TECHNICAL UNIVERSITY OF CRETE

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

DIPLOMA THESIS

Analysis of Remote Sensing Data using Artificial Intelligence Techniques in Order to Assess the Structural Stability of Buildings

---

Alkis Mavroudis

Examination Committee

(1) Professor Michail Zervakis (Supervisor)

(2) Professor Michail G. Lagoudakis

(3) Professor Georgios Stavroulakis (School of Production Engineering and Management)

Chania, 2025

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάλυση Δεδομένων Τηλεπισκόπησης με χρήση Τεχνικών Τεχνητής Νοημοσύνης για την Αξιολόγηση της Δομικής Ευστάθειας Κτιρίων

Άλκις Μαυρουδής

Εξεταστική Επιτροπή

(1) Καθηγητής Μιχαήλ Ζερβάκης (Επιβλέπων)

(2) Καθηγητής Μιχαήλ Γ. Λαγουδάκης

(3) Καθηγητής Γεώργιος Σταυρουλάκης (Σχολή Μηχανικών Παραγωγής και Διοίκησης)

Χανιά, 2025

Abstract

The preservation of buildings and infrastructure, increasingly vulnerable to damage from climate change-related natural disasters, remains a critical concern for both urban and rural areas, underscoring the need for advanced evaluation methods and mitigation strategies. Concurrently, advancements in Artificial Intelligence enable the development of tools that can automate significant aspects of structural stability assessment, enhancing efficiency and accuracy in response efforts. Regarding remote sensing data analysis in particular, Convolutional Neural Networks over the past decade, and more recently Vision Transformers, have shown promising results. However, existing publications often appear overfitted to address specific use cases, focusing on higher performance metrics, rather than on effectiveness for in-field applications. This diploma thesis proposes a unified pipeline for building damage assessment based on the two primary sources of remote sensing data, satellite and aerial imagery, leveraging contemporary methods incorporating a Siamese U-Net approach for satellite images and a custom-trained YOLO model for drone footage. The core purpose of this work is to provide a toolkit for obtaining an assessment overview using satellite imagery and enabling further investigation of areas of interest through the deployment of unmanned aerial vehicles. Experimental validations demonstrate the superior output in accuracy and inference speed of the proposed compared to the baseline models, while extended testing in real-world disaster scenarios in Greece and internationally highlights the generalizability of the process across a wide range of cases. The findings showcase the potential for real-time, deployable solutions in resource-constrained environments, bridging the gap between research and practical implementations. Ultimately, as automated disaster assessment continues to improve, the aggregation of analyzed data from previous events will become an invaluable resource for responding to crises moving forward.

Περίληψη

Η συντήρηση υποδομών και κτιρίων, τα οποία είναι ολοένα και πιο ευάλωτα σε φυσικές καταστροφές λόγω της κλιματικής αλλαγής, παραμένει ένα φλέγον ζήτημα για αστικές και αγροτικές περιοχές, επισημαίνοντας την ανάγκη για εξελιγμένες μεθόδους αξιολόγησης και στρατηγικές μετριασμού ζημιών. Συγχρόνως, η πρόσφατη πρόοδος στον τομέα της τεχνητής νοημοσύνης παρέχει τη δυνατότητα ανάπτυξης εργαλείων, με σκοπό την αυτοματοποίηση σημαντικών πτυχών της αξιολόγησης δομικής ευστάθειας, βελτιώνοντας την αποδοτικότητα στις προσπάθειες διάσωσης. Όσον αφορά την ανάλυση δεδομένων τηλεπισκόπησης, προσεγγίσεις όπως τα Συνελικτικά Νευρωνικά Δίκτυα κατά την τελευταία δεκαετία και οι Vision Transformers πιο πρόσφατα, εμφανίζονται ιδιαίτερα υποσχόμενες. Εντούτοις, οι υπάρχουσες δημοσιεύσεις είναι συχνά υπερεστιασμένες σε συγκεκριμένες περιπτώσεις, δίνοντας μεγαλύτερη σημασία στη βελτιστοποίηση των στατιστικών των επιδόσεων, έναντι της αποτελεσματικότητας σε πρακτικές εφαρμογές. Η παρούσα διπλωματική εργασία προτείνει μία ενιαία λύση για την εκτίμηση της δομικής ευστάθειας κτιρίων, αξιοποιώντας τις δύο βασικές πηγές δεδομένων τηλεπισκόπησης, τις δορυφορικές και τις εναέριες εικόνες. Πιο συγκεκριμένα, χρησιμοποιεί σύγχρονες μεθόδους ενσωματώνοντας μία Siamese U-Net προσέγγιση για τις δορυφορικές εικόνες και ένα ειδικά εκπαιδευμένο YOLO μοντέλο για πλάνα από drone. Ο κύριος στόχος της εργασίας είναι η δημιουργία ενός εργαλείου, το οποίο παρουσιάζει μια γενική εικόνα εκτίμησης της εκάστοτε κατάστασης, με βάση τις δορυφορικές εικόνες και μετέπειτα εξετάζει περαιτέρω τις περιοχές που έχει ξεχωρίσει με την αποστολή εναέριων μέσων. Μέσω πειραματικής επαλήθευσης επιδεικνύεται η βελτιωμένη λειτουργία της προτεινόμενης λύσης συγκριτικά με προηγούμενες, τόσο σε ακρίβεια, όσο και σε χρόνο εκτέλεσης, ενώ βάσει εκτενών δοκιμών σε πραγματικές περιπτώσεις καταστροφών εντός και εκτός Ελλάδας επισημαίνεται η δυνατότητα γενίκευσης της διαδικασίας. Τα αποτελέσματα τονίζουν τη δυνατότητα ανάλυσης δεδομένων τηλεπισκόπησης σε πραγματικό χρόνο, γεφυρώνοντας το χάσμα μεταξύ έρευνας και πρακτικών εφαρμογών. Καταληκτικά, όσο η αυτοματοποιημένη αξιολόγηση φυσικών καταστροφών συνεχίζει να βελτιώνεται, η συγκέντρωση των αναλυμένων δεδομένων από προηγούμενα γεγονότα θα αποτελεί αναπόσπαστο κομμάτι για την αντιμετώπιση μελλοντικών κρίσεων.

