ENHANCING GENERALIZABILITY IN BUILDING DAMAGE DETECTION: DOMAIN ADAPTATION AND AUGMENTATION APPROACHES FOR POST-DISASTER ASSESSMENT

A Thesis by

P. Bharath Chandra Reddy

Bachelor of Technology, St. Martins Engineering College, 2021

Submitted to the School of Computing
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

December 2024

ENHANCING GENERALIZABILITY IN BUILDING DAMAGE DETECTION:
DOMAIN ADAPTATION AND AUGMENTATION APPROACHES FOR
POST-DISASTER ASSESSMENT

The following faculty members have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Shruti Kshirsagar, Committee Chair

Rajiv Bagai, Committee Member

Atri Dutta, Committee Member

DEDICATION

I dedicate this thesis to my family and friends.

ACKNOWLEDGEMENTS

I would like to express my gratitude and deepest appreciation to my advisor, Dr. Shruti Kshirsagar, for her guidance throughout my program. I extend my gratitude to members of my committee, Dr. Atri Dutta, Dr. Rajiv Bagai for their suggestions and comments on this research work.

ABSTRACT

The increasing frequency of natural disasters necessitates rapid and reliable methods for assessing building damage to aid timely disaster response. This thesis investigates the potential for deep learning models to generalize across diverse geographical and environmental contexts for building damage detection, a critical requirement for models deployed in real-world scenarios. The primary objective is to evaluate biases within these models, specifically those that may limit performance in out-of-domain datasets, and to propose methodologies to enhance model robustness.

This study leverages two datasets, xBD and Ida-BD, utilizing both in-domain and out-of-domain data to analyze model biases. We introduce a novel Fusion Augmentation technique designed to enhance the model's ability to capture building edges, thereby improving classification of damage levels, especially in regions with dense vegetation. A series of supervised and unsupervised domain adaptation techniques, including CORAL, were applied to improve model generalizability across varied disaster scenarios without requiring target labels. Grad-CAM visualization techniques further support explainability by offering insights into the areas of focus in model predictions.

Results demonstrate that combining Fusion Augmentation with domain adaptation significantly improves model accuracy, especially in damage classes, and reduces location specific biases. This research contributes a practical framework for developing reliable and generalizable building damage detection models, which can serve as valuable tools in post-disaster assessment, ultimately supporting faster and more effective humanitarian responses.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

Chapter Page

TABLE OF CONTENTS (continued)

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF FIGURES (continued)

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Natural Disasters and Building Damage Detection

The frequency of natural disasters, including wildfires, hurricanes, floods, and earthquakes, has been rising, with climate change contributing to the growing severity of certain types of hazards, such as extreme weather events [17]. Natural disasters are defined by the United Nations Office for Disaster Risk Reduction (UNDRR) as extreme natural occurrences that result in the swift or gradual deterioration of ecosystems or the loss of human lives. These disasters often have devastating impacts, not only disrupting lives but also causing significant economic losses [18]. In the aftermath, it is essential to rapidly and accurately assess the extent of structural damage, particularly to critical infrastructure like buildings. Accurate damage assessments are crucial for humanitarian efforts as they provide key data that help decision-makers prioritize emergency responses and allocate resources efficiently to areas of greatest need.

When any disaster event occured, speed in rescue operations is critical. The length of time it takes to calculate damage in the aftermath of a catastrophic event can be the difference between life and death. Yet rescuers frequently must work through failures in local communication and transportation systems, making proper judgment dangerous, time-consuming and challenging. In a situation like this, Satellite and aerial imagery are a safer, more expansive alternative, but analysts still have to manually review pictures, which is time-consuming [19].

Machine learning (ML) models have started to show potential in damage detection tasks, but the extent of their operational use in real-time disaster assessments is still evolving. Significant research has been conducted to improve the efficiency of building damage assessment by employing deep learning (DL) algorithms and satellite imagery, reducing the need for time-intensive manual analysis [2]. The DL models can process vast amounts of

data and learn from it to detect damaged structures, which helps accelerate disaster response efforts [2]. For these models to be reliable and provide confident, actionable insights, it is critical to ensure they remain unbiased. Bias in this context refers to the model inadvertently dependent on certain patterns, features, or geographic areas present in the training data, which may affect its accuracy when deployed in real time rescue operations.

Any DL model trained to detect building damage should have high performance and accuracy under various challenging conditions that it face in real time disaster events and it should generalize well across different events, meaning it can accurately predict damage levels in new unseen disasters without requiring extensive retraining [20], [21]. Such robustness is essential to build trust among decision-makers and emergency response teams who rely on these models to guide their actions in rescue and recovery operations.

1.2 Background & Motivation

Traditional methods for building damage assessment rely on a variety of techniques that do not involve artificial intelligence or deep learning. These traditional approaches include field inspections, vibration-based methods, and remote sensing analysis. Field inspections involve on-site evaluations by engineers who visually assess structural damage, which is thorough but time-consuming. Vibration-based methods analyze the dynamic characteristics of buildings, such as natural frequencies and mode shapes, to detect changes that may indicate damage [22].

Accurate and quick assessment of building damage is crucial in disaster response, as it enables emergency teams to prioritize their efforts. Severely damaged areas require urgent assistance, while areas with minor or no damage may need fewer resources. However, traditional assessment procedures are often labor-intensive and time-consuming, particularly in large-scale disasters.

Recent advances in imaging sensor technologies in satellite and computer vision algorithms have enabled more efficient and precise analyses of disaster-hit regions [2]. Satellite imaging provides large-scale, high-resolution data that is essential for applications such as

disaster monitoring, while computer vision algorithms, especially those powered by deep learning, can process this data to detect and classify damage. In particular, Convolutional Neural Networks (CNNs), a type of deep learning model, have significantly transformed building damage identification [1]. CNNs are capable of analyzing complex patterns within large datasets, distinguishing between damage levels from minor cosmetic issues to severe structural failures through automated image analysis. However, DL models often require large, well-annotated datasets to achieve high performance, and their generalization capabilities remain a challenge.

1.2.1 xView2 Competition

The xView competition series, particularly the xView2 Challenge, is organized by the Defense Innovation Unit (DIU) to advance the use of computer vision algorithms for disaster response. The primary goal of the xView2 Challenge is to automate the process of assessing building damage using satellite imagery following natural disasters. This initiative seeks to improve the speed and accuracy of damage assessments, which are crucial for effective disaster response and recovery efforts [2].

The xView2 Challenge builds on its predecessor, the xView1 Challenge, by focusing specifically on localizing and categorizing building damage. Participants in the competition are tasked with developing machine learning models that can analyze pre- and post-disaster satellite images to detect and classify damage levels according to a standardized scale. The challenge utilizes the xBD dataset [1], which is one of the largest and most comprehensive collections of annotated satellite imagery for building damage assessment. This dataset includes images from various disaster scenarios such as wildfires, landslides, earthquakes, and floods. More details about the dataset are explained in Section 4.2.

In recent years, xView2 has been adopted by leading international organizations to support disaster and recovery. It is used by the US National Guard, the United Nations, the World Bank and other leading international humanitarian organizations [23]. The California National Guard and the Australian Geospatial-Intelligence Organization, for instance, use

xView2 as part of their workflows to address problems like wildfires. In Nepal, for example, the platform helped to assess flood damage and follow-up landslides, thereby helping to determine where resources should be deployed. All these partnerships showcase xView2 as an effective solution to help make operations more efficient during a disaster [23].

xView2 has also been extensively used in Turkey following the terrible earthquake in Adiyaman. It aided ground teams of the UNs International Search and Rescue Advisory Group in search and rescue operations, especially in locations where conventional assessments had become delayed. The platform enabled responders to find previously unrecognized damaged areas, crucial for rapid relief operations. Moreover, xView2 has been used by Turkeys Disaster and Emergency Management Presidency, the World Bank, the International Federation of the Red Cross, and the United Nations World Food Program to manage disasters effectively [23]. Such deployments highlight how crucial it is to offer measurable intelligence to help mitigate the impact of natural disasters on economic and livelihood of humans across the world.

CHAPTER 2

BACKGROUND

Machine learning and deep learning have revolutionized many industries from medicine to disaster management, giving machines the power to learn complex patterns and take data driven decisions. This chapter presents the basic ideas behind these technologies, examples of some of the most innovative algorithms, and an overview of the key approaches that are underlying this thesis.

2.1 Machine Learning VS Deep Learning

Artificial Intelligence (AI) is the broad science involved in building machines that can do the work that humans would normally do. These tasks involve reasoning, problem-solving, natural language processing, and perception. AI is a collection of techniques and methods designed to allow computers to perform cognitive tasks. Machine Learning (ML) is a subset of artificial intelligence (AI) that focuses on developing algorithms that allow computers to learn from and make decisions based on data without being explicitly programmed. In ML, algorithms identify patterns within large datasets, use these patterns to make predictions or decisions, and improve their performance over time with more data. These algorithms are broadly categorized into supervised learning, unsupervised learning, and reinforcement learning, depending on the nature of the learning task [24].

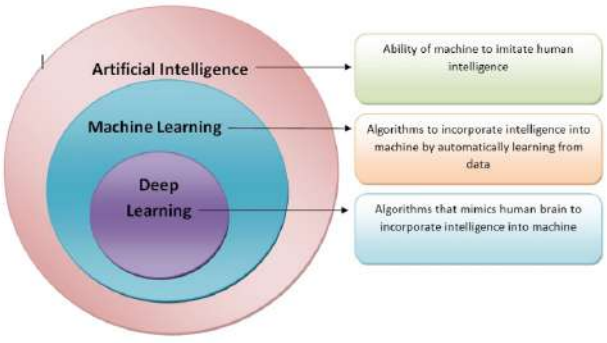


Figure 2.1: ML and DL are subsets of AI [6]

Deep learning (DL)is a specialized subfield of machine learning, which is built on the foundation of artificial neural networks (ANNs) and the concepts of ML. Neural Networks are computational models inspired by the human brain. These networks are composed of layers of interconnected nodes (neurons), through which data is processed to enable learning [24]. Deep learning algorithms are categorized based on their architecture, training method, and specific tasks they are designed to perform. Figure 2.2 shows biological brain and a simple neural network with one input layer, one hidden layer and one output layer. DL models are particularly powerful in handling large-scale datasets and are designed to automatically extract features from raw data, making them suitable for tasks like image recognition, natural language processing, and speech recognition [24].

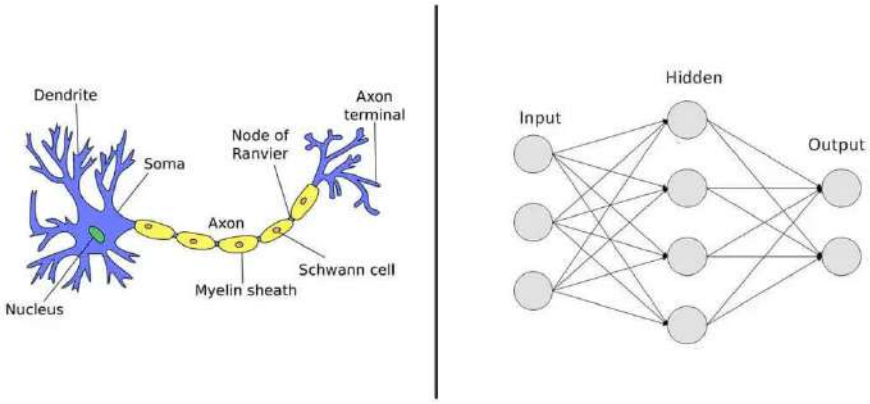


Figure 2.2: Biological Brain and Neural Network [7]

2.1.1 Difference Between Machine Learning and Deep Learning

While both machine learning and deep learning are subsets of AI, deep learning represents a more advanced evolution of machine learning. Table 2.1 gives a breakdown of the key differences with examples:

Table 2.1: Difference Between Machine Learning and Deep Learning

Aspect	**Machine Learning**	**Deep Learning**
Data Dependencies	Works well with small to medium-sized datasets. Requires feature extraction to be performed manually by experts.	Requires large datasets to perform optimally. Automatically performs feature extraction, making it highly suitable for unstructured data like images, audio, and text.
Hardware Requirements	Can run on conventional hardware like CPUs.	Requires significant computational power, usually involving Graphics Processing Units (GPUs) or specialized hardware like TPUs due to the complexity of neural networks.
Training Time	Training time is relatively fast, depending on the algorithm and data size.	Takes longer to train due to the deep architecture and large volume of parameters being optimized.
Interpretability	Easier to interpret, as models like decision trees or linear regression offer insights into how decisions are made.	Harder to interpret due to the "black box" nature of deep neural networks, where its challenging to understand how decisions are made internally.
Performance	Can perform well on structured data (e.g., tabular data) and tasks like regression and classification with moderate accuracy.	Delivers superior performance, especially in complex tasks like image recognition, speech processing, and language translation, where unstructured data is prevalent.
Example Use Case	Example: Predicting housing prices using features such as location, size, and condition.	Example: Image classification with convolutional neural networks (CNNs), such as recognizing handwritten digits (MNIST dataset) or distinguishing between cats and dogs.

2.1.2 Applications of Machine Learning

Machine learning has gained wide adoption across various industries due to its versatility and ability to improve decision-making through data-driven insights. Below are some of the primary applications of machine learning.

- **Healthcare:** In healthcare, ML models are used for diagnosing diseases, predicting patient outcomes, and optimizing treatment plans. A notable example is the use of ML algorithms to detect cancer in medical images, such as radiology scans, or to predict disease progression based on patient data [25].
- **Finance:** In the financial sector, ML is employed for fraud detection by analyzing transaction patterns and identifying anomalies. Additionally, ML is used in algorithmic trading, where it helps optimize trading strategies based on real-time data [26].
- **Marketing and Advertising:** E-commerce platforms and streaming services leverage ML for personalized recommendations. These recommendation systems analyze users' behavior and preferences to suggest relevant products, content, or services [27].
- **Manufacturing:** Predictive maintenance is a key application of ML in manufacturing, allowing companies to monitor the condition of machinery and predict equipment failures before they occur, reducing downtime and maintenance costs [28].

2.1.3 Applications of Deep Learning

Deep learning's ability to process unstructured data has led to breakthroughs in various fields. Its applications often involve tasks that require understanding images, audio, text, or video data. Some of the most impactful applications are:

- **Image and Video Processing:** Convolutional Neural Networks (CNNs) are widely used in image recognition, object detection, and video processing tasks. For instance, deep learning has been applied in medical imaging to detect tumors in radiographs and MRIs with high accuracy [29].

- **Natural Language Processing (NLP):** Deep learning models, such as transformers, have revolutionized NLP tasks including machine translation, sentiment analysis, and text generation. For example, Google Translate utilizes DL models for translating text between different languages [30].
- **Speech Recognition:** Automatic speech recognition (ASR) systems, such as those used by virtual assistants like Apple's Siri and Amazon's Alexa, rely on deep learning models to convert spoken language into text. Recurrent Neural Networks (RNNs) and attention mechanisms are key technologies in this domain [31]
- **Generative Models:** Deep learning has given rise to generative models like Generative Adversarial Networks (GANs), which are used for tasks such as generating realistic images, videos, and even deep fakes. GANs have found applications in creative industries, including video game design and virtual reality [32].

2.2 Overview of Algorithms in Deep Learning

1. **Feedforward Neural Networks (FNNs):** Feedforward Neural Networks, also called Multi-Layer Perceptron's (MLPs), are the most basic type of neural network. They consist of an input layer, one or more hidden layers, and an output layer. Each neuron in one layer is connected to every neuron in the next layer through weighted edges, and data flows only in one direction from the input to the output.

   Working Mechanism:

   In FNNs, neurons apply activation functions to the weighted sum of their inputs and propagate the output to forward. Common activation functions include the sigmoid, hyperbolic tangent (tanh), and Rectified Linear Unit (ReLU). The learning process involves adjusting the weights using a back propagation algorithm, which minimizes the error (loss) between predicted and actual outputs by updating the weights via gradient descent.

   Applications:

While relatively simple, FNNs are effective for tasks involving structured data, such as regression and classification problems. For example, they are commonly used in predictive modeling, where the relationship between input features and the target variable is learned.

2. **Convolutional Neural Networks (CNNs):** Convolutional Neural Networks (CNNs) are specialized deep learning architectures designed for processing grid-like data, such as images. They consist of convolutional layers, pooling layers, and fully connected layers. A key feature of CNNs is the use of convolutional filters (kernels) that scan through the input data and capture spatial features, enabling the model to recognize patterns such as edges, textures, and shapes.

   Working Mechanism:

   - **Convolutional Layers:** These layers perform convolution operations, where small filters slide over the input data and generate feature maps. This reduces the spatial dimensions while preserving important features.
   - **Pooling Layers:** Pooling (often max pooling) reduces the dimensionality of the feature maps, which helps in making the network more computationally efficient and less sensitive to the position of features in the input.
   - **Fully Connected Layers:** After several convolutional and pooling layers, the resulting feature maps are flattened and passed through fully connected layers to make predictions.

   Applications:

   CNNs have revolutionized computer vision and are widely used in tasks such as:

   - **Image Classification:** Assigning a label to an image (e.g., recognizing objects in images)
   - **Object Detection:** Identifying and locating multiple objects within an image (e.g., self-driving cars detecting pedestrians and other vehicles).

- **Image Segmentation:** Dividing an image into different regions or segments based on the object class (e.g., medical imaging to identify regions of interest such as tumors).

3. **Recurrent Neural Networks (RNNs):** Recurrent Neural Networks (RNNs) are designed for sequential data, where the order of inputs is crucial. Unlike feedforward networks, RNNs have connections that form cycles, allowing them to maintain memory of previous inputs. This memory enables RNNs to process time-series data, natural language, and any other data that involves sequences.

