection systems. *arXiv preprint arXiv:1912.12162*, 2019. 3

[41] Chuyi Wu, Feng Zhang, Junshi Xia, Yichen Xu, Guoqing Li, Jibo Xie, Zhenhong Du, and Renyi Liu. Building damage detection using u-net with attention mechanism from pre-and post-disaster remote sensing datasets. *Remote Sensing*, 13(5):905, 2021. 37

[42] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 3

[43] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 45

[44] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 45

[45] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 5

[46] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019. 3

[47] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 6