Acknowledgements

First of all, I would like to thank my family, who supported me in every possible way throughout the years of my studies. Additionally, I have to thank Professor Georgios Stavrakakis and Professor Georgios Stavroulakis for their guidance during the entire period of writing my thesis, as well as Mr. Nikolaos Sxetakis and his team, for solving issues whenever they emerged. Special mention has to go to Professor Michail Zervakis and Professor Michail G. Lagoudakis, for contributing in completing this thesis.

Ευχαριστίες

Θα ήθελα πρωτίστως να ευχαριστήσω την οικογένειά μου, η οποία με στήριξε με κάθε δυνατό τρόπο κατά τα χρόνια των σπουδών μου. Παράλληλα, οφείλω να ευχαριστήσω θερμά τους καθηγητές κ. Γεώργιο Σταυρακάκη και κ. Γεώργιο Σταυρουλάκη για την καθοδήγησή τους καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας και τον κ. Νικόλαο Σχετάκη και την ομάδα του για τη συμβολή τους στην επίλυση ζητημάτων, όποτε αυτά προέκυπταν. Χρήζει ειδικής μνείας η συνδρομή των καθηγητών κ. Μιχαήλ Ζερβάκη και κ. Μιχαήλ Γ. Λαγουδάκη, οι οποίοι βοήθησαν στην ολοκλήρωση της παρούσας διπλωματικής εργασίας.

Table of Contents

Chapter 1 - Introduction

1.1 - Remote Sensing Overview

Remote Sensing

Remote Sensing is the field of obtaining data pertaining to certain objects or areas from a distance, without direct physical presence [1]. The two main ways for gaining access to said data is either through satellite, or aerial sensors, which in turn capture electromagnetic radiation, such as visible light, infrared and radar, to generate imagery, as well as, thermal emotions and gravitational measurements. Additionally, apart from specialized sensors, remote sensing also incorporates the process of capturing high-resolution images and video footage, mainly from Unmanned Aerial Vehicles (UAVs) or aircraft, for the purposes of environmental monitoring, disaster management, urban planning and agriculture, among many others.

Satellite Imagery

Satellite imagery refers to images of either Earth or other planets, captured by imaging satellites, operated by either governments or private businesses, around the world [1]. These resources have a wide range of applications, including but not limited to environmental monitoring, agriculture, forestry, urban planning, military intelligence and disaster response. For the latter in particular, a fundamental component of this dissertation, remote sensing data provides critical information that assists in all stages of disaster management, ranging from preparedness, all the way to recovery. Some defining traits of geospatial imagery are spatial resolution, which refers to the size of the smallest detectable object, spectral resolution, meaning the ability to capture data across various wavelengths of light, temporal resolution, alluding to the frequency of image updates over a certain area and radiometric resolution, considering the capability of detecting subtle differences in energy intensity. The imagery examined throughout this thesis is neither exclusively panchromatic, nor multispectral, since the highest resolution images, required for discerning minor details, are panchromatic, whereas color images useful for visualization are multispectral.

Panchromatic images are, by their nature, in grayscale, when visual interpretation often demands coloured ones. This can be alleviated with a process of combining panchromatic and multispectral images called pan-sharpening [2]. Pan-sharpening resamples the multispectral image, in order to match the spatial resolution of the panchromatic image, which is then combined with the resampled

multispectral bands. The end product, as illustrated by the provided example in Figure 1.1, is an image that is enhanced both in terms of spatial resolution, as well as color information. High-resolution satellite imagery, in general, refers to images with a Ground Sample Distance (GSD) of at least 1 meter, which means that each pixel in an image represents an area of 1 square meter.