   Working Mechanism:

   At each time step, an RNN takes the current input and the hidden state from the previous time step to generate an output and update the hidden state. This mechanism allows RNNs to maintain a dynamic memory of past inputs and adapt their predictions based on sequential information. The challenge with traditional RNNs, however, is that they suffer from the vanishing gradient problem, making it difficult for them to learn long-term dependencies.

   Applications:

   - **Time-Series Forecasting:** RNNs are commonly used for tasks such as stock price prediction or weather forecasting, where historical data plays a critical role.
   - **Natural Language Processing (NLP):** RNNs are used in tasks like language modeling, sentiment analysis, and machine translation.

   Variants of RNNs:

   - **Long Short-Term Memory (LSTM):** A special kind of RNN designed to address the vanishing gradient problem. LSTMs have memory cells that can selectively retain information over long periods, making them ideal for tasks involving long-term dependencies [33].
   - **Gated Recurrent Units (GRUs):** A simplified variant of LSTMs that use fewer

gates, making them computationally more efficient while still effectively handling long-term dependencies [34].

4. **Generative Adversarial Networks (GANs):** Generative Adversarial Networks (GANs) consist of two neural networks: a generator and a discriminator. These networks are trained simultaneously through adversarial learning. The generator tries to produce realistic data (e.g., images, audio), while the discriminator evaluates whether the data is real (from the training set) or fake (produced by the generator). The goal is for the generator to learn to create data that the discriminator can no longer distinguish from the real data.

   Working Mechanism:

   - **Generator:** Takes random noise (often sampled from a uniform or Gaussian distribution) and generates data samples.
   - **Discriminator:** Receives real and generated data and outputs a probability indicating whether the input is real or fake.
   - The two networks are trained in a min-max game: the generator tries to fool the discriminator, and the discriminator tries to distinguish between real and generated data. The training continues until the generator produces data indistinguishable from real samples.

   Applications:

   - **Image Generation:** GANs are capable of generating realistic images from scratch, such as creating high-resolution photographs of human faces.
   - **Image-to-Image Translation:** Applications like converting sketches to photos or enhancing images (e.g., turning black-and-white images into color).
   - **Deepfakes:** GANs have been used to create realistic but fake videos or images, such as altering the faces of individuals in videos [32].

5. **Autoencoders:** Autoencoders are unsupervised learning models that aim to learn a compressed representation (encoding) of data and then reconstruct it back to its original form. They consist of two parts: an encoder that compresses the input data into a lower-dimensional representation and a decoder that reconstructs the data from this compressed representation.

   Working Mechanism:

   - **Encoder:** Transforms the input into a compressed latent space.
   - **Decoder:** Reconstructs the original input from this latent representation. Autoencoders are typically trained by minimizing the reconstruction error, which measures how different the reconstructed data is from the original input.

   Applications:

   - **Dimensionality Reduction:** Similar to Principal Component Analysis (PCA), autoencoders can reduce the dimensionality of data while retaining important features.
   - **Anomaly Detection:** Autoencoders can identify anomalies by training on normal data and detecting deviations during the reconstruction process.
   - **Denoising:** Denoising autoencoders are used to remove noise from data by training the model to reconstruct clean data from noisy inputs.

   Variants:

   - **Variational Autoencoders (VAEs):** These are generative models that, unlike traditional autoencoders, impose a probabilistic structure on the latent space, making them capable of generating new data samples [35].

6. **Transformers:** Transformers are a type of deep learning architecture that is especially effective for processing sequential data, such as language, without relying on recurrence like RNNs. Instead, transformers use self-attention mechanisms to model dependencies between inputs, making them highly efficient for tasks that require understanding long

range dependencies.

Working Mechanism:

- **Self-Attention Mechanism:** The core idea of self-attention is to compute a weighted sum of all input elements, where the weights (attention scores) are learned based on the relevance of each element in the sequence to the current element.
- **Positional Encoding:** Since transformers do not process inputs in sequence like RNNs, they use positional encodings to provide information about the order of inputs.

Applications:

- **Natural Language Processing (NLP):** Transformers have led to breakthroughs in language models, including the development of BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pretrained Transformers), which have achieved state-of-the-art results in tasks such as translation, text summarization, and question answering [30].
- **Vision Transformers (ViT):** Transformers have also been extended to image processing, where they achieve competitive performance on image classification tasks.

Deep learning algorithms, from simple feedforward networks to advanced architectures like transformers, play a crucial role in solving complex problems across domains. Each algorithm has unique strengths tailored to specific data types and tasks, from sequential data processing (RNNs, LSTMs) to image recognition (CNNs) and generative modeling (GANs, VAEs). By leveraging these algorithms, deep learning continues to drive advancements in fields such as computer vision, natural language processing, and autonomous systems.

2.3 Convolutional Neural Networks for Computer Vision

Convolutional Neural Networks (CNNs) are a specialized type of deep learning architecture designed to process data with a grid-like topology, such as images, where the spatial structure of the data is crucial. CNNs have revolutionized the field of computer vision, and they are now widely applied in various domains, including image classification, object detection, and video analysis. This section provides a comprehensive overview of CNNs, including their structure, key components, working mechanism, and applications.

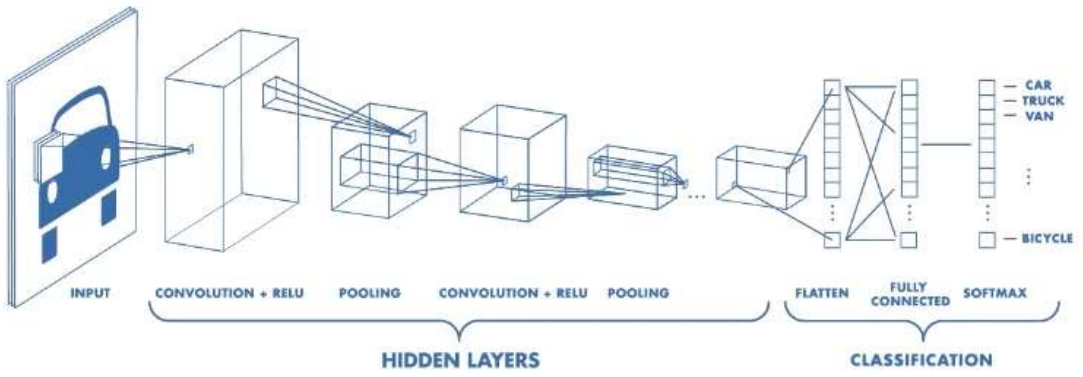


Figure 2.3: Convolutional Neural Network Architacture [8]

2.3.1 Overview of Convolutional Neural Networks (CNNs)

1. **Structure of Convolutional Neural Networks**

   A CNN is composed of several layers, each of which plays a specific role in processing the input data. These layers work together to progressively extract higher-level features from the raw input. The fundamental building blocks of a CNN are:

   - **Input Layer:** The input to a CNN is typically a multi-dimensional array representing the raw data. For example, in image processing, the input is a 2D array (or a 3D array if the image has multiple channels, such as RGB images). Each element in the array represents a pixel value.

- **Convolutional Layer:** The convolutional layer is the core component of a CNN, responsible for extracting features from the input data. It uses a set of learnable filters (also called kernels), which slide across the input and perform element-wise multiplications. The filter extracts local features, such as edges, textures, or colors, depending on its learned weights. The convolution operation preserves the spatial relationship between pixels, making CNNs effective for image processing tasks [8].
- **Mathematics of Convolution:** For a 2D input image I and a 2D filter (kernel) K, the convolution operation is mathematically represented as:

$$(I * K)(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} I(x + i, y + j)K(i, j) \tag{1}$$

  where I(x,y) represents the pixel value at position (x,y) of the input image, and K(i,j) represents the weight of the kernel at position (i,j). The result is a 2D feature map that captures the response of the filter to the input.
- **Activation Function (Non-Linearity):** After each convolution operation, an activation function is applied to introduce non-linearity, enabling the model to learn complex patterns [8]. The most commonly used activation function in CNNs is the Rectified Linear Unit (ReLU), defined as:

$$f(x) = max(0, x) \tag{2}$$

  ReLU replaces all negative values in the feature map with zero, introducing non-linearity and improving the models capacity to capture intricate patterns in the data. Other examples of activation function is shown in Figure 2.4.
- **Pooling (Subsampling) Layer:** Pooling layers are used to reduce the spatial dimensions of the feature maps, thereby decreasing the number of parameters and computational complexity. Pooling helps in making the CNN invariant to small translations or distortions in the input data [8]. The two most common types of pooling are:

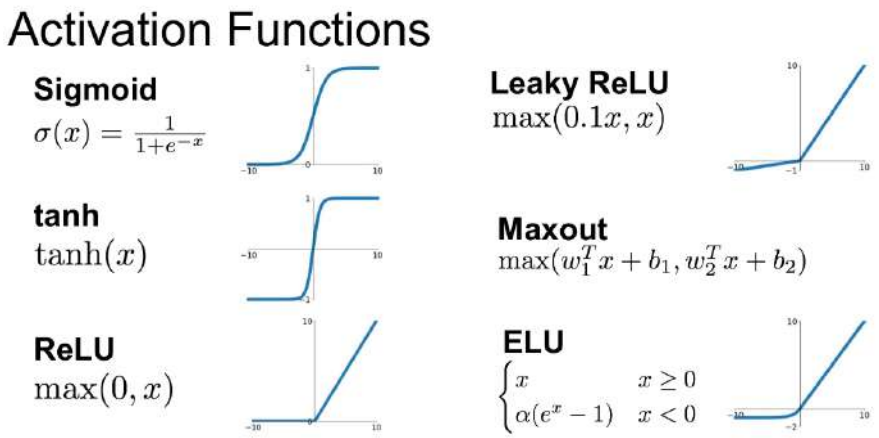


Figure 2.4: Examples of Activation Functions [9]

- **Max Pooling:** Takes the maximum value from each patch of the feature map.
- **Average Pooling:** Takes the average value from each patch of the feature map.

Max pooling is more commonly used in modern CNN architecture, as it tends to preserve the most important features. Pooling reduces the size of the feature map but retains the essential information, making the model more efficient.

- **Fully Connected (FC) Layer:** After several convolutional and pooling layers, the high-level, abstracted features are flattened into a 1D vector, which is then passed to one or more fully connected (dense) layers. Each neuron in a fully connected layer is connected to every neuron in the previous layer, allowing the model to make predictions based on the learned features [8].

  In the final fully connected layer, an activation function like softmax or sigmoid is applied to generate the output probabilities for classification tasks [8].

2. **Key Concepts in Convolutional Neural Networks:**

- **Receptive Field:** The receptive field refers to the region of the input image that a particular neuron in the feature map is sensitive to. As the depth of the network increases, the size of the receptive field grows, allowing the model to capture more global features of the input. Initially, neurons may detect simple features like edges,

but deeper layers can detect complex structures like objects.

- **Stride:** Stride defines the step size of the convolutional filter as it moves across the input. A stride of 1 means the filter moves one pixel at a time, while a larger stride value will move the filter in larger steps. Increasing the stride reduces the size of the output feature map, but also reduces the overlap between receptive fields, which may result in a loss of spatial information.
- **Padding:** Padding refers to adding extra pixels (typically zeros) around the input image. This is done to control the spatial size of the output feature map. Padding helps preserve the spatial dimensions of the input, especially in deeper networks where successive convolutions can drastically reduce the feature map size. There are two main types of padding:
  - **Valid Padding:** No padding is applied, and the feature map size decreases after convolution.
  - **Same Padding:** Padding is applied to ensure that the output feature map has the same size as the input.
- **Parameter Sharing:** In CNNs, the same filter is applied across different parts of the input image, meaning the filter's weights are shared across the entire input. This reduces the number of learnable parameters compared to fully connected networks, making CNNs more efficient and less prone to overfitting.

3. **Working Mechanism of CNNs**

The CNN architecture is based on a hierarchical approach to learning features from data. The working mechanism of a CNN can be broken down into the following steps:

- **Input:** The network takes a raw image as input, typically represented as a grid of pixel values. For example, a grayscale image can be represented as a 2D array of pixel intensities, while an RGB image has three channels (Red, Green, Blue), forming a 3D array.
- **Convolution:** The first convolutional layer applies multiple filters to the input

image. Each filter scans the image, producing a feature map that highlights specific patterns, such as edges or corners.

- **Activation:** After convolution, the ReLU activation function is applied to introduce non-linearity. This allows the network to model more complex relationships between the input and output.
- **Pooling:** Pooling layers downsample the feature maps, reducing their dimensions and the number of parameters in the network. Max pooling is commonly used to retain the most important features while discarding less significant information.
- **Deeper Convolution:** Additional convolutional layers are added to learn increasingly abstract and complex features. The early layers might detect basic shapes, while deeper layers detect higher-level structures such as objects or faces.
- **Flattening and Fully Connected Layers:** Once the features have been extracted by the convolutional and pooling layers, they are flattened into a single vector. This vector is passed through fully connected layers, which combine the learned features and make predictions.
- **Output:** The final layer typically uses a softmax function (for multi-class classification) or a sigmoid function (for binary classification) to output probabilities or class labels.

2.4 Architectures Used in Top-3 Winning Solutions of xView2 Competition

Neural network architectures have evolved significantly over the years, with different designs emerging to address specific challenges and improve performance in various tasks [36]. The development of architectures such as ResNet, ensemble methods, and Siamese neural networks has been driven by the need to overcome limitations of previous models and to tackle increasingly complex problems in machine learning and computer vision. The authors of winning solutions have used multiple complex methods and neural networks to

get the best results possible for the given task. This section gives an introduction to the concepts and methods used in the top-3 winning solutions of the xView2 competition.

2.4.1 Ensemble Networks

The first place winning solution [3] and the third place winning solution [5] have used the ensemble methods to give me the best performance solution. The ensemble methods use multiple learning algorithms to obtain better predictive performance than could be achieved with any of the constituent models alone [10]. The key idea is to train several "base models" or "weak learners" on the same task and then combine their output to produce a single, stronger prediction.

Ensemble Learning Techniques:

1. **Bagging (Bootstrap Aggregating):** Bagging creates diversity by generating random samples from training data and fitting the same model to each sample [10]. This approach is known as a homogeneous parallel ensemble. Random Forests are a popular application of bagging

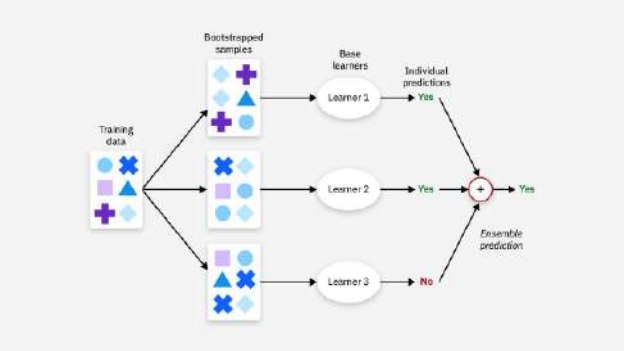


Figure 2.5: Bagging Ensemble Technique [10]

2. **Boosting:** Boosting follows an iterative process, sequentially training each base model on the up-weighted errors of the previous model [10]. This produces an additive model that reduces errors in the final ensemble. Gradient Boosting Machines are a well-known boosting technique.

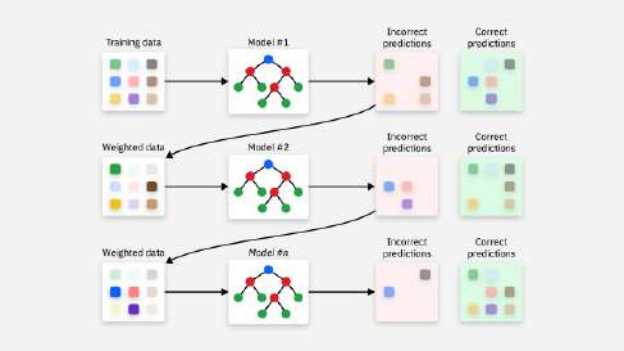


Figure 2.6: Boosting Ensemble Technique [10]

3. **Stacking (Blending):** Stacking combines different types of base models, each trained independently, to create a heterogeneous parallel ensemble [10]. The outputs of these diverse models are then used as inputs to a "meta-learner" that makes the final prediction.

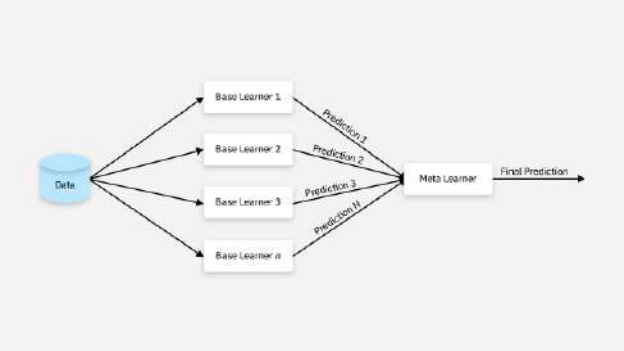


Figure 2.7: Stacking Ensemble Technique [10]

Benefits of Ensemble Networks

1. **Improved Accuracy:** Ensembles often yield better results than individual models, especially when there is significant diversity among the base models [10]
2. **Reduced Overfitting:** By combining multiple models, ensembles can reduce the risk of overfitting to the training data.
3. **Increased Stability:** Ensembles are generally more robust to noise and outliers in the data.