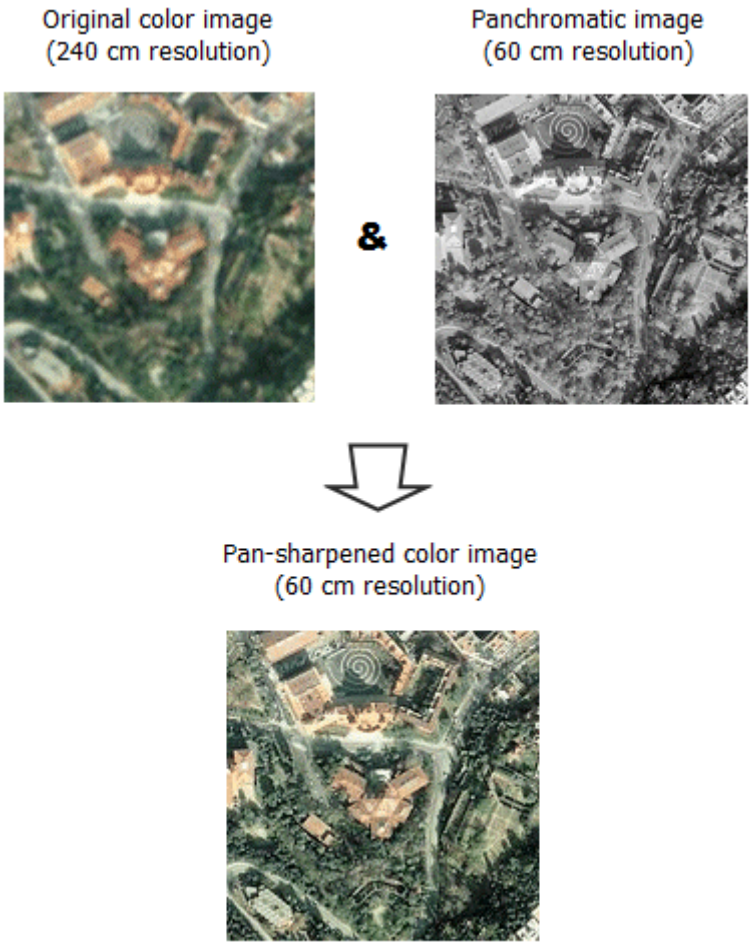


Figure 1.1. Visual example of image pan-sharpening [2].

GeoTIFFs

GeoTIFF is a public domain metadata standard, for embedding georeferencing information within a Tagged Image File Format (TIFF) [3]. It is the format most commonly used in the fields of remote sensing and Geographic Information Systems (GIS), for storing raster data. GeoTIFFs are particularly useful for satellite imagery analysis, since they combine high-resolution uncompressed image data, with georeferencing spatial metadata, that defines location, scale and projection of the raster data, allowing them to be accurately placed in geographic space. Additionally, since this format has been adopted as the industry norm, it is compatible with most, if not all, GIS software, while also being versatile in terms of the range of metadata it supports, for all sources of remote imagery. Overall,

GeoTIFFs can store multiple types of raster data, like aerial photography and digital elevation models (DEMs), on top of satellite imagery, finding use in applications such as land management and 3D modeling.

Aerial Imagery

Aerial imagery denotes either photographs, or video footage, which were taken from an airborne platform, whether that being drones, helicopters or airplanes [1]. This medium provides high-resolution images of both the Earth's surface and further potential perspectives, expanding on the ones available by satellites. Aerial footage can be utilized on its own, or complementary to satellite imagery, for similar reasons as stated above, such as agriculture and disaster response. Developments in the past decade, regarding professional, along with consumer drones, as well as high-resolution cameras, with advanced image processing techniques, renders aerial imagery an indispensable tool for gaining insights and making informed choices, based on accurate visual data.

1.2 - Artificial Intelligence Overview

Artificial Intelligence seems to monopolize the conversation around potential advancements in technology these days. To an extent, the publicity it receives seems justified, since the expected rise in productivity in a plethora of industries can't be understated. However, there appears to be some confusion between the terms of Artificial Intelligence, Machine Learning and Deep Learning, which are often used interchangeably, when in fact referring to different concepts [4].

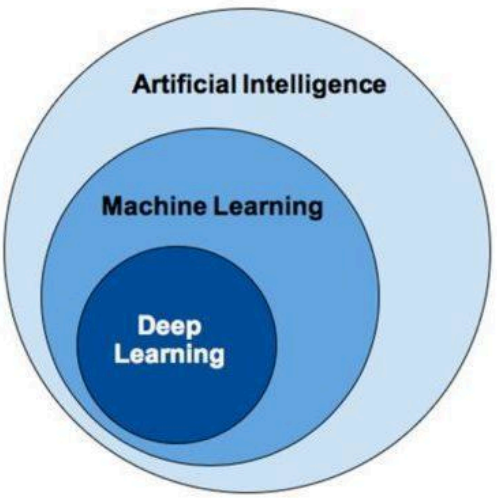


Figure 1.2. Visual representation of Artificial Intelligence and its subsets [5].

Artificial Intelligence

Artificial Intelligence (AI) is the umbrella term used for all types of technology possessing attributes resembling some form of intelligence, akin to human behavior [4]. Examples of AI implementations include but are not limited to speech recognition, visual perception, language translation and overall decision making. The overarching purpose of AI is to aid people in bettering their quality of life, whether referring to optimizing their work flow or easing their day to day functioning.

Machine Learning

Machine Learning (ML) is a subset of AI, focusing on developing statistical algorithms, which are then trained, to acquire the ability to complete specific problems, without further human intervention [4]. ML systems are usually trained on large amounts of data, in which they eventually identify patterns and relationships between the various inputs. Training can be approached in a variety of ways, with either supervised, unsupervised or reinforcement learning. Some cases utilizing ML are recommendation systems, fraud detection and predictive analytics, among others.

Deep Learning

Deep Learning (DL) is a subset of machine learning, which specializes in utilizing artificial neural networks (ANNs) with multiple layers, to process vast amounts of data [4]. The term neural network is alluding to structure and functioning similar to that of human neurons, rendering them particularly useful for analyzing and processing various multi-dimensional data types, such as audio and images. Some of the most commonly used ANN implementations today are convolutional neural networks, the functioning of which is explained in further detail below. Core fields driven forward by advancements in DL include healthcare diagnostics, autonomous driving and even artistic output, with generative models producing music and visual imagery. Most notably, in natural language processing, with DL, machines are able to understand and respond in human language, powering popular chatbots (e.g. ChatGPT), along with translating and interpreting tools.

Convolutional Neural Networks

As referenced before, convolutional neural networks (CNNs) are some of the most widely used deep learning algorithms in the industry [6]. Due to their unique set of attributes they are considered perfect for processing structured grid data, like images. Contrary to older implementations of neural networks, CNNs use, as their name suggests, multiple convolutional layers in order to extract and

therefore detect features such as edges, textures and patterns. This is achieved with filters, which are also known as kernels, that slide over the input data, perform element-wise multiplications and return feature maps highlighting specific patterns.

CNN architecture is composed of feature extraction and classification layers, to enable CNNs to observe hierarchical representations of data, in a way that captures low-level features in the initial extracting layers and high-level, more abstract features in the deeper ones [7]. Feature extracting layers are split into convolutional layers, which extract local elements and pooling layers, which make computation more efficient by reducing the dimensionality of the data. Classification layers, in essence, are fully connected layers that integrate all the features that have been extracted and processed for the purpose of making a final prediction.

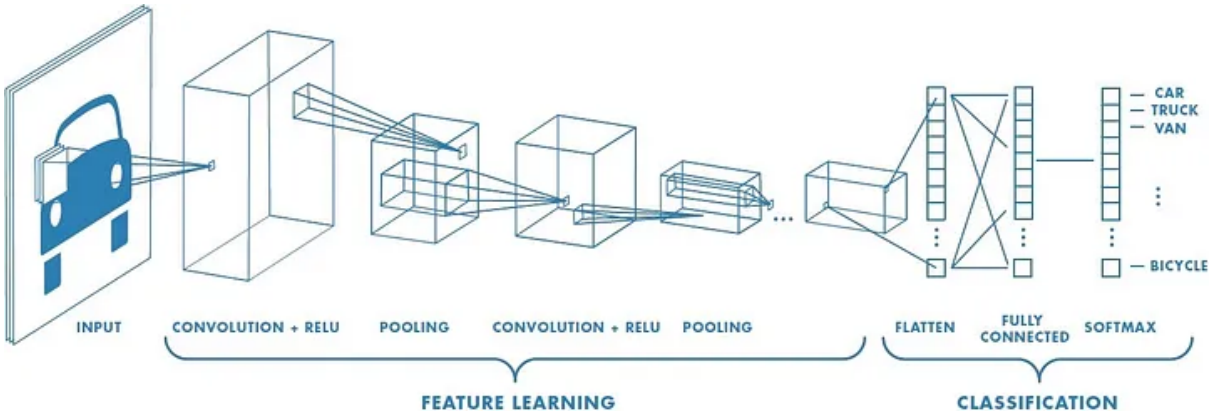


Figure 1.3. Convolutional neural network architecture overview [6]

As highlighted in the figure above, after specific features are extracted by the convolutional layer, non-linearity is introduced by the activation function, which in most cases is Rectified Linear Unit (ReLU). In some cases, other functions such as Sigmoid or Tanh are preferred, but ReLU stands as the norm since it is generally more efficient.

$$\mathrm{ReLU}(x) = \max(0, x)$$

ReLU replaces all negative values in the feature map with zero, while leaving positive values unchanged, for the purposes of speeding up training and preventing vanishing gradient problems. After expanding the network's capabilities past linear functions, making it in turn able to learn complex patterns, a pooling filter slides over the feature map, similarly to the previous convolution operation. This filter performs either Max Pooling, meaning taking the maximum value of a specific region of the feature map, like 2x2, or in rarer cases Average Pooling, which takes the corresponding

average, for reducing the spatial dimensions of the feature map. This ensures that the network is more robust, by making it more invariant to small translations in the input and it decreases the overall computational complexity.