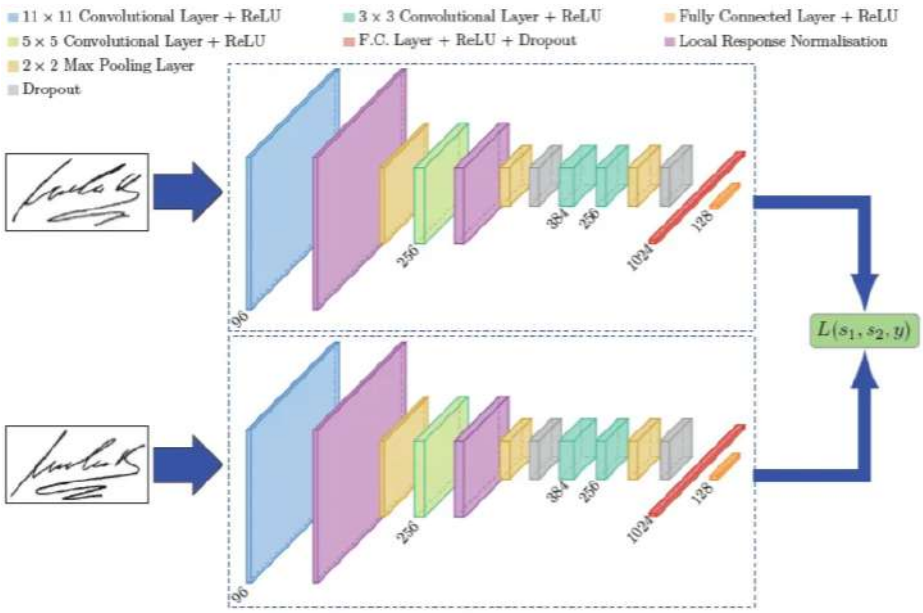


Figure 2.8: Siamese Network used in Signet [11]

2.4.2 Siamese Neural Network

Siamese Neural Networks (SNNs) are a specialized type of neural network architecture designed to compare and measure the similarity between inputs. The authors of the second place [4] winning solution of xView2 competition have used Siamese Neural Networks. A Siamese Neural Network consists of two or more identical subnetworks with the same architecture, configuration, and weights [37]. These subnetworks process different inputs in parallel but share parameters, allowing the network to learn similarities or differences between inputs rather than classifying them directly [38].

SNNs are popular for tasks that involve similarity or matching between data points. Here are some real-world applications.

1. **Face Verification and Recognition:** An SNN is trained to determine whether two face images represent the same person by comparing feature embeddings. It can generalize well to unseen faces.

2. **Signature Verification:** The SNN compares feature representation of a new signature against a database of known signatures for that person.

3. **One-Shot Learning for Object Identification:** An SNN can identify new classes based on one or more labeled examples, enabling quick adaptation to novel categories.

4. **Medical Imaging and Diagnosis:** SNNs help in comparing new medical images to existing ones to identify anomalies or categorize conditions, especially when data is scarce for certain diagnoses.

5. **Building Damage Detection:** SNNs help in comparing bitemporal images (pre and post disaster images) and find the difference between two input images that might caused by any event.

2.5 Performance Measures

In deep learning, particularly in applications like image classification, object detection, segmentation, and natural language processing, a variety of performance measures are used to evaluate model accuracy, robustness, and generalization. Performance metrics help quantify how well the model is performing and indicate areas for improvement. Here is an overview of some commonly used performance measures in deep learning.

1. **Accuracy:** Accuracy is the most straightforward metric, calculated as the percentage of correctly predicted instances out of all predictions made [39]. While accuracy is a good baseline metric, it can be misleading in cases of class imbalance, where one class is dominant (e.g., a model that predicts all samples as "not damaged" in a building damage detection task might still yield high accuracy if most buildings are undamaged).

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negative}}{\text{Total Instances}} \tag{3}$$

2. **Precision:** Precision measures the proportion of true positive predictions among all positive predictions (both true and false). It reflects the models accuracy in identifying

only relevant cases. Higher precision means fewer false positives [39].

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4}$$

3. **Recall(Sensitivity or True Positive Rate):** Recall measures the proportion of true positives identified by the model out of all actual positive cases. Higher recall indicates that the model successfully captures a high number of relevant instances, reducing false negatives [39].

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5}$$

4. **F1 Score:** The F1 score is the harmonic mean of precision and recall, balancing the two metrics. Its particularly useful in cases where there is an uneven class distribution or where both false positives and false negatives are costly [39].

$$\text{F1 Score} = 2.\frac{\text{Precision.Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

5. **Intersection over Union (IoU):** IoU is commonly used in object detection and image segmentation tasks. It measures the overlap between the predicted bounding box (or mask) and the ground truth. IoU values range from 0 to 1, with 1 indicating a perfect overlap. IoU is crucial for evaluating model performance in tasks requiring precise localization, such as damage detection or medical imaging [39].

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{7}$$

2.6 Domain Adaptation

Domain adaptation is a subset of transfer learning that focuses on transferring knowledge from a source domain with labeled data to a target domain with limited or no labeled data, where both domains differ in data distribution [12]. These differences, known as domain shifts, can arise from variations in environmental conditions, sensor types, geographical lo-

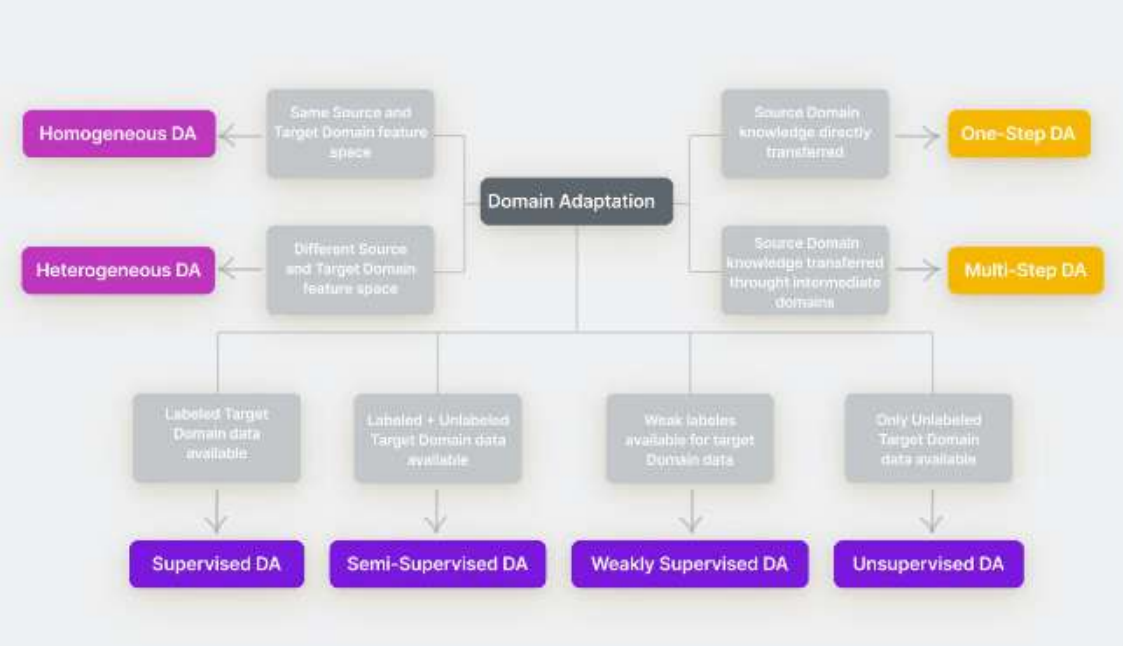


Figure 2.9: Categorization of Domain Adaptation [12]

cations, or temporal differences between the source and target domains. The main objective of domain adaptation is to ensure that models trained on the source data perform effectively on the target data, despite these discrepancies [40].

In practical applications like remote sensing and building damage detection, labeled datasets are often scarce and costly to obtain, especially in post-disaster scenarios where conditions differ widely from the pre-disaster data. Domain adaptation addresses this challenge by using labeled data from different but related contexts, allowing for improved model performance in the target environment [14]. There are several types of domain adaptation based on the availability of labels in the target domain and the similarity between domains as shown in Figure 2.9

Depending on the availability of labeled data in the target domain, domain adaptation techniques can be broadly classified into supervised, unsupervised, and semi-supervised approaches, each offering unique advantages and challenges.

1. **Supervised Domain Adaptation:** In supervised domain adaptation (SDA), a small amount of labeled data is available in the target domain alongside the labeled source domain data. This enables models to learn from both datasets and adjust to the specific

characteristics of the target domain while retaining general features from the source domain [41]. Typical techniques in SDA include fine-tuning, where models pre-trained on source data are re-trained on labeled target data, and re-weighting approaches that adjust the importance of target samples based on domain relevance. Studies show that supervised fine-tuning on even a small portion of labeled target data can significantly improve performance in specialized applications, such as medical imaging and remote sensing, where domain-specific differences are prominent [14]. However, SDA is limited by the need for labeled target data, which is costly and labor-intensive to obtain.

2. **Unsupervised Domain Adaptation:** Unsupervised domain adaptation (UDA) is widely applied in scenarios where no labeled data is available in the target domain, making it highly applicable to real-world situations with limited annotation resources. UDA techniques align source and target data distributions using methods such as adversarial training, feature alignment, and self-supervised learning. For example, Domain-Adversarial Neural Networks (DANN) use a domain discriminator to minimize the discrepancy between source and target features. In remote sensing and natural hazard segmentation, CycleGANs and self-attention mechanisms are used to align visual style and features across domains, effectively bridging differences like lighting, weather, and geographical variation [42]. UDA excels in generalization but may struggle when the domain shift is extreme, as without target labels, fine-grained alignment remains challenging.

3. **Semi-Supervised Domain Adaptation:** Semi-supervised domain adaptation (SSDA) combines aspects of both SDA and UDA by leveraging a small amount of labeled data in the target domain alongside a larger set of unlabeled target data [14]. In SSDA, pseudo-labeling has proven effective in tasks like image classification, where a few labeled target samples can help the model generate accurate labels for similar samples [41]. However, SSDA requires careful handling of pseudo-labels to prevent error propagation, a common issue when pseudo-labels are noisy or incorrect.

CHAPTER 3

LITERATURE REVIEW

This chapter reviews recent advancements in deep learning models for building damage detection, focusing on Building Damage Classification, data augmentation, domain adaptation, generalization and Explainable AI techniques. It also examines how these methods address challenges like data scarcity, class imbalance, and model bias.

3.1 Building Damage Classification

Due to recent advancements in deep learning, a lot of solutions have been proposed to classify building damage using satellite imagery due to its synoptic coverage and predictable availability. Satellite imagery is currently used as input for the identification of building damages by the International Charter, as well as the Copernicus Emergency Management Service for the production of damage grading and reference maps [43]. After introducing the use of Convolutional Neural Networks (CNN) for object instance segmentation [44], the use of CNN's have shown promising results for building damage detection [45].

Currently, two methods are used for building damage detection: 1) By using bitemporal methods that detect changes between pre and post disaster data and 2) by using methods that take a single post disaster image as input [46].

Advantages of using bitemporal and post event images:

Bitemporal analysis allows for quantitative measurement of changes in building appearance, shape, and surroundings [45]. These measurements can be used to develop more robust damage assessment algorithms. In dense urban areas with complex building structures, bitemporal analysis helps in accurately delineating individual buildings and assessing their damage states, which can be challenging with single-image analysis.

While using bitemporal (pre- and post-disaster) images generally offers more comprehensive information for building damage detection, there are some advantages to using only

single post-event disaster images. Post-disaster images are often available more quickly than matched pre-disaster images. This helps in rapid initial damage assessments without waiting for pre-disaster imagery to be sourced or aligned [47]. For sudden, unexpected disasters (like the Beirut explosion), pre-disaster imagery might not be readily available. Single post-event analysis can still provide valuable information in these cases [48].

3.1.1 Types of Building Damage Detection

In building damage detection, the classification of damage severity is crucial for effective disaster response and resource allocation. Models typically adopt either 2-level classification or 4-level classification schemes, each serving different purposes based on the level of detail required [49].

2-Level Classification (Binary Classification):

A 2-level classification simplifies the problem by categorizing buildings into only two classes, such as "damaged" or "undamaged." This approach is beneficial in scenarios where quick, broad assessments are needed, such as during early disaster response. By providing a binary decision, it allows emergency teams to rapidly identify whether a building requires attention, streamlining the initial stages of disaster management.

Advantages:

- Faster and simpler to implement, making it ideal for real-time applications.
- Requires less computational power and training data.
- Useful for immediate assessments when high-level decisions need to be made quickly, such as evacuations or resource deployment.

Limitations:

- Lacks granularity, making it less useful for detailed recovery planning.
- Cannot differentiate between varying degrees of damage, potentially leading to inefficient allocation of resources.

4-Level Classification (Multiclass Classification)

A 4-level classification scheme provides a more detailed assessment by categorizing buildings into four distinct damage levels, such as "no damage," "minor damage," "major damage," and "destroyed." This granular approach offers a more comprehensive understanding of the extent of the damage, enabling better-informed decisions for both immediate disaster response and long-term recovery efforts [49].

Advantages:

- Provides a detailed breakdown of damage severity, allowing for more targeted interventions.
- Helps prioritize resources by identifying buildings that are severely damaged or destroyed versus those that need minor repairs.
- Useful for insurance assessments, structural evaluations, and recovery planning.

Limitations:

- More complex, requiring higher computational power and a larger, more balanced dataset for training.
- Can lead to misclassifications in cases where damage categories overlap or the visual differences between them are subtle.

4-level classification is more suitable for detailed damage assessments where more granular information is needed to guide recovery efforts, allocate resources, or perform insurance evaluations [50]. For instance, knowing whether a building is "minorly damaged" versus "destroyed" enables agencies to prioritize areas that require urgent reconstruction and immediate rescue operations versus those that can be repaired with fewer resources [1].

3.2 In-Domain and Out-of-Domain Datasets

The evaluation of machine learning models on out-of-domain (OOD) datasets has become a critical research area, especially in high-impact applications such as disaster re-

sponse and building damage detection [51]. While traditional model evaluation is often based on in-domain datasets where the training and test data share similar distributions this approach may not adequately capture the models ability to generalize to real-world scenarios. Out-of-domain evaluation tests the models robustness by exposing it to data that diverges significantly from the training distribution in terms of geographic, temporal, or contextual factors.

Evaluating models on both in-domain and out-of-domain data provides insights into their generalization capabilities. As one study noted, this allows researchers to determine if a model is "more suitable for domain-specific use cases" or achieves "better performance on a cross-domain task" [52].

In-domain datasets are often biased due to the homogeneity of training data, leading to models that are overfitted to certain locations, environmental conditions, or types of disasters [51]. Out-of-domain (OOD) datasets are essential for evaluating a models ability to generalize to real-world scenarios, where data distributions often differ from the training set [51].

In building damage detection (BDD), models trained on in-domain (ID) datasets may perform well on familiar disaster images but struggle with unseen disaster types [53], [21]. Evaluating models on OOD datasets provides a more accurate measure of their generalization ability. For example in the paper by Benson and Ecker [51] two techniques: Adaptive Batch Normalization (AdaBN) and Stochastic Weight Averaging (SWA) were explored to improve OOD performance. In the paper [21], authors implemented domain adaptation methods to improve the performance on OOD dataset. But since the unexplored regions and locations will be huge, there's no definite solution to improve the model performance on unseen location.

3.2.1 Available Datasets for Building Damage Detection

Annotating building damage is a labor-intensive task, and the training datasets currently available for building damage identification are highly limited. This has led to ef-

forts in organizing publicly accessible, building-related datasets. In this paper, eight publicly available building damage datasets have been systematically compiled and categorized into two main types based on the imaging method: satellite, aerial. Among these, some datasets, which cover a range of hazards such as earthquakes, floods, wildfires, hurricanes, and tsunamis, are classified under satellite imagery.

Table 3.1 provides an overview of the data, serving as a visual reference to the publicly accessible building damage datasets. These datasets offer authentic and diverse training resources for deep learning models, providing a solid data foundation for research in building damage detection.

Table 3.1: List Of Publicly Available Datasets For Building Damage Detection

Ref	**Name**	**Imagery Type**	**Disaster Type**
[1]	xBD dataset	Satellite	Combined dataset
[14]	Ida-BD Dataset	Satellite	Hurricane
[54]	Haiti earthquake and Nepal Earthquake Dataset	Satellite	Earthquake
[55]	Hurricane Harvey Dataset	Satellite	Hurricane
[56]	Satellite Images of Hurricane Damage Dataset	Satellite	Hurricane
[57]	Yushu earthquake	Airborne	Earthquake
[58]	ABCD dataset	Airbone	Tsunami
[59]	ISBDA dataset	Airborne	Hurricane
[60]	FloodNet dataset	Airborne	Hurricane

3.3 Bias in Building Damage Detection

In the context of building damage detection, bias refers to systematic errors or tendencies that lead a model to perform inconsistently across different datasets or situations, particularly when the models training data does not fully represent the variety of conditions it will encounter in real-world scenarios [61]. The xBD dataset which is the major open-source data available for public use is the most commonly used dataset for building damage detection.

The bias can occur due to several factors like Data Imbalance, Geographic Bias, Sensor or Image Bias, Class Label Bias [62]. It was found that, in most of the existing open datasets, non-damaged buildings account for the majority, and severely damaged buildings account for a minority [1], [21], which is not a favorable factor in post disaster rescue. The

model trained on xBD dataset are more biased towards the majority class [21], [14], [45].

Furthermore, In the paper by Melamed et.al., [61] The authors identify a significant bias in the xBD dataset, which is widely used for building damage detection from satellite imagery. This bias stems from the spatial distribution of damaged buildings, where damaged structures tend to be clustered together. The study reveals that the damage level of a building is strongly correlated with the damage levels of nearby buildings. This study [61] analyzes the top-5 solutions from the xView2 competition, which uses the xBD dataset. The authors find that these models perform well on densely damaged areas but struggle with isolated damaged buildings. This performance discrepancy could lead to critical oversights in disaster scenarios and potentially delay humanitarian aid [61]. They suggest creating auxiliary datasets and challenges that focus on more localized and limited damage scenarios to improve model robustness.