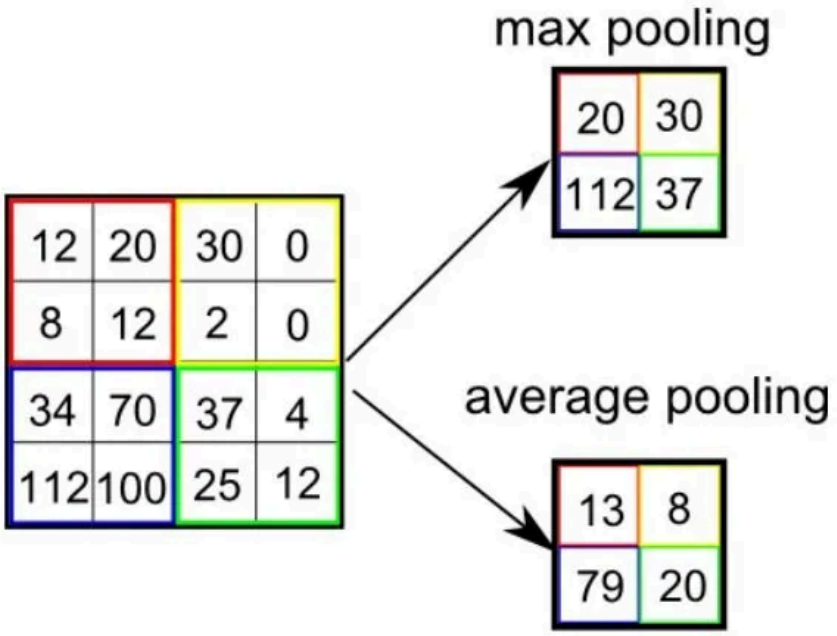


Figure 1.4. Max and average pooling examples [6].

For classification, a fully connected layer is deployed, which as the name suggests, connects every neuron in the previous layer to every neuron in the next, functioning like a traditional neural network, combining the learned features to predict the output. In essence, the high-level features are flattened into a 1D vector and then passed into fully connected layers for either classification or regression. In the final fully connected layer, a Softmax activation function is used, which converts raw output scores from the network into probabilities that sum to 1, allowing the model to make a probabilistic prediction over multiple classes. Finally, CNNs are trained using backpropagation where the network's weights are updated based on the error between the predicted output and the actual target.

Regarding the field of computer vision in particular, a key pillar of this study, CNNs have expanded on the potential ceiling of what can be achieved, for the likes of image classification, object detection and image segmentation [6]. In other spheres, outside of the ones covered in the research conducted for this thesis, CNNs are deployed for analyzing video footage and natural language processing. There are also notable implementations for detecting medical issues from images, that further illustrate AI's capabilities for humanitarian assistance. Overall, CNNs are powerful tools for any task requiring to go through large amounts of visual data, especially because they automatically and adaptively learn spatial hierarchies of features [7].

Siamese Neural Networks

Siamese Neural Networks (SNNs) are a unique implementation of neural network architecture, used specifically for identifying similarities between inputs [8]. This is achieved by two identical subnetworks that share both weights and parameters. SNNs are particularly effective for tasks where the end goal is determining the degree of similarity between two inputs and in some cases whether they are identical, such as facial recognition, signature verification and one-shot learning. Each subnetwork processes one of the two input data points, typically converting them into fixed-size feature vectors. These feature vectors are then compared using a distance metric, often Euclidean distance, to determine the similarity between the inputs. Training a Siamese Neural Network involves using pairs of inputs, labeled as similar or dissimilar. A common loss function used is the contrastive loss, which encourages the network to bring similar pairs closer in the feature space while pushing dissimilar pairs further apart, fine-tuning the network to accurately measure the similarity between input pairs. A visual representation of a siamese convolutional neural network implementation for writer independent offline signature verification, SigNet, is showcased in Figure 1.5, with two identical CNNs that are joined by a loss function at the top.

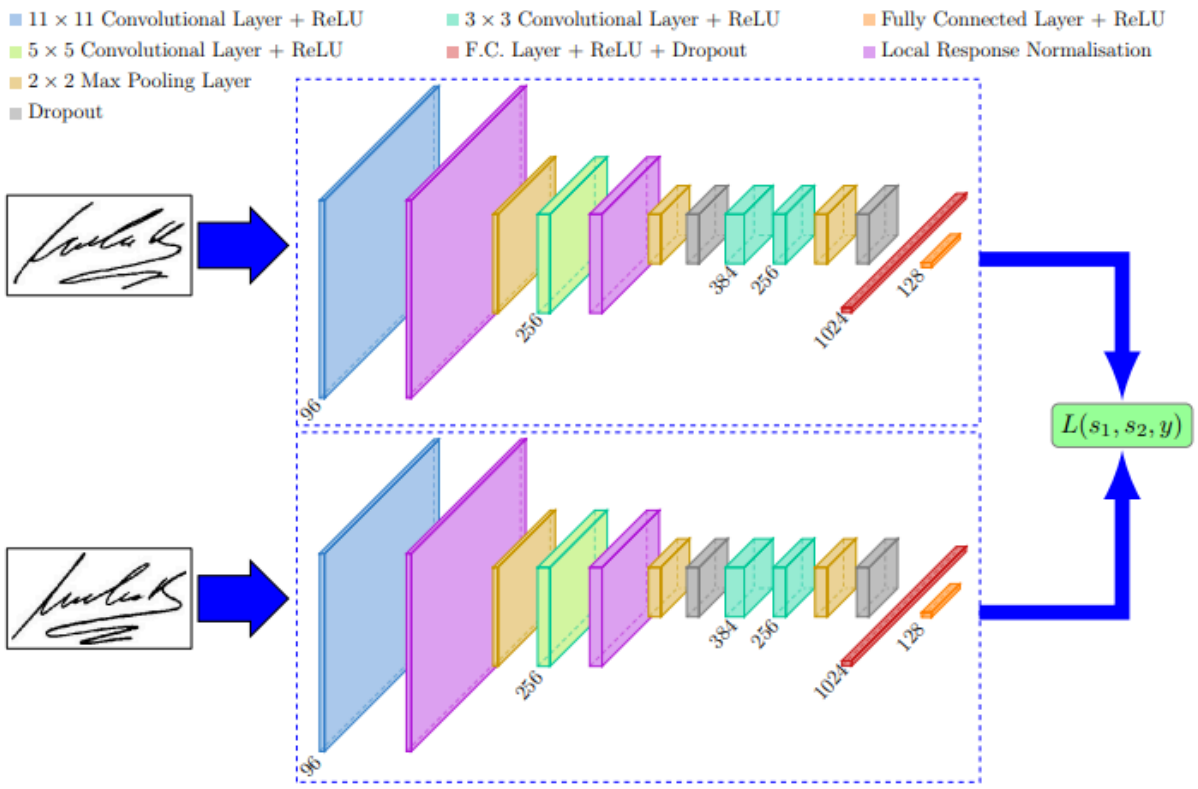


Figure 1.5. SigNet, convolutional siamese neural network implementation, architecture overview [8].

Overall, Siamese Neural Networks are one of the primary deep learning tools for tasks involving similarity assessment, while their ability to generalize from few examples makes them particularly valuable in applications where labeled data is limited [8].

1.3 - Methods for Image Normalization and Evaluation Metrics

Image Normalization

Image normalization achieves consistency within a varied dataset when it comes to how the model interprets each individual input [9]. Since images can vary greatly in terms of brightness, contrast, and overall pixel value ranges, normalizing images to have a mean of 0 and a standard deviation of 1 helps to bring all images into a consistent scale. This consistency is crucial for effective training and convergence of CNNs. Additionally, normalization helps in accelerating the convergence of the training process. When the data is normalized, the gradients during backpropagation are more stable, which can lead to faster and more stable convergence. Mean and Standard Deviation are defined by the following formulas.

$$\text{Mean} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{SD} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \text{Mean})^2}$$

Which in turn lead to the most common formula for normalization, expressed below.

$$x_{\text{norm}} = \frac{x - \mu}{\sigma}$$

By subtracting the mean pixel value from the image, the data is centered around zero, ensuring that the input features have zero mean, which is beneficial for the optimization algorithms used in training CNNs, such as gradient descent. Dividing by the standard deviation scales the pixel values to have unit variance. This makes sure that each feature contributes equally to the learning process, preventing features with larger variances from dominating the model [9].

Evaluation Metrics

A series of performance metrics commonly utilized for evaluating machine learning model performance, were also incorporated in the workflow of this dissertation. The calculation for the

majority of these metrics stems from measuring actual known classes, against total predicted classes, resulting in a confusion matrix like the one illustrated by Table 1.1 below.

		Actual Class	
		Positive (P)	Negative (N)
Predicted Class	Positive (P)	True Positive (TP)	False Positive (FP)
	Negative (N)	False Negative (FN)	True Negative (TN)

Table 1.1. Visual explanation of 2x2 confusion matrix [10].

For image segmentation tasks, the following four metrics are essential for deriving the robustness of the model at hand and are represented by the corresponding formulas [10].

➔ **Precision** measures the correctness of the positive predictions made by the model.

$$\text{Precision} = \frac{TP}{TP + FP}$$

➔ **Recall** measures the ability of a model to identify all relevant instances.

$$\text{Recall} = \frac{TP}{TP + FN}$$

➔ **Accuracy** measures the overall correctness of the model's predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

➔ **F1 Score** combines precision and recall into a single metric, which is useful when a balance is needed between the two, especially in cases of imbalanced datasets [11].

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

For image segmentation in particular, pixelwise F1 score refers to the comparison of the classification of each pixel, whereas for object detection, object-level F1 score refers to the comparison of predicted objects to ground-truth objects [11].

For object detection models, the three listed below are some additional fundamental evaluation metrics [12].