3.4 Data Augmentation

Data augmentation is a critical tool for enhancing deep learning models, especially in domains like building damage detection where datasets are often limited. Across the reviewed studies, different augmentation techniques such as flipping, rotation, patching, and masking have been shown to improve model performance by expanding the training set, reducing overfitting, and improving generalization.

In the paper by Adedeji et.al., [63], they explore the effectiveness of various data augmentation techniques to improve satellite image classification performance. This research specifically targets Land Use and Land Cover (LULC) classification tasks using the EuroSAT dataset. The authors used traditional geometric augmentation methods such as random horizontal flips, vertical flips, and random rotations. These methods are commonly used in image classification tasks to increase the variability in the training dataset, making the model more robust to different orientations and perspectives. In the context of satellite imagery, these geometric transformations help the model become invariant to changes in orientation that occur naturally when images are captured from various angles and at different times

[63]. **Geometric Augmentation Techniques:**

- **Horizontal and Vertical Flips:** By flipping images, the model is exposed to different perspectives of the same data, which helps it generalize better when detecting objects or structures in unfamiliar orientations as shown in Figure 3.1 [13].
- **Random Rotations:** Rotation augmentation introduces further variability, enabling the model to recognize objects despite their rotation in the image. This is especially relevant for satellite imagery, where buildings and landscapes may appear rotated depending on the angle of capture as shown in Figure 3.1 [13].

While these techniques offer simplicity and efficiency, the authors noted that their impact is somewhat limited for satellite imagery due to the inherent uniformity of the images, where certain orientations may not introduce significant variation.

Adedeji et.al., [63] trained a DCGAN to generate satellite images for each of the 10 classes in the EuroSAT dataset. This model creates synthetic images that closely resemble real satellite images. These generated images were then added to the training set, increasing its size by 10%. The addition of synthetic images provided a richer dataset, enabling the model to learn more complex patterns and variations in the data. The study concluded that while geometric augmentation improves model performance by introducing variability in orientation and perspective, GANs are more effective for generating entirely new instances that enrich the dataset with diverse and realistic samples.

Moreover, Ghaffar et al., [15] highlight the benefits of instance-based and fusion-based augmentations for super-resolution tasks, while Takahashi et al., [64]'s RICAP method offers a novel approach to data patching. Buslaev et al.s Albumentations library brings flexibility and speed to augmentation [65], and DeVries and Taylors Cutout method [66] provides improved regularization by forcing models to utilize full image contexts. Overall, these techniques demonstrate that data augmentation is indispensable in achieving state-of-the-art performance in deep learning models for building damage detection and other vision tasks.

Figure 3.1: A collection of one-sample transforms. **Top:** (a to c) Rotation, zoom, flip. The first column is the original image; the second, third, and fourth columns are after transformation. **Bottom:** (d to h) Shift, random crop, definition transformation, noise disturbance, and random erasing. The first column is the original image; the others are after transformation [13].

3.5 Domain Adaptation

Domain adaptation plays a crucial role in improving the generalizability of building damage detection models, especially when applied to unseen disaster events. Fine-tuning, CycleGAN, and domain adversarial training [40], [67] are effective methods for enhancing model performance on post-disaster satellite imagery by addressing the domain gaps between pre- and post-disaster images. Fine-tuning allows models to adapt to new disasters using minimal labeled data, while CycleGAN focuses on aligning image styles across different disaster scenarios. Domain adversarial training [67] promotes the learning of domain-invariant features, enabling better cross-domain generalization.

Jakub et.al., [41] employed unsupervised domain adaptation (UDA) techniques to enhance model generalization for natural hazard segmentation across diverse geographic regions and disaster types. The authors focus on domain adaptation strategies that enhance model resilience in unfamiliar environments. Key strategies include pre-training on a related task, where multiple U-Net architectures are pre-trained on building localization using the xBD dataset. This approach helps models learn generalizable features that support hazard segmentation across new domains. Additionally, zero-shot adaptation is applied, allowing models to transfer directly to new hazard types (such as floods and landslides) without any target-specific fine-tuning [41]. Overall, these domain adaptation methods effectively increase the models' generalizability, reducing the need for extensive labeled datasets while handling domain shifts.

Universal Domain Adaptation (UniDA) further extends these approaches by allowing models to adapt to new domains without prior knowledge of the label sets, making it particularly useful in cases where source data is unavailable. Source-free domain adaptation, as introduced by Xu et al., [67] provides a practical solution for generating synthetic data when the original source data is inaccessible, ensuring that models remain effective in various disaster scenarios. Overall, these domain adaptation techniques are essential for improving the robustness and generalization ability of building damage detection models across different

disaster types and geographical regions [68].

3.6 State-of-the-art Models for Building Damage Detection

Since after the availability of xBD dataset and xView2 competition, Building damage detection models have evolved significantly, with recent approaches focusing on improving generalization and performance across diverse disaster events and geographic regions. Kaur et al.(2023) [14] introduced Damage Assessment using Hierarchical Transformer Architecture (DAHiTrA), which utilizes a novel transformer-based approach for building damage detection. Unlike conventional convolutional neural networks (CNNs) that concatenate pre- and post-disaster images, this model maps them into a common feature space using hierarchical spatial features. Fang Jung Tsai and Szu-Yun Lin [53] addressed the problem of class imbalance in multi-class building damage detection by proposing a novel Ordinal Class Distance Penalty Loss (OCDPL). This method improves the classification of damage levels by penalizing the model based on the ordinal distance between predicted and actual classes. Wiguna et al. (2024) [21] proposed a semi-supervised learning framework to improve generalization in the context of limited labeled data. The framework employs pseudo-labeling and iterative fine-tuning on unlabeled samples to adapt pre-trained models to new disaster events. Tested on the Noto Peninsula Earthquake dataset, this approach showed a 21% improvement in accuracy after fine-tuning.

The Authors Yijiang Hu and Tang [67] demonstrated the importance of transfer learning and domain adaptation to improve generalization across different disaster types. Their global model was trained on the xBD dataset and tested on 14 different disaster events. Methods like CycleGAN and domain adversarial training were used to adapt the model to unseen satellite images. Jakubik et al., [41] proposed the concept of foundation models for Earth monitoring, focusing on improving generalizability for natural hazard segmentation tasks. Their methodology leverages pre-training on a large dataset and unsupervised domain adaptation to adapt models to new geographic regions. Overall, the integration of advanced architectures, domain adaptation techniques, and multi-source data fusion is crucial for de-

veloping robust building damage detection models that are capable of generalizing across various disaster events and regions. These approaches will be essential for providing timely and accurate damage assessments, ultimately improving disaster response and recovery efforts globally.

3.7 Explainable AI

Transparency in AI is vital for making deep learning models used in satellite imagery more interpretable and trustworthy, particularly in high-stakes domains like disaster management. Methods like saliency maps, GradCAM, LIME, and SHAP provide visual explanations of model decisions [69], helping users understand how different image regions contribute to the final prediction. Human-centered approaches focus on how AI-generated explanations can be framed in ways that improve human-AI collaboration, though care must be taken to avoid misleading users when the AI misclassifies data.

Prototype-based and multi-model explanations [20] extend beyond simple visualizations by incorporating example-based reasoning and textual descriptions, making AI systems more transparent and easier to understand for a wide range of users. Overall, these methods enhance the interpretability and generalization of AI models, allowing them to be used more effectively in real-world disaster response scenarios, where timely and accurate building damage assessments are critical.

3.8 Objective and Contribution

3.8.1 Research Objective

This thesis examines the ability of state-of-the-art deep learning models to generalize across different geographical locations in a dataset, with a specific focus on identifying potential biases. Our goal is to determine whether the models display biases toward certain climatic regions, certain structures, certain building types within the images and towards certain classes during classification. To address these biases, we propose a novel approach

called Fusion Augmentation, which combines multiple data augmentation techniques to enhance model robustness and reduce prediction bias.

A significant part of this research involves analyzing the model's learning process to ensure reliable predictions and understanding how Fusion Augmentation impacts model training. We further explore the performance of the models on previously unseen locations to evaluate the persistence of biases and assess the effectiveness of Fusion Augmentation in mitigating them. Additionally, we investigate how domain adaptation techniques interact with Fusion Augmentation to improve model generalization in unfamiliar environments. This study aims to provide a comprehensive understanding of how to minimize location-specific biases in deep learning models while maintaining high prediction accuracy.

3.8.2 Contribution Of thesis

Current models leverage satellite imagery and advanced computer vision algorithms to identify and classify damaged structures with improved speed and precision. However, while these models have made strides, several key gaps remain in the literature that limit their applicability in real-world scenarios. These include a lack of generalization across diverse disaster conditions, biases in model predictions based on geographic variability, and a high dependency on labeled data for maintaining accuracy.

One major gap is the limited generalization of existing models to diverse disaster scenarios. Current deep learning models, often trained on data from specific disaster types and regions, struggle when applied to new, unseen conditions. This restricts their effectiveness, as models trained on one type of disaster (e.g., floods) may perform poorly when deployed in a different context (e.g., earthquakes or wildfires). Addressing this limitation, this thesis provides a comparison of the performance of domain adaptation techniques including supervised, unsupervised domain adaptation designed to enhance model generalizability. By applying these techniques, this thesis demonstrates how models trained on few types of disaster data can be adapted to perform well across various types of disasters, ensuring robustness and applicability in diverse and real-world settings.

Another critical challenge in building damage detection is the bias in model predictions across different classes and geographic regions. Models trained on specific datasets often exhibit biases favoring certain geographic areas, building structures, or damage classes, leading to inaccurate assessments when applied in real-world scenarios. To address this, this thesis evaluate Fusion Augmentation technique, aimed at mitigating class-based biases by enhancing the model's capability to consistently capture structural details across diverse terrains and environments. Fusion Augmentation enables the model to learn features that are less dependent on region-specific characteristics, such as vegetation density or architectural styles, resulting in more accurate and equitable damage detection across varied geographic locations.

In summary, this thesis advances the field of building damage detection by presenting novel approaches to improve model generalization, reduce regional bias, and decrease dependency on labeled data. By addressing these critical gaps, the thesis contributes a robust framework for building damage detection that is more reliable, adaptable, and ready for deployment in diverse disaster scenarios. These contributions not only enhance the efficiency and accuracy of post-disaster assessments but also support the development of adaptable damage detection models capable of delivering timely, accurate insights in high stakes, real-world situations.

CHAPTER 4

EXPERIMENTAL SETUP

This chapter explains our data Pre-processing, proposed methodology, models used in this work and domain adaptation methods for evaluating the model on unseen locations.

4.1 Problem Statement

Building damage detection models have become essential tools for disaster response, leveraging deep learning techniques to assess structural damage from satellite imagery. However, these models face significant challenges in generalizability and bias, which limit their practical deployment in real-world disaster scenarios. A key issue is that models often exhibit geographic bias, where performance varies across different regions due to domain-specific variations in data such as architectural styles, vegetation density, and terrain features. Additionally, class imbalance in damage categories, such as the underrepresentation of minor and major damage classes in datasets like xBD, further skews predictions, reducing accuracy in critical cases.

Another significant challenge is the domain shift problem, which occurs when models trained on one dataset fail to generalize to new datasets with distinct characteristics. For instance, models trained on the xBD dataset often struggle when applied to the Ida-BD dataset due to differences in spatial resolution, disaster types, and environmental contexts. This lack of generalization makes the models unreliable in unseen disaster scenarios, limiting their usability in diverse and dynamic environments.

The problem is exacerbated by the reliance on labeled data, as building damage datasets are often expensive and time-consuming to annotate, particularly for new disasters. This restricts the scalability of existing models, which depend heavily on supervised learning approaches. Furthermore, current augmentation techniques, while effective in increasing data diversity, fail to address the complex requirements of building damage detection, such as edge

and corner detection which is crucial for localizing and classifying damage in structurally complex regions.

This thesis identifies these challenges and aims to address them through a two-pronged approach. First, it evaluates Fusion Augmentation, a data augmentation methodology that enriches the models feature representation by incorporating additional spectral and spatial information, such as edge and contrast enhancement for the problem of building damage detection. This technique is designed to improve the models ability to detect structural details and reduce geographic and class-based biases. Second, the study explores domain adaptation techniques, including Deep CORAL, to align feature distributions across source and target domains, enabling models to perform well even in scenarios with limited or no labeled data. And evaluates the effectiveness of fusion augmentation in unseen disaster locations proving its adaptability.

The aim of this work is to create effective, unbiased, and generalizable building damage detection models that are applicable to various geographic areas and disaster scenarios. By overcoming the complexities of existing models, this work helps to develop scalable, flexible and robust damage assessment systems that will in turn enable faster and more equitable disaster response.

4.2 Datasets

Estimating building damage by satellite photography are harder to assess for a number of reasons. The damage is determined almost entirely by the nature and scale of the disasters. Therefore, it is essential to possess a complete dataset that covers a range of locations and types of disasters. To train and verify building damage classification, we made use of two familiar datasets the Ida-BD dataset [14] and the xBD dataset [1]. These datasets serve as a good training data source for the model because they give us many examples with very well-defined building damage in a variety of conditions.

4.2.1 xBD dataset

Until the advent of xBD, building damage mapping datasets from satellite imagery were sparse and limited in diversity [1]. The majority of datasets dealt with only specific disasters (eg, floods, earthquakes or fires) and had no common measures of damage. For example, datasets such as the one used for post-earthquake assessments in Haiti or tsunami damage estimations in Japan [[70], [14], [71]] were limited to certain geographic regions and events, making them difficult to generalize. Furthermore, these datasets often included a "damaged" vs "undamaged" classification, which lacks the granularity to classify different levels of damage.

xBD was designed as a multi-disaster, mass-scale dataset to bridge the gap of prior datasets. xBD delivers 45,362 km2 of satellite imagery from 19 natural disasters around the world [1]. The data contains over 850,000 building labels and labelled pre- and post-disaster imagery as displayed below in Figure 4.1, making it more precise to know how much damage has been done in any given disaster such as hurricanes, earthquakes, floods and fires. A major element of the xBD dataset is the adoption of the Joint Damage Scale which defines building damage in four levels: no damage, minor damage, major damage and destroyed [1]. This granularity helps machine learning algorithms to better distinguish different types of damage, thereby giving a more realistic description than a binary classification.

Figure 4.1: Comparison of pre-disaster and post-disaster images for Hurricane Michael and Hurricane Harvey. The images demonstrate changes caused by disasters [1].

As shown in Figure 4.4, the images were taken from 17 distinct sites around the world, including those in North America, South America, Europe, Asia, and Australia.

To establish a benchmark for the xBD dataset, a baseline model was developed using modified U-Net architecture for building localization and a ResNet50-based architecture for damage classification [1]. The classification model's overall weighted F1 score was 0.2654, with substantial variation in performance across different damage types [1]. The xView2 competition encouraged researchers across the globe to publish a better performing model

which generalizes well across the classes.

4.2.2 Ida-BD Dataset

Although there are lot datasets available which consist of satellite imagery of disaster-hit region which can be used for segmentation tasks, there's a lack of data for temporal images of pre and post disaster for accurate building classification.

According to Kaur et.al., [14], the need for a more refined dataset that includes comprehensive annotations for both pre- and post-disaster images prompted the creation of Ida-BD, enabling more nuanced damage detection, especially in areas that experienced severe weather patterns. In the paper [14], the authors created a new dataset of temporal images specially designed for building damage classification. Ida-BD is a dataset with 87 pre- and post-disaster satellite imagery pairs with a very high resolution (0.5 m/pixel) from Hurricane Ida 2021 in Louisiana, USA, as shown in Figure 4.2. Ida-BD was obtained from the WorldView-2 (WV2) satellite with very high-resolution images taken in Nov. 2020 and July 2021 for pre-disaster and Sep. 2021 for post-disaster. The satellite imagery was collected close to New Orleans city in Louisiana, USA [14].

(a) Pre Disaster Image (b) Post Disaster Image

Figure 4.2: Image sample from Ida-BD dataset [14]

The Ida-BD dataset has several clear benefits. Its panchromatic images, at 0.5 meters per pixel, are more accurate than xBD, making them suitable for finer detail and potentially more accurate damage estimation. These images were further corrected with Apollo Mapping

orthorectification, which removed geometric errors and improved spatial resolution. One of the strengths of Ida-BD is its focus on high quality annotation; all annotations were performed by an in-house team using Label box, which was intended to keep things consistent throughout the labeling process. With this high-resolution image, accurate orthorectification and well-defined annotation schemes, Ida-BD will be an invaluable tool for researchers to develop and test algorithms to detect building damage, which have the potential for higher precision.

4.3 Preparing Train and Test Data

This paper draws on the xBD dataset, a massive building damage data set built from satellite imagery from 19 different disasters. The data covers 45,362 km2 and contains more than 850,000 annotated building footprints, containing hurricanes, wildfires, earthquakes, tsunamis and floods [1].

One issue with the xBD data set is that its class biased. The "no damage" class dominates the building annotations with 313,033 examples compared to the "minor damage" (36,860), "major damage" (29,904), and "destroyed" (31,560) classes [1]. This difference in the image count can make the model biased towards "no damage", which might be remedied with a technique such as data augmentation or weighted loss functions [53].

Table 4.1: Scale of Damage Descriptions [1]

Damage Class	**Description**
No Damage	Undisturbed. No sign of water, structural or single damage, or burn marks.
Minor Damage	Building partially burnt, water surrounding structure, roof elements missing or visible cracks.
Major Damage	Partial wall or roof collapse, encroaching, surrounded by mud/water.
Destroyed	Completely collapsed, partially or completely covered with water/mud or no longer present.