➔ **Confidence Score** represents the model's certainty about its predictions. It usually ranges from 0 to 1, where a score closer to 1 indicates higher confidence. While there isn't a single formula for the confidence score since it can depend on the model architecture, a common way to express it is the following, in which $P(y|x)$ is the probability that the model assigns the predicted class y given the input x.

$$\text{Confidence Score} = P(y|x)$$

➔ The **mean Average Precision (mAP)** averages the precision values at different levels of recall.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

Where, Average Precision (AP) is the area under the precision-recall curve for a specific class, summarizing the precision-recall curve into a single number.

➔ **Intersection over Union (IoU)** measures the overlap between the predicted bounding box and the ground truth bounding box. It is represented by the following formula:

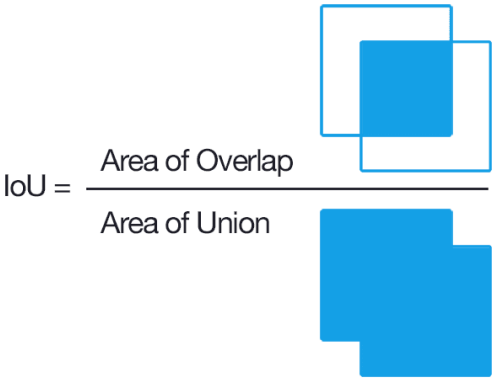


Figure 1.6. Intersection over Union formula and visual representation [13].

Where, Area of Overlap is the area where the predicted bounding box and the ground truth bounding box intersect, while Area of Union is the total area covered by both the predicted and ground truth bounding boxes, as seen in Figure 1.6. In previously defined terms from Table 1.1, IoU can be expressed as seen below.

$$\text{IoU} = \frac{TP}{TP + FP + FN}$$

1.4 - State of the Art and Thesis Framework

This thesis explores the emerging field of Building Damage Assessment (BDA) by analyzing remote sensing data using Artificial Intelligence techniques that have emerged in the last decade. It consolidates fragmented insights from existing literature and addresses the absence of robust tools through the development of a unified pipeline, offering new insights and advancing the foundational understanding of this domain. The pursuit of more effective methods for assessing the structural stability of buildings has consistently been a prominent area of research, both for the purpose of restoring cultural heritage structures, as well as responding to natural disasters more effectively. Historically, considerable human effort was required for these processes, but with recent advancements in hardware and software significant portions of these tasks can now potentially be automated [14]. In terms of remote sensing hardware, the imagery available has vastly improved over the turn of the millennium, with satellites being able to provide high-resolution images with a GSD of up to 0.1m [15]. Concurrently, both research-grade and consumer-grade UAVs have demonstrated advancements in stabilization, range capabilities and cost effectiveness rendering them an indispensable tool for collecting remote sensing data [16], [17]. As for software, progress in the discipline of AI and the prospective efficiencies in structural evaluation cannot be overstated. Sophisticated CNN implementations, coupled with more recently developed Vision Transformers (ViT) and State Space models have returned remarkable results as of late [18]. At the same time, new Large Language Models (LLMs) give capabilities to make the whole process more accessible to the end-user. The proposal by Tani et al. [19] demonstrates an implementation which narrows down from a large sum of field images, based on custom queries determined by each individual case. This signifies that through a potential synergy between image analysis methods and LLMs more relevant outputs can be derived in a swift and efficient manner. However, available applications utilizing LLMs appear to be limited at the time of writing this thesis.

In this dissertation, several available deep learning propositions and their applicability for BDA are examined, for both remote sensing major subdivisions, satellite and aerial imagery. The majority of building structural integrity instances analyzed through this work pertain to natural disasters, since there is an ever growing need for addressing said crises effectively [14]. By providing essential insights, the primary objective is to mitigate the impact on communities and preserve resources through timely intervention. At the same time, in the context of developing tools, particularly for model training, these scenarios present sufficiently large sample sizes composed of widely available data. The current landscape, which is presented in further detail in Chapter 2, is characterized by isolated research projects that tackle narrowly defined issues, resulting in a fragmented approach to the broader field. This can be attributed to the fact that the domain is still in its early stages, with only a few coordinated efforts conducted by major organizations [14]. While various publications highlight notable achievements regarding implemented techniques and evaluation metrics, a comprehensive assessment of the current state of the field, along with a unified approach to BDA, is lacking. These are the two primary issues that this thesis aims to address by reviewing the current state in Chapter 2 and utilizing existing publicly accessible resources, which are detailed in Chapter 3, to propose an integrated pipeline for BDA that encompasses all types of remote sensing data in Chapter 4. At the same time, while temporal efficiency is often overlooked in the development of deep learning models in favor of maximizing evaluation metrics, it is absolutely crucial for humanitarian response, and thus the aspect of expediting the process was given serious consideration. Several real-life scenarios are examined in Chapter 5, using raw data sourced online and processed to integrate into the proposed methodology. Ultimately, this work intends to provide a baseline assessment toolkit that offers scalability regarding the systems on which it can be deployed and generalizability concerning the real world applications for which it can yield substantial results.