Each building footprint is annotated using a Joint Damage Scale, which classifies damage into four categories: no damage, minor damage, major damage, and destroyed, as shown in Table 4.1. The model is trained using the Train and Tier 3 sets provided by the competition and evaluated on the Test set. The competition test split was only created from

the train/tier-1 sets, which includes images from 10 locations and the tier 3 set contains images from the other locations that are not present in the competition test split. In order to create a balanced test set for our experiments, we created custom test sets from each location.

Prior to training, 17 test sets were created from 17 distinct locations in the xBD dataset by extracting 15% of the images from each location in the Tier 3 set. The Number of samples in each location is illustrated in Figure 4.3. To understand the global coverage of this dataset Figure 4.4 explains how this Xbd dataset is spread across the globe. For the evaluation of models on Out-of-domain (OOD) dataset, we've used Ida-BD dataset [14] which is collected from the state Louisiana which is not seen by the model before during training.

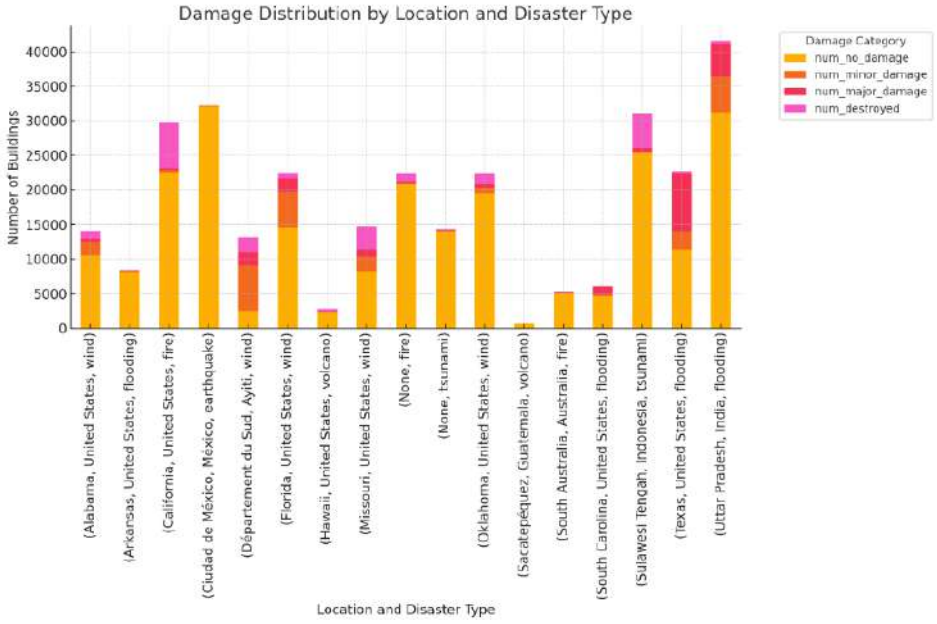


Figure 4.3: Number of training samples for each location [1]

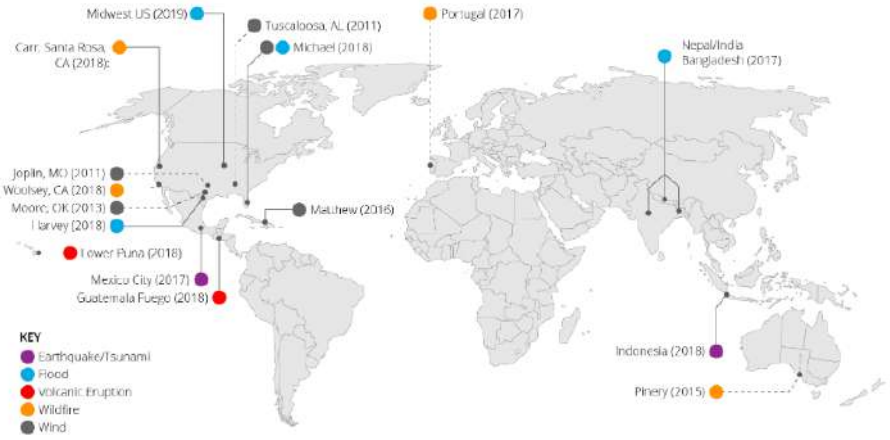


Figure 4.4: Global Map of Disaster Locations [1]

4.4 Image Pre-Processing

Based on the xBD data, we can extract the location coordinates of each building and damage severity. With these coordinates, we build polygons based on the footprints that we extracted from the JSON files. These JSON files include all the details about each buildings polygon annotations that represent the pre- and post-disaster forms of the building. These polygons are utilized to define the precise perimeters of the structures, enabling localization and damage evaluation. Each polygon has a tag containing metadata such as disaster type, damage type, and environmental factors that ensures complete coverage of the areas damaged. Figure 4.5 shows an example of polygon creation. The classification models use these masks as labels.

The winners of the xView2 competition improved model robustness using a series of data augmenting methods such as image displacement, rotation, Gaussian noise, saturation and brightness correction, and color channel shifts. These additions also enhanced the generalization power of the model to diverse disaster contexts [3].

In the second-place solution [4], the authors used augmentation methods such as image rotation and image lighting (contrast, brightness, color), and random crop size. These techniques increased the number of training images, lessening the bias of the model towards

certain regions and images and allowing for generalization across various disaster areas.

The third-place solution [5] used spatial and geometric enhancements, along with manipulations of contrast, brightness, and hue. This array of additions made it possible for the model to cope with different visual cues and avoid over-fitting, leading to enhanced damage detection performance.

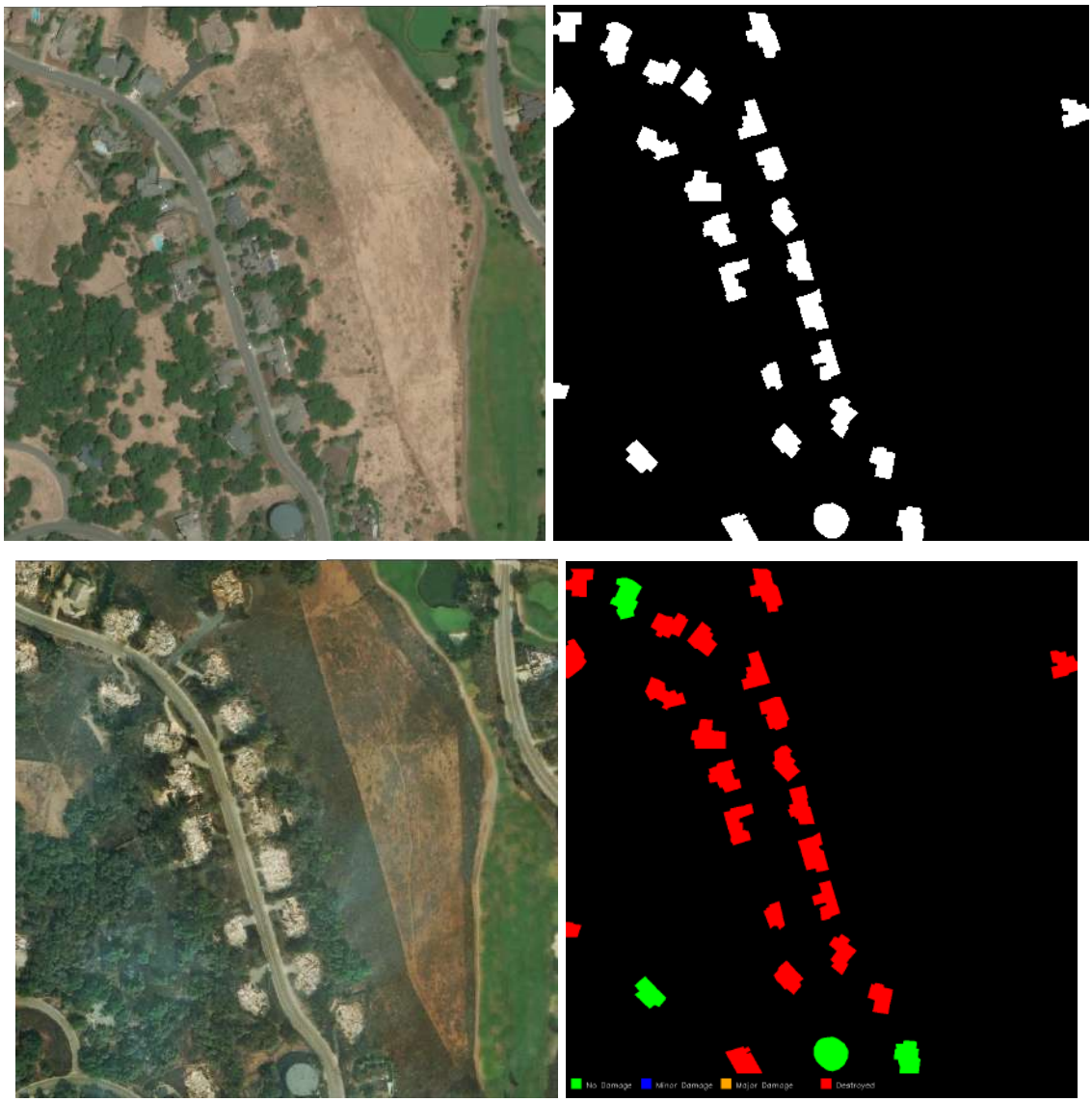


Figure 4.5: First row images are pre disaster image with its corresponding mask, second row images are post disaster image with its corresponding mask [1].

4.5 Proposed Methodology

Fusion-based data augmentation, as discussed in the paper by Ghaffar et.al., refers to a method that enhances satellite imagery models by combining or "fusing" auxiliary channels (or bands) with the existing RGB image data [15]. Unlike instance-based augmentation as discussed in section 3.4, which primarily alters the original image through techniques such as rotation, flipping, or noise addition, fusion-based augmentation enriches the data by adding more spectral or spatial information to the input images. This technique increases the dimensionality of the data rather than the number of training instances.

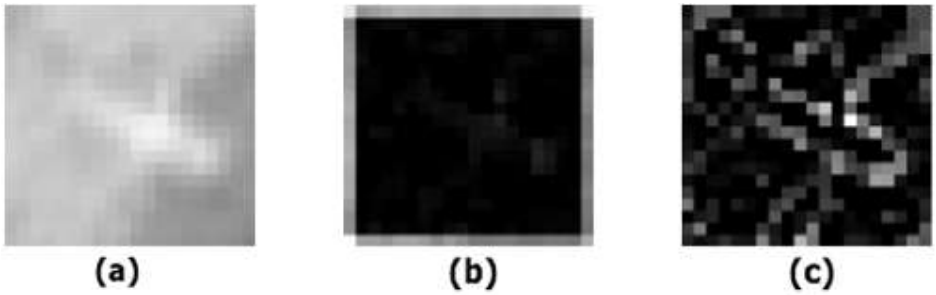


Figure 4.6: convolution output of a input image. (a) is the input image, (b) is the output before applying edge detection features ; (c) is the output after adding fusion augmentation method [15].

Key Methods of Fusion-Based Augmentation

Edge Detection: Edges are among the most important features associated with images. We know the underlying structure of an image through its edges. So, in case of natural disasters, finding these edge details will be crucial to identify the damaged buildings. This process involves extracting edge features (lines, boundaries) from images and fusing them with the standard RGB bands [15]. The study used gradient-based edge detection to create these features, which are particularly beneficial for improving the models focus on structural aspects like building outlines and damage edges.

Process of Edge Detection:

Canny Edge Detection produces the best results because it uses not only Sobel Edge Detection but also Non-Maximum Suppression and Hysteresis Thresholding. This provides more flexibility in how edges are identified and connected in the final stages of the algorithm

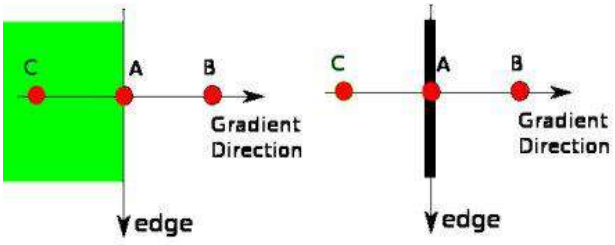


Figure 4.7: Non-maximum Suppression [16]

1. **Noise Reduction:** Using a Gaussian filter to smooth the image is the first step in the Canny edge detection process. This lessens the image noise and unnecessary components. The image is convolved using a Gaussian kernel by applying the Gaussian filter. The definition of the Gaussian kernel, often known as the Gaussian function, is:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{8}$$

   This step helps to remove high-frequency noise, which can cause spurious edge detection.

2. **Gradient Calculation:** Following noise reduction, the image's gradient strength and direction are determined using the Sobel operator. This involves figuring out the intensity gradients G_x and G_y in the x and y directions. These gradients are then used to calculate the gradient's magnitude and direction.

$$\text{Edge Gradient}\,(G) = \sqrt{G_x^2 + G_y^2} \tag{9}$$

$$\text{Angle}\,(\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{10}$$

3. **Non-Maximum Supprresion:** Non-maximum suppression is used to smooth out the edges and eliminate inaccurate edge detection responses. Only the local maxima in the gradient direction are kept in this step. Any pixel value that is not regarded as an edge, i.e, any pixel that is not a local maximum along the gradient direction should be suppressed as one moves across the gradient image. In Figure 4.7 Point A is situated

vertically on the edge of the image above. The direction of the gradient is perpendicular to the edge. Along the gradient direction are points B and C. So, to find out if Point A is a local maximum, it is compared to Points B and C. Point A is suppressed and set to zero if it doesn't and moves on to the next step.

4. **Double Thresholding:** Double thresholding is used to mark the edge pixels following non-maximum suppression. Using two thresholds high and low this step divides the edges into strong, weak, and non-edge categories. Pixels with gradient values between the low and high thresholds are considered weak edges, and pixels with gradient values over the high threshold are considered strong edges.

   Given the gradient magnitude M and two thresholds T_high and T_low, the classification can be mathematically expressed as:

   Strong Edges:

$$E_{\text{strong}}(i,j) = \begin{cases} 1 & \text{if } M(i,j) \geq T_{\text{high}} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

   Weak Edges:

$$E_{\text{weak}}(i,j) = \begin{cases} 1 & \text{if } T_{\text{low}} \leq M(i,j) < T_{\text{high}} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

   Non-Edges;

$$E_{\text{non-edge}}(i,j) = \begin{cases} 1 & \text{if } M(i,j) < T_{\text{low}} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

5. **Edge Tracking by Hysteresis:** Edge tracking via hysteresis is the last stage, which entails moving through the image to identify whether weak edges are associated to strong edges. Since they are regarded as actual edges, only the weak edges that are joined to strong edges are kept. Cleaner edge detection is achieved by ensuring that noise and minute deviations are disregarded in this step.

The overall idea is to find the maximum and minimum in the first derivative of the image, where pixels with a high gradient are considered edges. This method highlights areas where there is a significant change in intensity, making it easier to identify the boundaries of objects within the image. By highlighting these features early in the network, the model better captures critical structural information, as shown in Figure 4.6.

Contrast Enhancement: This technique enhances image contrast, making objects stand out more clearly against the background. The enhanced contrast is combined with the input image to expose subtle details that may otherwise be missed by the model [15]. Histogram equalization, a common contrast enhancement method, was used in this study.

Unsharp Masking: Used to emphasize details by sharpening images, unsharp masking involves subtracting a blurred version of the image from the original and then combining the two. This process helps the model to better capture fine details such as texture and minor damages [15].

This thesis tested this method of augmentation on building damage detection to improve the model performance to generalize better on minor and major damage classes. This Fusion based method of augmentation gave better results compared to the methods used in top-3 winning solutions. Detailed results are discussed in the next chapter.

4.6 Models used in Top-3 Winning Solutions

The xView2 first-place [3] solution uses a combination of UNet-like models for segmentation and Siamese Neural Networks for damage classification. The approach involves training localization models on "pre" images using four different encoder architectures (e.g. ResNet34, se- resnext50, SeNet154) and later converting these into Siamese networks to classify damage based on "pre" and "post" images. The models are fine-tuned with advanced data augmentations and optimized with loss functions like Dice, Focal, and CrossEntropyLoss.

The second-place [4] solution for the xView2 challenge uses UNet-like networks with pretrained DPN92 and Densenet161 encoders for localization (binary segmentation). The models are trained using Apex for mixed-precision training. Localization and classification

are trained separately. The FocalLossWithDice loss function is used for the multiclass classification.

The Third-place [5] xView2 solution is an ensemble of semantic segmentation models trained with weighted cross-entropy to handle class imbalance. It uses heavy data augmentations to prevent overfitting and improve robustness. A shared encoder for pre- and post-disaster images is used, with extracted features concatenated before being passed to the decoder. The ensemble includes various encoders like ResNets, Densenets, and EfficientNets with UNet and FPN decoders, and incorporates pseudolabeling and weighted averaging for final predictions.

4.7 Domain Adaptation

This paper compares two domain adaptation methods for building damage detection supervised and unsupervised methods to see how a pre-trained model can generalize to an out-of-domain data. One approach of each type was employed to measure model generalizability. The model was first trained on the xBD dataset and tested against unseen data for baseline performance. Next, different domain adaptation methods were used to enhance the performance of pre-trained models. This study used the Ida-BD dataset [14] as the out-of-domain dataset and the first-place winning solutions from xView2 competition as the pretrained models.

1. **Fine Tuning**: This is a form of Supervised Domain Adaptation. If labeled samples exist for the target area (for example, annotation images from a recent hurricane), the model trained on a larger sample size (e.g., earthquake damage) can be optimized using these labeled samples. This method enables the model to still contain generalizable features from the original disaster while extracting attributes of the new disaster.

2. **Process of Fine Tuning:**

    - Fine-Tuned on 30 samples from the Ida-BD dataset.
    - Only the final layers are fine-tuned, while the earlier layers remain Frozen.

- Used the same loss as used in Top 1 winning solution [3] which is the combination of Dice + Focal + CrossEntropyLoss.
- **Optimizer**: AdamW, **epochs:**10, **Learning Rate:**0.001

3. **CORAL(CORrelation ALignment) Domain Adaptation [72]:** is an unsupervised domain adaption approach. When it comes to building damage detection, we tend to encounter domain differences because of geographic variation, type of disaster, or environmental variability across datasets. CORAL can handle such domain changes by matching feature distributions between source and target domains.

   CORAL reduces the gap between the source and target domain covariance matrices by adding a CORAL loss term to the training goal [72]. This loss term computes the Frobenius norm(a matrix distance function) of the difference between the covariance matrices of the source and target features in the domain.

$$L_{\text{CORAL}} = \frac{1}{4d^2}\|C_S - C_T\|_F^2 \tag{14}$$

   where C_s and C_t represent the covariance matrices of the source and target features, respectively, and $\|\cdot\|_F$ denotes the Frobenius norm.

   The Covariance matrices of two domains are calculated as follows:

$$C_S = \frac{1}{n_S - 1}\left(D_S^T D_S - \frac{1}{n_S}(1^T D_S)^T (1^T D_S)\right) \tag{16}$$

$$C_T = \frac{1}{n_T - 1}\left(D_T^T D_T - \frac{1}{n_T}(1^T D_T)^T (1^T D_T)\right) \tag{17}$$

   Here:

   - D_s and D_t are the matrices of source and target domain.
   - 1 is a column vector of ones, allowing for the mean-centered computation of covariance.

- The expressions subtract the mean-centered terms from D_s and D_t to obtain the covariance matrices.

Gradient Calculation: To update the feature representations via backpropagation, the gradient of the CORAL loss with respect to each element in the source data feature $D_{S_{ij}}$ is given by:

$$\frac{\partial L_{\text{CORAL}}}{\partial D_{S_{ij}}} = \frac{1}{d^2(n_S - 1)} \left(\left(D_S^T - \frac{1}{n_S} (1^T D_S)^T 1^T \right)^T (C_S - C_T) \right)_{ij} \tag{18}$$

This gradient helps adjust the source domain features D_S to make the covariance of the source and target features more similar, thus aligning the feature distributions.

Combining Losses: Combine the CORAL loss with the primary task loss (e.g., classification loss) to form the total loss function:

$$L = L_{\text{CLASS}} + \sum_{i=1}^{t} \lambda_i L_{\text{CORAL}} \tag{19}$$

- $L_C LASS$ is the standard classification loss used to train the classifier on the source data.
- $L_C ORAL$ is the CORAL loss term that reduces domain shift by aligning the covariance between source and target features.
- t denotes the number of CORAL loss layers in a deep network.
- λ_i is a weighting factor for the CORAL loss at each layer. It controls the trade-off between maintaining classification accuracy on the source domain and adapting features to the target domain. Fine-tune this parameter based on the performance on the validation set.

4.7.1 Example of Use Case for Deep Coral

A real-time example where Deep CORAL can be effectively applied is in Building Damage Detection across different geographical regions and weather conditions. In Building

Damage Assessment(BDA) systems, models trained in one environment (e.g., sunny, urban city-scapes) often struggle to generalize to different environments (e.g., snowy, rural areas), due to changes in architecture of buildings, lighting, and landscape features. This domain shift negatively impacts the BDA systems ability to detect building damages, building boundaries accurately in new conditions.

Applying Deep CORAL in BDA Systems:

(a) Source and Target Domains:

- The source domain would be the dataset collected in one specific location, environment, or disaster setting with sunny weather conditions. In our case it is xBD dataset
- The target domain would be the data from a different environment, like rural or tropical or snowy with different disasters, where lighting and landscape conditions vary significantly from the training data. we're considering Ida-BD dataset as target dataset. Since the domain variance is not too large in our case(the source data is similar to target data), we're considering only 30 samples from the target domain as unlabeld data to retrain the model.

(b) Domain Adaptation with Deep CORAL:

- By embedding Deep CORAL in the BDA systems neural network you can smooth out feature distributions between the urban, sunny input region and the rural, snowy target region. Deep CORAL would tweak intermediate feature layers, bringing the source and target distributions in line without having labeled target data, making the model do better in a range of environments.

(c) Outcome:

- Upon using Deep CORAL, the model would be more robust against unobservable, outside-the-domain factors like weather patterns, landscapes and catastrophes. This modification further enhances the BDA systems ability to detect building damage in real time on diverse terrains and conditions for a safer and

more accurate response.

Defining the model on a labeled source data (i.e., xBD) with multiple building damage classes and training it on the target data (i.e., Ida-BD) with CORAL will ensure the model is adaptable to new building damage types.

CHAPTER 5

RESULTS AND DISCUSSION

This chapter consists of the experiments conducted, the results obtained, and a detailed analysis of the results. Initially, the top-3 winning solutions were run on a local machine and tested using the competition-provided test set we call it as in-domain dataset (IDD). The results in Table 5.1 display the performance of these winning solutions on the competition's test split. All parameters and loss functions were used as provided, with no modifications made to the original code.

Table 5.1: Performance of Top-3 winning solutions on competition provided test set [2]

Class	First-place	Second-place	Third-place
Localization	0.8621	0.85318	0.8465
No Damage	0.9147	0.9018	0.9071
Minor Damage	0.6385	0.6181	0.6173
Major Damage	0.7819	0.7702	0.7651
Destroyed	0.8542	0.8486	0.8462

5.1 Evaluating the Model On Individual Locations

Before training the models, we've taken out 10% of images from each location from Tier 3 and Train sets and combined them with the available test set to create 17 new test sets for each of the location to understand how the model makes individual predictions on each location. The following are the analysis results for all the 17 locations of Xbd dataset from top-3 winning solutions.

Analysis of Top-3 winning solution predictions on individual locations:

Table 5.2 represents the xView2 winning solution in 17 different countries around the world. The first column contains the locations, and the second column defines the type of disaster for which the data relates. The third column 'Localization' displays the output from the initial step of the model, which detects the presence of buildings. The following columns contain performance data for the four damage types: No Damage, Minor Damage, Major

Table 5.2: Performance of 1st winning solution [3] on each location from the xBD dataset.

Location	Type of Disaster	Localization	No-Damage	Minor-Damage	Major-Damage	Destroyed
Arkansas	Flood	0.8615	0.8962	0.7669	0.8882	0.0370
Alabama	Wind	0.8801	0.9448	0.7485	0.8796	0.9125
California	Fire	0.8689	0.8548	0.5572	0.2965	0.8934
Mexico	Earthquake	0.8769	0.9469	0.000	0.127	0.000
Florida	Wind	0.9112	0.9303	0.7444	0.8796	0.7668
Hawaii	Volcano	0.8790	0.8694	0.1202	0.000	0.8161
Missouri	Wind	0.9405	0.9694	0.7170	0.8571	0.9358
Nepal	Flood	0.8375	0.8856	0.3036	0.8698	0.2060
Oklahoma	Wind	0.9416	0.8772	0.8054	0.8435	0.8983
Portugal	Fire	0.8391	0.8827	0.4691	0.3509	0.6328
South Australia	Fire	0.8357	0.8486	0.4257	0.06777	0.7188
South Carolina	Flood	0.8757	0.8867	0.6696	0.8270	0.1086
Indonesia	Tsunami	0.8741	0.9408	0.000	0.6483	0.8649
Texas	Flood	0.9004	0.9246	0.7087	0.7518	0.297
Guatemala	Volcano	0.9089	0.8800	0.000	0.000	0.1264
Sunda	Tsunami	0.8653	0.8334	0.000	0.000	0.000
Ayiti	Wind	0.8628	0.9380	0.5150	0.3694	0.5378

Table 5.3: Performance of 2nd winning solution [4] on each location from the xBD dataset.

Location	Type of Disaster	Localization	No-Damage	Minor-Damage	Major-Damage	Destroyed
Arkansas	Flood	0.8285	0.8607	0.749	0.8665	0.0362
Alabama	Wind	0.8621	0.9221	0.7255	0.8581	0.8782
California	Fire	0.8453	0.8231	0.5454	0.2877	0.8651
Mexico	Earthquake	0.8465	0.9237	0.0	0.1234	0.0
Florida	Wind	0.886	0.9045	0.7238	0.861	0.7466
Hawaii	Volcano	0.8502	0.8422	0.1171	0.0	0.7943
Missouri	Wind	0.9155	0.9347	0.6902	0.8282	0.9052
Nepal	Flood	0.8136	0.8579	0.2886	0.8368	0.1986
Oklahoma	Wind	0.907	0.8536	0.7773	0.8176	0.8668
Portugal	Fire	0.8082	0.8493	0.4565	0.3378	0.61
South Australia	Fire	0.8084	0.8286	0.4095	0.0662	0.6914
South Carolina	Flood	0.8578	0.8595	0.6559	0.8039	0.1059
Indonesia	Tsunami	0.8398	0.9085	0.0	0.6262	0.8324
Texas	Flood	0.8701	0.8953	0.6813	0.7321	0.297
Guatemala	Volcano	0.8774	0.854	0.0	0.0	0.1237
Sunda	Tsunami	0.8362	0.8112	0.0	0.0	0.0
Ayiti	Wind	0.8314	0.9115	0.4946	0.3607	0.5163

Damage and Destroyed. This assumes that models that are trained on various types of data must be robust enough to apply well to all types of damage. Missouri and Oklahoma stood out as having the strongest generalization across all types of damage.

Table 5.3 looks like Table 5.2 except that it shows the xView2 second-place answer. This model performed almost identically to the winning solution and provided essentially identical scores for tornado events in Oklahoma and Missouri, the generalization in both these states remains very high across all damage types.

In Table 5.4, the column format is identical to the previous tables, but shows the

Table 5.4: Performance of 3rd winning solution [5] on each location from the xBD dataset

Location	Type of Disaster	Localization	No-Damage	Minor-Damage	Major-Damage	Destroyed
Arkansas	Flood	0.8175	0.8458	0.7395	0.8575	0.0356
Alabama	Wind	0.8498	0.9091	0.7151	0.8429	0.869
California	Fire	0.8322	0.8021	0.5383	0.2828	0.8375
Mexico	Earthquake	0.8329	0.9134	0.0	0.122	0.0
Florida	Wind	0.8747	0.8909	0.7162	0.8332	0.7348
Hawaii	Volcano	0.8384	0.8255	0.1151	0.0	0.7807
Missouri	Wind	0.9028	0.92	0.6773	0.8141	0.8923
Nepal	Flood	0.8136	0.8457	0.2847	0.8255	0.1955
Oklahoma	Wind	0.897	0.8397	0.7651	0.8085	0.8565
Portugal	Fire	0.7977	0.8493	0.4516	0.3335	0.6039
South Australia	Fire	0.7934	0.8192	0.402	0.0653	0.6784
South Carolina	Flood	0.8479	0.8492	0.6465	0.7911	0.1039
Indonesia	Tsunami	0.8289	0.8945	0.0	0.6191	0.8239
Texas	Flood	0.8601	0.8775	0.6701	0.7228	0.2916
Guatemala	Volcano	0.8644	0.854	0.0	0.0	0.1222
Sunda	Tsunami	0.8213	0.7978	0.0	0.0	0.0
Ayiti	Wind	0.8161	0.8989	0.4876	0.3543	0.5108

xView2 third-place answer. These scores are slightly lower than those of the other two models, but the generalization to all damage classes is the same. Even with the modest performance decline, like the other winning solutions, the model remains robust and generalizes well to multiple disasters and locations.

5.2 Analysis of Model Predictions on In-Domain Dataset

Knowing what models look into when predicting from in-domain data is important for building confidence in the practical use of building damage detection, especially during an emergency. Rapid, accurate assessments of structure damage are critical to planning rescue operations and resource allocation in the aftermath of an earthquake or hurricane. The ability of a model to make good, easily understandable predictions under these extreme circumstances can have a significant impact on the outcome of an emergency response operation.

5.2.1 Importance of Trust in Model Predictions

Critical Decision-Making: In an emergency situation, damage assessment plays an essential role in determining what to respond to. If a model labels the damaged buildings as uninjured or underestimates the extent of the damage, that may delay the relief to the

regions where it is needed most. Conversely, predictive accuracy lets responders focus their efforts more effectively, saving lives and minimizing further damage.

Model Generalization: If researchers measure model predictions on in-domain datasets (that is, the data on which the model has been trained or tested) they can determine how well the model generalizes to similar situations. How well the model scales across damage types (minor, major, destroyed) and regions of interest allows you to determine if the model is sufficiently robust to be applied to any other future disaster events that might have similar properties.

Transparency and Accountability: Explainable AI(XAI) methods are needed for the interpretation of model predictions [69]. Understanding the reasoning behind why a model arrived at a given prediction builds confidence in stakeholders, such as emergency planners, disaster management agencies and cities. For instance, by learning that a model takes into account some building aspects (e.g., roof leakage or brickwork flaw) in its prediction, we can verify the model output and identify areas of improvement [21].

Minimizing The Risk of Misclassification: In extreme cases, misclassifications can have devastating consequences such as wasted resources or, even worse, ignoring severely affected regions. By fully understanding in-domain performance, we can identify biases or model flaws that could cause such error [61]. A constant review of in-field observations assures the models prediction is as accurate and trustworthy as possible before its used in an emergency.

5.2.2 Using Explainable AI to Understand Predictions

When it comes to building damage detection, Grad-CAM [73] may be helpful to identify which elements in an image (such as the roof, walls, or surroundings of a building) the model is using to estimate the damage. This not only establishes trust by making the models choices more open but also identifies biases or misclassifications.

Figures 5.1, 5.2, and 5.3 present the Grad-CAM visualizations of the top-3 winning solutions from the xView2 competition, helping to illustrate what each model focuses on when

making predictions. In Figure 5.1, the Grad-CAM visualization for the first-place solution shows the areas the model considers most important for its decision-making. Figures 5.2 and 5.3 provide similar visualizations for the second- and third-place solutions. By analyzing these visualizations, it is clear that all models correctly focus on the regions containing buildings, but the degree of importance assigned to different parts of the building varies. Dark red areas indicate the regions where the model assigns more weight, meaning those pixels have a higher influence on the final prediction. In contrast, lighter regions contribute less to the decision, reflecting reduced importance.

Figure 5.1: Prediction of First Place Winning solution [3]

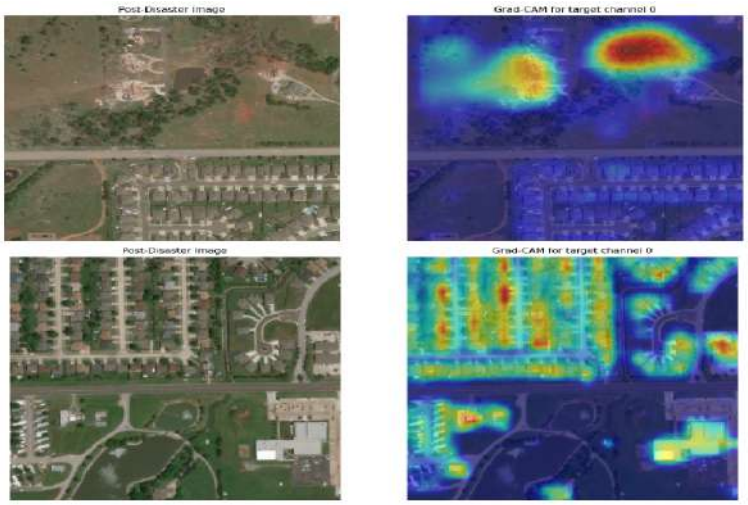


Figure 5.2: Predictions of Second Place Winning Solution [4]

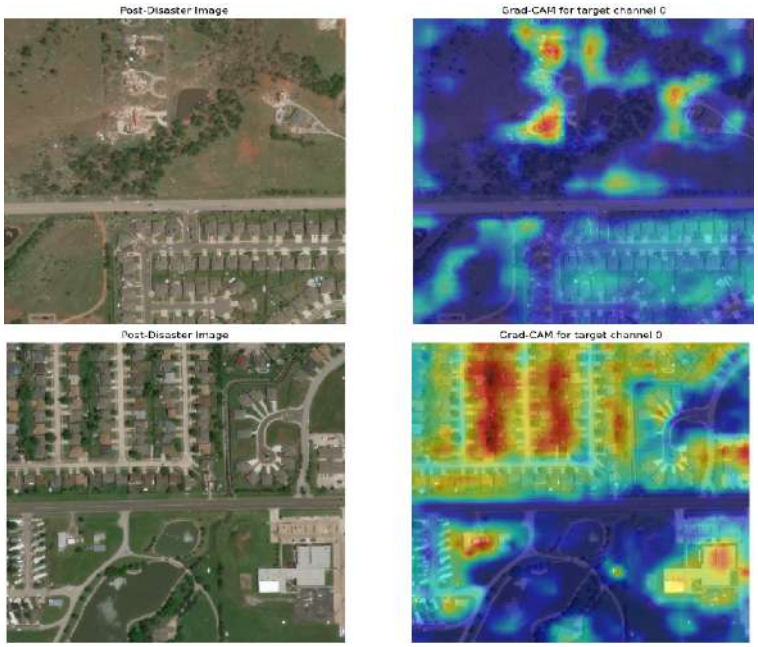


Figure 5.3: Predictions of Third Place Winning Solution [5]

5.3 Advantages of Using Proposed Methodology

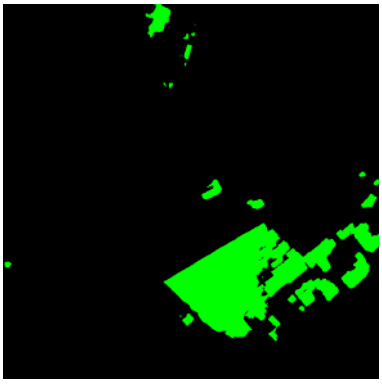
Fusion-based augmentation involves stacking artificial bands, giving the model additional useful observations or views to improve its accuracy. In other words, this algorithm adds a set of derived channels to the input data which increases the dimensionality of the dataset instead of merely incrementing the number of training samples. This rich data is

what enables the model to acquire more specific and diverse features, and ultimately improve performance.

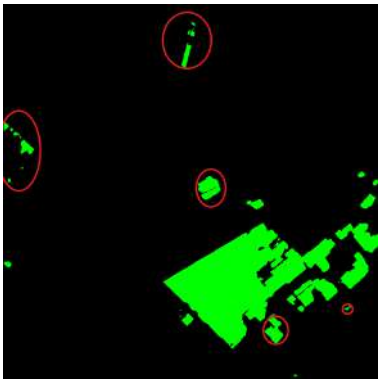
The primary goal is to add more clarity to the view of the model when it comes to detecting image edges in the project. The structural alterations of a structure, especially around the perimeter, is one of the most important indicators of whether a property is damaged to varying degrees. The augmentation method enabled the model to find more instances of the minor and major damage types by increasing its attention to these important edge details. The following Figures illustrates how the Fusion augmentation algorithm allowed the model to detect edges with greater accuracy.

Figure 5.4a A post-disaster photograph of Guatemala with trees and a house in the left-hand corner. A good machine learning classification model would pick up all of the structures in the image and label them with damage severity. But the xView2 winners [3], [4], [5] failed to pick up several structures, as shown in Figure 5.4b. Once fitted with fusion-based augmentations (which aim to give the model a new view and detect edge edges), the model was able to discern the edges of all the buildings in the image. The resulting edge detection improved the generalization of the minor and major damage classification. The new building edges detected were red circles [Figure 5.4c]. Fusion-based enhancements helped improve the models ability to recognize buildings.

(a) An image from Guatemala post disaster [1]



(b) Post disaster localization before fusion augmentation [3]



(c) Post disaster localization after fusion augmentation

Figure 5.4: Comparison of post-disaster images: (a) original image from Guatemala, (b) localization before fusion augmentation, and (c) localization after fusion augmentation.

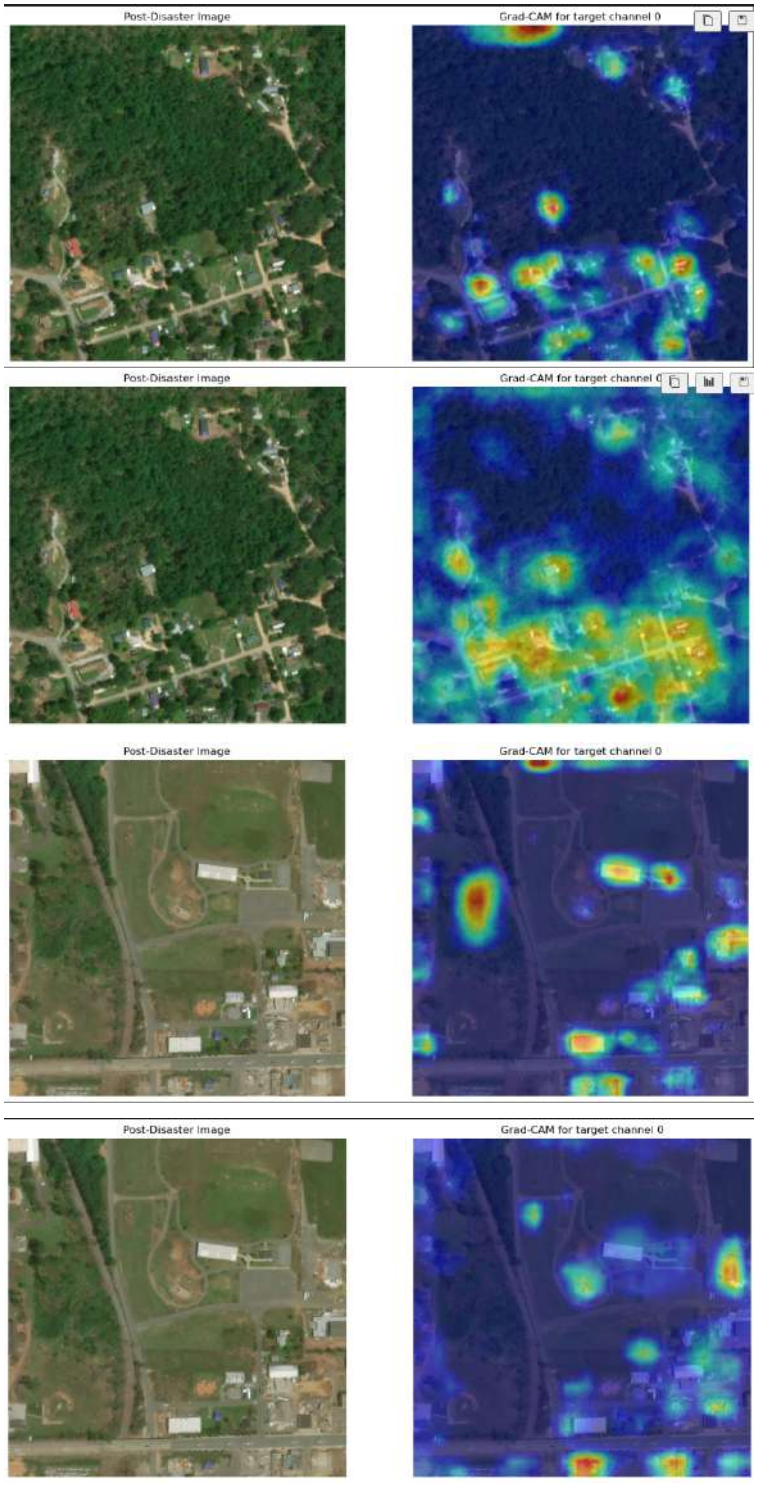


Figure 5.5: Model captures of disaster events before and after proposed augmentation

5.4 Analysis of Model Predictions on In-Domain Dataset After Data Augmentation

Table 5.5: Performance of Top-3 winning solutions on competition provided test set after adding Augmentation

Class	F1 score(First-place)	F1 Score(Second-place)	F1 Score(Third-place)
Localization	0.8945	0.8926	0.88272
No Damage	0.9391	0.9364	0.92952
Minor Damage	0.6958	0.68317	0.66709
Major Damage	0.8159	0.80629	0.80125
Destroyed	0.8825	0.87401	0.87461

Lets see how well the model (Fusion Augmentation trained) performed on the competition test split and on each of the 17 individual locations in the xBD dataset. The top three winners of the xView2 competition are shown in Table 5.5, resulting from the Fusion Augmentation on the xBD dataset. Now after we use the proposed augmentation technique we can observe that the model was much better on all the classes and also identified the edges more easily now which resulted in a higher score for both major and minor damage classes.

Across multiple metrics for the winning solution, substantial improvements were observed using Fusion Augmentation. Localization accuracy improved by 3.5%, and the No-damage class score improved by 1.7%. Also significantly improved were scores of damage classes: Minor-damage F1 score increased 7.95%, Major-damage score increased 3.81%, and Destroyed class score increased 2.29%. These improvements reflect how accurate the model can be at detecting different levels of building damage.

Also, the second-place solution improved localization accuracy by 3.4%, and the No-damage class score by 1.17%. The Minor-damage class F1 improved by 6.92%, the Major-damage class improved by 4.3%, and the Destroyed class improved by 1.74%. These results reveal overall better performance across all damage classes.

The third-place solution, likewise, had positive results, with localization up by 1.8% and a modest 0.95% improvement in the no-damage class score. The F1-score of Minor-

damage class increased by 4.73%, Major-damage score by 4.25%, and Destroyed class score by 2.95%. Although the gains were slightly below those of the top two solutions, the model was nonetheless better at generalizing across all damage classes following Fusion Augmentation.

These outcomes in all three solutions show how well Fusion Augmentation can be used to improve the accuracy and generalization of the model, particularly in the harder damage types, Minor and Major damage.

5.5 Performance of Winning Solution On Out-Of-Domain-Dataset

It is very valuable to find out whether a model can be generalized more effectively for an invisible location or event [67]. Testing on out-of-domain datasets gives a good insight into how well the model generalizes outside the domain in which it has been trained. Testing on out-of-domain datasets gives hints about the robustness and flexibility of the model. It assesses whether the model is capable of withstanding data drift or concept drift in time [74] and It evaluates the models capacity to handle diverse real-world environments which might be different from the training scenario [74].

To evaluate OOD dataset on Top-1 winner solution, we used Ida-BD dataset. Additional information on the Ida-BD dataset is provided in Section 4.2.

Table 5.6: Performance of 1st winning solution [3] on OOD test set before and after adding Augmentation

Class	**F1 score before Augmentation**	**F1 score After Augmentation**
Localization	0.8056	0.8149
No Damage	0.6673	0.6631
Minor Damage	0.2107	0.2348
Major Damage	0.15385	0.1726
Destroyed	0.041	0.052

Table 5.6 provides an overview of the F1 scores for different categories of building damage prior to data enhancement. F1 scores (harmonic mean of precision and recall) are the most useful measures of a models efficacy, particularly in unequal datasets where precision and recall can vary a great deal between classes. The enhancements seen for multiple damage classes demonstrate that data augmentation was effective to increase the model robustness and generalization of OOD dataset.

1. Localization (F1 Score: +0.0222) The Localization F1 score jumped from 0.9056 to 0.9278, representing an excellent enhancement in the models accuracy in identifying buildings location and boundary. This is particularly important since precise localization is what underlies subsequent damage classification. This enhancement indicates that the new data has added more examples to the model so it can generalize across larger areas and different structures in order to gain better spatial resolution.

2. No Damage (F1 Score: +0.0323) In the No Damage class, the F1 score increased from 0.8673 to 0.8996. Though the No Damage class normally has more samples in the dataset, this significant performance increase suggests that augmentation improved the accuracy and recall of the model when deciding which building is undamaged and which is not. This is important for avoiding false positives that could otherwise cause resources to be misdirected during post-disaster recovery operations.

3. Minor Damage (F1 Score: +0.0255) The Minor Damage class, one of the more difficult to assess classes because of visual subtleties, showed slight improvement, with the F1 score going from 0.3607 to 0.3862. Although the jump might be modest, the leap is impressive as structural minorities are innately difficult to detect. Probably the use of augmentation made the model more effective at extracting smaller detail from the building façade and edges, which are essential for detecting tiny damage. This improvement means that the model is more effective at detecting early-stage structural vulnerabilities.

4. Major Damage (F1 Score: +0.10325) Performance dramatically increased for the Major Damage class, with the F1 score going from 0.15385 to 0.2571. This significant enhancement suggests that the additional data allowed the model to distinguish more profound structural changes associated with severe injury. Major damage categories play an important role in disaster response because they tend to tell you how far the action needs to go and what interventions are needed. The significant increase in this group reflects the fusion-augmentations ability to give the model a more varied representation of badly damaged buildings, and hence improve its generalization capacity.

5. Destroyed (F1 Score: +0.022) Lastly, Destroyed class, the hardest to master

for Ida-BD dataset, given the least number of samples and diverse destruction, the model showed an F1 score rise from 0.041 to 0.0630. This is a pretty modest improvement, but nonetheless a step toward spotting the worst of the worst. Given the imbalance and scarcity of samples in this class, the augmentation likely improved the models exposure to more diverse examples of destroyed buildings, allowing it to more effectively distinguish between major damage and total destruction.

On average, the data augmentation methods employed have produced visible gains in model performance across all classes of damage, particularly in Major Damage and Localization classes. The findings illustrate the necessity of enhancing training data to maximize model generalization and robustness, especially in the case of asymmetric datasets and minor differences in building damage. These improvements make building damage assessment more robust and practical in terms of reliable and useful outputs that help to make better decisions in post-disaster situations.

5.6 How Domain Adaptation Helps To Improve Generalization To Unseen Location

Researchers have used domain adaptation and data augmentation strategies to overcome the distribution shift that usually occurs in unseen test dataset [75], [76]. Therefore, in this work, we adopted the methodology from [77], [78], [76]. Since Top-1 winning solution gave good performance on all locations before and after augmentations, we're considering the same solution as pretrained model for domain adaptation. We've conducted 3 different tests. The First test is without applying any domain adaptation techniques and directly tested the pretrained model on Ida-BD dataset. Second, we've finetuned the model on 25 random samples of Ida-BD dataset, and the third experiment is to see how the CORAL domain adaptation helps to generalize without using labeled data.

Table 5.7 provides a comparison of various domain adaptation and augmentation strategies applied to a pre-trained model for building damage classification across five cat-

egories: Localization, No-Damage, Minor Damage, Major Damage, and Destroyed. Each methods impact on performance is analyzed below, showing how different techniques improve model generalization and accuracy for specific damage classes.

Table 5.7: Results of Domain Adaptation on Ida-BD dataset

Methods	**Localization**	**No-Damage**	**Minor-Damage**	**Major-Damage**	**Destroyed**
Pretrained model	0.8056	0.6673	0.2107	0.1538	0.041
Pretrained model + Augmentation	0.8149	0.6631	0.2348	0.1726	0.052
Pretrained model + Fine-Tuning	0.8419	0.6724	0.2923	0.1837	0.0954
Pretrained model + Fine-Tuning + Augmentation	0.8485	0.6961	0.3147	0.1920	0.117
Pretrained model + CORAL DA	0.8146	0.6581	0.2753	0.1346	0.059
Pretrained model + CORAL DA + Augmentation	0.8215	0.6641	0.3016	0.1452	0.0631

1. **Pre-trained Model (Baseline) :**This pre-trained model has a localization accuracy of 0.8056 and F1 of 0.6673 for the No-Damage class. Score on damage categories (Minor Damage, Major Damage, Destroyed) are low with F1 scores of 0.2107, 0.1538, and 0.041, respectively. This finding means that, although the model is quite accurate in its ability to detect undamaged structures, it has problems detecting varying degrees of damage.

2. **Pre-trained Model + Augmentation:** Pre-trained Model + Augmentation: Adding augmentation effects to the pre-trained model slightly boosts its efficiency in damage ranges. The Localization accuracy increases a little to 0.8149 and the F1 scores for Minor Damage and Major Damage increase to 0.2348 and 0.1726 respectively. The Destroyed class also improves from 0.041 to 0.052. Yet No-Damage class representation deteriorates a bit, likely because the model becomes more tuned to damage classes, suggesting that augmentation could help maintain class representation balance by introducing more diverse examples to the model.

3. **Pre-trained Model + Fine-Tuning:** Fine-tuning the pre-trained model yields significant improvements for every class, including damage classes. The localization score

increases to 0.8419 and the No-Damage score increases a bit to 0.6724. This gives huge boosts to Minor Damage and Major Damage, with F1 values of 0.2923 and 0.1837 respectively. Perhaps most importantly, the Destroyed class increases substantially to 0.0954, suggesting that fine-tuning allows the model to extract finer details from damaged structures by making it exactly fit the characteristics of the target domain.

4. **Pre-trained Model + Fine-Tuning + Augmentation:** Fine-tuning combined with augmentation yields the best overall performance. Location accuracy reaches 0.8485, and the No-Damage F1 score increases to 0.6961 which indicates very good results for undetected structures. For damage classes Minor Damage increases to 0.3147, Major Damage to 0.1920, and Destroyed to 0.117. This setup works especially well since augmentation probably gives you more diversity of data, and fine-tuning brings the model into alignment with target features, yielding healthy improvements across all categories and substantially improving the detection of severe damage.

5. **Pre-trained Model + CORAL DA:** Combining CORAL domain adaptation (DA) with the pre-trained model produces inconsistent results. Localization accuracy lags slightly behind fine-tuning at 0.8146, and the No-Damage class score sinks to 0.6581. Yet, CORAL DA does better in the damage categories (Minor Damage 0.2753 and Destroyed 0.059). These findings suggest that CORAL matches domain features well, at least for damage classes, but is not as useful in improving localization and undamaged classes as fine-tuning.

6. **Pre-trained Model + CORAL DA + Augmentation** The combination of CORAL DA with augmentation leads to better performance than CORAL alone. Localization fidelity goes up to 0.8215 and the No-Damage score increases to 0.6641. Minor-damage score improves to 0.3016 and destroyed increases to 0.0631. This pairing is not as efficient as fine-tuning with augmentation, but its still good at picking up classes of damage (minimal damage, for example). This blend seems to benefit from both domain alignment from CORAL and diversity provided by augmentation, but it is not nearly

as accurate as fine-tuning.

5.6.1 Analysis of Top-3 Winning Solution Models And Understanding Bias

When one compares the output of each location and the model performance scores in Section 5.1 and Tables 5.2 - 5.4, one can observe a clear correlation between geographic regions and model performance. Places that perform well on the pre-trained model are primarily in the United States (Arkansas, Alabama, California, Florida, Missouri, Oklahoma, South Carolina, Texas), indicating that the model would be better adapted to local environmental and climate conditions. These regions must have similar landscape, building, and disaster type characteristics that the model has been trained on to make them more accurate.

By contrast, regions of medium performance are located in Mediterranean and Tropical climates in the Equatorial Region (Hawaii, Mexico, Portugal, South Australia, Indonesia, Guatemala, Sunda, Ayiti). Such climates can have environmental effects (e.g. different vegetation composition, relative humidity, or light) that slightly deviate from the areas where the model was trained and achieve medium accuracy. In such climates, variation in architecture and infrastructure can also be the source of the models poor performance.

The worst performing spot is a quieter part of the Indian subcontinent (Nepal), where the model performs poorly despite having decent samples to train. It must be a very different area in terms of geographic and structural properties than what was in the model training dataset. Those differences cause bad performance and indicate that the model cannot apply well to these environments.

These results are contrast to what previous papers have observed. Other solutions state that the model performs better in locations that have higher number of training samples [1]. But here we observed a different pattern. For example, even though the Nepal/Utter Pradesh location has highest number of training samples in the whole dataset, during testing this location performed poorly which is not expected. These results point to a previously unknown geographic bias in the model, it appears to be more sensitive to some climate regions and less responsive to others. This bias may be due to lack of representative geographic areas

in the training data, which biases performance with respect to the climate and environment of the test area.

To analyze the bias of model in more detail. We have considered testing the model on unseen locations from the same geographical regions as described above. We've considered Texas from United States, Portugal and Ayiti locations from Mediterranean and Tropical regions. In all the previous experiments during training we've used these locations as training data, but we cannot use the same model to test because these data has already been seen by the model.

We trained the Top-1 winning solution [3] from scratch by taking out the samples of these locations from train data and tested on Texas, Portugal and Ayiti which are now unseen locations and Out-of-domain data to the model. We also implemented the fusion augmentation to check its effectiveness in reducing the bias and used domain adaptation methods to improve the scores of model in unseen locations.

In the table the below the pretrained model refers to top-1 winning solution trained xBD but excluding Texas, Ayiti and Portugal.

Table 5.8: Results of Domain Adaptation on Texas

Methods	**Localiza-tion**	**No-Damage**	**Minor-Damage**	**Major-Damage**	**Destroyed**
Pretrained model[3]	0.8236	0.6673	0.4105	0.5528	0.2141
Pretrained model + Augmentation	0.8285	0.6631	0.4344	0.5721	0.2154
Pretrained model + Fine-Tuning	0.8429	0.6742	0.4723	0.6037	0.2931
Pretrained model + Fine-Tuning+Augmentation	0.8485	0.6961	0.5047	0.6320	0.3117
Pretrained model + CORAL DA	0.8246	0.6581	0.4553	0.4446	0.2259
Pretrained model + CORAL DA + Augmentation	0.8315	0.6641	0.4616	0.4494	0.2631

Table 5.9: Results of Domain Adaptation on Ayiti

Methods	**Localization**	**No-Damage**	**Minor-Damage**	**Major-Damage**	**Destroyed**
Pretrained model[3]	0.6852	0.6214	0.3108	0.1892	0.2046
Pretrained model + Augmentation	0.7043	0.6638	0.3916	0.2631	0.2354
Pretrained model + Fine-Tuning	0.8124	0.6946	0.4315	0.2981	0.2852
Pretrained model + Fine-Tuning + Augmentation	0.8413	0.6961	0.4958	0.3419	0.3140
Pretrained model + CORAL DA	0.7246	0.6417	0.4054	0.2531	0.2419
Pretrained model + CORAL DA + Augmentation	0.7515	0.6511	0.4212	0.3058	0.2913

Table 5.10: Results of Domain Adaptation on Portugal

Methods	**Localization**	**No-Damage**	**Minor-Damage**	**Major-Damage**	**Destroyed**
Pretrained model[3]	0.7049	0.6415	0.3049	0.1992	0.3985
Pretrained model + Augmentation	0.7251	0.6832	0.3257	0.2118	0.4157
Pretrained model + Fine-Tuning	0.8028	0.7212	0.4082	0.3046	0.4453
Pretrained model + Fine-Tuning+Augmentation	0.8285	0.7516	0.4267	0.3685	0.4875
Pretrained model + CORAL DA	0.7564	0.6762	0.3481	0.2478	0.4295
Pretrained model + CORAL DA + Augmentation	0.7917	0.6839	0.3795	0.3495	0.4495

These results help us to analyze how the model is biased. As you can observe the results of Texas, even though it has less number of training samples than Portugal, the model generalized better on Texas rather than Portugal and Ayiti. This proves that the model doesnt't have only representational bias, but also geographic and structural bias.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This thesis explores the generalization abilities and biases of deep learning models for building damage detection from satellite images, addressing important gaps in model robustness for a variety of terrains and environments. One thing that we noticed is that the existing models, even the best-performing models from the xView2 competition, have a bias against real-world disasters and generalization problem in Minor and Major damage classes. To overcome these drawbacks, we introduce Fusion Augmentation, a new method that enriches diversity of data by improving edge detection and structure. This is a powerful method for better generalization of models, especially when it comes to detecting Minor Damage and Major Damage.

By using both in-domain (xBD) and out-of-domain (Ida-BD) data, the research discusses domain adaptation strategies, both supervised and unsupervised approaches. For instance, CORAL domain adaptation in conjunction with augmentation can map feature distributions across domains with excellent results even when labeled target data is not available. These techniques lead to increased damage detection performance, especially during unseen disasters, and are therefore highly useful for real-time disaster planning.

In addition, this study highlights the significance of explainable AI tools like Grad-CAM to make models interpretable and provide visualization of model predictions which can be useful to decision-makers in disaster planning. These visualizations facilitate transparency and trust by ensuring predictions are based on meaningful regions in images instead of spurious correlations or data biases.

Despite these advancements, the study acknowledges the need for further research to address challenges such as scaling models to larger datasets and improving performance on low represented data. Future work could explore integrating additional data augmentation strategies to improve generalization towards unseen structures and unseen geographical regions.

In summary, Fusion Augmentation combined with domain adaptation and explainable AI can be used effectively to improve the robustness, precision, and reliability of the models for damage detection. These findings open the door to further work to develop flexible, neutral models that can provide quick and accurate damage assessments that can help in disaster response and recovery planning.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. Gupta, B. Goodman, N. Patel, R. Hosfelt, S. Sajeev, E. Heim, J. Doshi, K. Lucas, H. Choset, and M. Gaston, "Creating xbd: A dataset for assessing building damage from satellite imagery," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 10–17, 2019.

[2] xView2 Access Building Damage, "Computer vision for building damage assessment." https://xview2.org/. Accessed: Nov 26, 2024.

[3] V. Durnov, "xview2 first place." https://github.com/DIUx-xView/xView2 first place/tree/master, 2020. Accessed: 2024-09-10.

[4] S. Seferbekov, "xview2 second place." https://github.com/DIUx-xView/xView2 second place/tree/master, 2020. Accessed: 2024-09-10.

[5] E. Khvedchenya, "xview2 third place." https://github.com/DIUx-xView/xView2 third place/tree/master, 2020. Accessed: 2024-09-10.

[6] N. Sharma, "Machine learning vs deep learning vs artificial intelligence (ml vs dl vs ai)." https://www.analyticsvidhya.com/articles/machine-learning-vs-artificial-intelligence-vs-deep-learning/. Accessed: Nov 26, 2024.

[7] J. Joseph, "A gentle introduction to neural networks." https://clevertap.com/blog/neural-networks/. Accessed: Nov 26, 2024.

[8] S. Saha, "A comprehensive guide to convolutional neural networks." https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Accessed: Nov 26, 2024.

[9] S. Jadon, "Introduction to different activation functions for deep learning." https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092. Accessed: Nov 26, 2024.

[10] E. K. Jacob Murel, Ph.D., "What is ensemble learning." https://www.ibm.com/topics/ensemble-learningf01. Accessed: Nov 26, 2024.

[11] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Lladós, and U. Pal, "Signet: Convolutional siamese network for writer independent offline signature verification," *arXiv preprint arXiv:1707.02131*, 2017.

[12] K. Rohith, "Domain adaptation in computer vision: Everything you need to know." https://www.v7labs.com/blog/domain-adaptation-guide, 2022. Accessed: Nov 26, 2024.

[13] X. Hao, L. Liu, R. Yang, L. Yin, L. Zhang, and X. Li, "A review of data augmentation methods of remote sensing image target recognition," *Remote Sensing*, vol. 15, no. 3, pp. 827–867, 2023.

[14] N. Kaur, C.-C. Lee, A. Mostafavi, and A. Mahdavi-Amiri, "Large-scale building damage assessment using a novel hierarchical transformer architecture on satellite images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 15, pp. 2072–2091, 2023.

[15] M. Ghaffar, A. McKinstry, T. Maul, and T. Vu, "Data augmentation approaches for satellite image super-resolution," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, pp. 47–54, 2019.

[16] O. S. C. Vision, "Canny edge detection." https://shorturl.at/zG3g8. Accessed: Nov 26, 2024.

[17] Climate.NASA.GOV, "What are the effects of climate change on extreme weather." https://science.nasa.gov/climate-change/extreme-weather/. Accessed: Nov 26, 2024.

[18] U. N. O. for Disaster Risk Reduction, "Hazard definition and classification review: Technical report." https://www.undrr.org/publication/hazard-definition-and-classification-review-technical-report. Accessed: Nov 26, 2024.

[19] D. I. Unit, "Assessing building damage from satellite imagery." https://www.diu.mil/latest/assessing-building-damage-from-satellite-imagery. Accessed: Nov 26, 2024.

[20] J. Gyegyiri, "Automatic detection of building damage caused by hurricane on florida coastal area from aerial images," Master's thesis, Florida Atlantic University, 2024.

[21] S. Wiguna, B. Adriano, E. Mas, and S. Koshimura, "Evaluation of deep learning models for building damage mapping in emergency response settings," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 5651–5667, 2024.

[22] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. J. Inman, "A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications," *Mechanical systems and signal processing*, vol. 147, pp. 077–107, 2021.

[23] T. Ryan-Mosley, "How ai can actually be helpful in disaster response." https://www.technologyreview.com/2023/02/20/1068824/ai-actually-helpful-disaster-response-turkey-syria-earthquake/. Accessed: Nov 26, 2024.

[24] IBM, "Ai vs. machine learning vs. deep learning vs. neural networks: Whats the difference?." https://www.ibm.com/think/topics/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks. Accessed: Nov 26, 2024.

[25] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[26] J. Heaton, N. G. Polson, and J. H. Witte, "Deep learning in finance," *arXiv preprint arXiv:1602.06561*, 2016.

[27] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2015.

[28] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE systems journal*, vol. 13, no. 3, pp. 2213–2227, 2019.

[29] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[30] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 50, pp. 5998–6008, 2017.

[31] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[33] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[34] K. Cho, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[35] D. P. Kingma, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[36] P. Baheti, "The essential guide to neural network architectures." https://www.v7labs.com/blog/neural-network-architectures-guide, 2021. Accessed: Nov 26, 2024.

[37] P. Raval, "A gentle introduction to siamese neural networks architecture." https://www.projectpro.io/article/siamese-neural-networks/718, 2024. Accessed: Nov 26, 2024.

[38] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, pp. 1–30, Lille, 2015.

[39] JavaTpoint, "Performance metrics in machine learning." https://www.javatpoint.com/performance-metrics-in-machine-learning. Accessed: Nov 26, 2024.

[40] Q. Xu, Y. Shi, X. Yuan, and X. X. Zhu, "Universal domain adaptation for remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023.

[41] J. Jakubik, M. Muszynski, M. Vössing, N. Kühl, and T. Brunschwiler, "Toward foundation models for earth monitoring: Generalizable deep learning models for natural hazard segmentation," in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*, pp. 5638–5641, IEEE, 2023.

[42] Y. Li, C. Lin, H. Li, W. Hu, H. Dong, and Y. Liu, "Unsupervised domain adaptation with self-attention for post-disaster building damage detection," *Neurocomputing*, vol. 415, pp. 27–39, 2020.

[43] D. Duarte, F. Nex, N. Kerle, and G. Vosselman, "Satellite image classification of building damages using airborne and satellite image samples in a deep learning approach," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, pp. 89–96, 2018.

[44] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[45] C. Wu, F. Zhang, J. Xia, Y. Xu, G. Li, J. Xie, Z. Du, and R. Liu, "Building damage detection using u-net with attention mechanism from pre-and post-disaster remote sensing datasets," *Remote Sensing*, vol. 13, no. 5, pp. 905–927, 2021.

[46] L. Dong and J. Shan, "A comprehensive review of earthquake-induced building damage detection with remote sensing techniques," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 84, pp. 85–99, 2013.

[47] D. Kim, J. Won, E. Lee, K. R. Park, J. Kim, S. Park, H. Yang, and M. Cha, "Disaster assessment using computer vision and satellite imagery: Applications in detecting water-related building damages," *Frontiers in Environmental Science*, vol. 10, 2022.

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

[49] L. Wang, J. Wu, Y. Yang, R. Tang, and R. Ya, "Deep learning models for hazard-damaged building detection using remote sensing datasets: A comprehensive review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[50] F. Yamazaki, "Applications of remote sensing and gis for damage assessment," *Structural Safety and Reliability*, vol. 1, pp. 1–12, 2001.

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

[52] D. Mercier, S. T. R. Rizvi, V. Rajashekar, S. Ahmed, and A. Dengel, "Utilizing out-domain datasets to enhance multi-task citation analysis," in *International Conference on Agents and Artificial Intelligence*, pp. 113–134, Springer, 2021.

[53] F. J. Tsai and S.-Y. Lin, "A class distance penalty deep learning method for post-disaster building damage assessment," *KSCE Journal of Civil Engineering*, pp. 1–15, 2024.

[54] S. T. Seydi, M. Hasanlou, J. Chanussot, and P. Ghamisi, "Bdd-net+: A building damage detection framework based on modified coat-net," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 4232–4247, 2023.

[55] S. Kaur, S. Gupta, S. Singh, V. T. Hoang, S. Almakdi, T. Alelyani, and A. Shaikh, "Transfer learning-based automatic hurricane damage detection using satellite images," *Electronics*, vol. 11, no. 9, pp. 1448–1463, 2022.

[56] Y. Trochun, Z. Wang, O. Rokovyi, G. Peng, O. Alienin, G. Lai, Y. Gordienko, and S. Stirenko, "Hurricane damage detection by classic and hybrid classic-quantum neural networks," in *2021 International Conference on Space-Air-Ground Computing (SAGC)*, pp. 152–156, 2021.

[57] Z. Hong, H. Zhong, H. Pan, J. Liu, R. Zhou, Y. Zhang, Y. Han, J. Wang, S. Yang, and C. Zhong, "Classification of building damage using a novel convolutional neural network based on post-disaster aerial images," *Sensors*, vol. 22, no. 15, pp. 5920–5934, 2022.

[58] J. Xia, B. Adriano, G. Baier, and N. Yokoya, "Building damage mapping via transfer learning," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 4841–4844, 2019.

[59] M. Dinh, V. L. Bui, D. C.Bui, D. P. Long, N. D. Vo, and K. Nguyen, "Performance evaluation of optimizers for deformable-detr in natural disaster damage assessment," in *2022 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pp. 1–6, 2022.

[60] F. Safavi, T. Chowdhury, and M. Rahnemoonfar, "Comparative study between real-time and non-real-time segmentation models on flooding events," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 4199–4207, 2021.

[61] D. Melamed, C. Johnson, I. D. Gerg, C. Zhao, R. Blue, A. Hoogs, B. Clipp, and P. Morrone, "Uncovering bias in building damage assessment from satellite imagery," in *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*, pp. 8095–8099, IEEE, 2024.

[62] C. M. Gevaert, T. Buunk, and M. J. van den Homberg, "Auditing geospatial datasets for biases: Using global building datasets for disaster risk management," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 12579–12590, 2024.

[63] O. Adedeji, P. Owoade, O. Ajayi, and O. Arowolo, "Image augmentation for satellite images," *arXiv preprint arXiv:2207.14580*, 2022.

[64] R. Takahashi, T. Matsubara, and K. Uehara, "Data augmentation using random image cropping and patching for deep cnns," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 2917–2931, 2019.

[65] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *Information*, vol. 11, no. 2, pp. 125–145, 2020.

[66] T. DeVries, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[67] Y. Hu and H. Tang, "On the generalization ability of a global model for rapid building mapping from heterogeneous satellite images of multiple natural disaster scenarios," *Remote Sensing*, vol. 13, no. 5, pp. 984–1008, 2021.

[68] Y. Ma, S. Chen, S. Ermon, and D. B. Lobell, "Transfer learning in environmental remote sensing," *Remote Sensing of Environment*, vol. 301, 2024.

[69] A. Somani, A. Horsch, and D. K. Prasad, *Interpretability in deep learning*, vol. 2. Springer, 2023.

[70] A. Fujita, K. Sakurada, T. Imaizumi, R. Ito, S. Hikosaka, and R. Nakamura, "Damage detection from aerial images via convolutional neural networks," in *2017 Fifteenth IAPR international conference on machine vision applications (MVA)*, pp. 5–8, IEEE, 2017.

[71] S. A. Chen, A. Escay, C. Haberland, T. Schneider, V. Staneva, and Y. Choe, "Benchmark dataset for automatic damaged building detection from post-hurricane remotely sensed imagery," *arXiv preprint arXiv:1812.05581*, 2018.

[72] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450, Springer, 2016.

[73] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[74] S. Uddin and H. Lu, "Dataset meta-level and statistical features affect machine learning performance," *Scientific Reports*, vol. 14, no. 1, pp. 1670–1682, 2024.

[75] S. Kshirsagar and T. Falk, "Cross-language speech emotion recognition using bag-of-word representations, domain adaptation, and data augmentation," *Sensors*.

[76] S. R. Kshirsagar, *Affective Human-Machine Interfaces: Towards Multi-lingual, Environment-Robust Emotion Detection from Speech*. PhD thesis, Institut National de la Recherche Scientifique (Canada), 2022.

[77] S. R. Kshirsagar and T. H. Falk, "Quality-aware bag of modulation spectrum features for robust speech emotion recognition," *IEEE Transactions on Affective Computing*, vol. 13, no. 4, pp. 1892–1905, 2022.

[78] S. Kshirsagar, A. Pendyala, and T. H. Falk, "Task-specific speech enhancement and data augmentation for improved multimodal emotion recognition under noisy conditions," *Frontiers in Computer Science*, vol. 5, p. 1039261, 2023.