Chapter 2 - Literature Review

2.1 - Machine Learning Implementations for Image Analysis

U-Net

For image segmentation specifically, one of the preferred CNN implementations is U-Net, which was first developed for biomedical imagery analysis, but in the following years has been adopted for a variety of needs concerning precise localization and feature classification within an image [20]. The network's defining trait is its symmetrical, U-shaped structure, which consists of two paths, a contracting and an expansive one, something that enables it to both extract features and perform detailed segmentation, thereby returning notable results for tasks requiring both context and spatial precision.

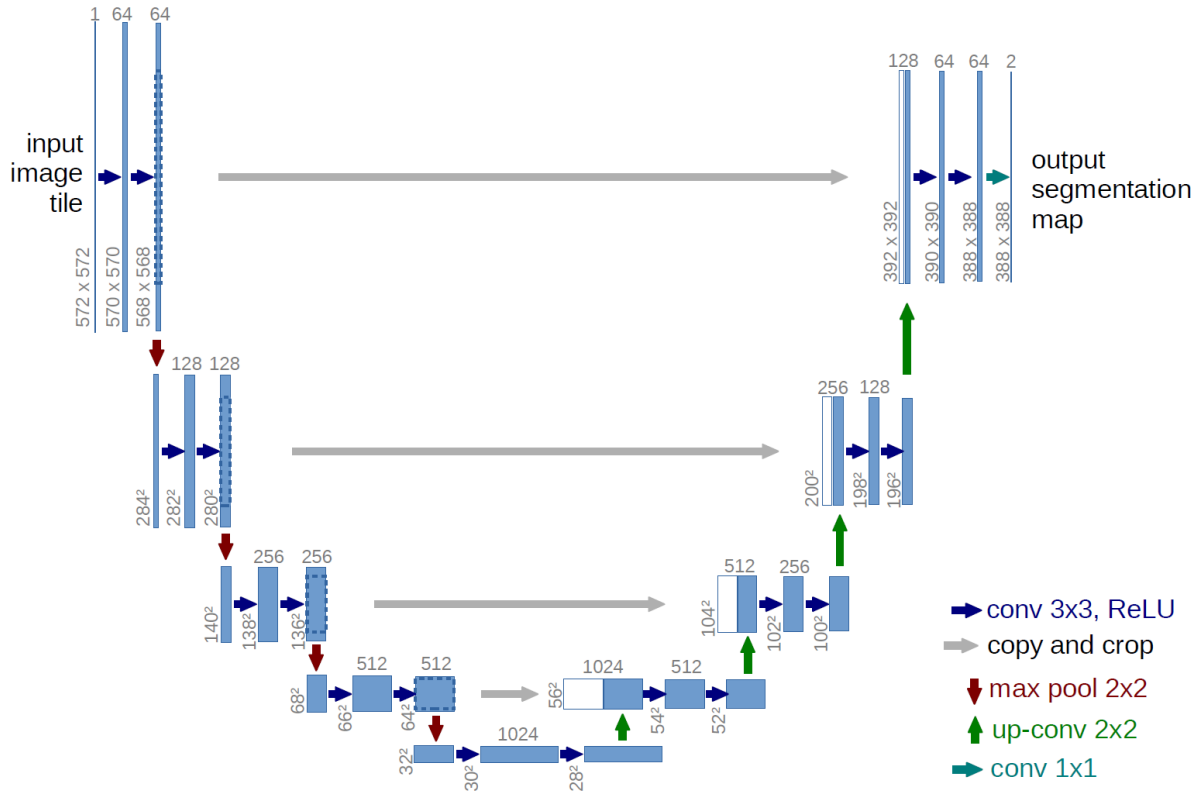


Figure 2.1. U-Net architecture overview, for 32x32 pixels in the lowest resolution [20].

The contracting path, which is also referred to as the encoder, is responsible for capturing the context of the input image. Similar to the typical CNN architecture presented in 1.2, a repeated application of convolutional layers, activation functions and pooling layers is deployed for sophisticated feature extraction. The connection between the contracting and expansive paths is facilitated by a bottleneck

layer, which consists of two convolutional layers with 3x3 filters, followed by ReLU activations, without any pooling. This ensures that high-level contextual information is bridged from the encoder to the decoder. The expansive path, or decoder, is responsible for precise localisation and reconstruction of the segmented image. With the utilization of progressive upsampling, the spatial resolution of the feature maps is increased, leading to maps featuring the original resolution, for pixel-level classification. To recover spatial information lost during the downsampling in the contracting path, U-Net uses skip connections, which pass feature maps from the contracting path to the decoder, enabling the network to retain both high-level and low-level information. At the end of the U-Net, a 1x1 convolution is applied to reduce the number of feature maps to the desired number of classes.

U-Net is widely adopted for image segmentation tasks, like medical imagery segmentation and autonomous driving, along with satellite and aerial image analysis. This can range from identifying tumor boundaries in MRI scans and detecting lanes and pedestrians to characterizing plant health for agricultural purposes, by segmenting the appropriate aerial footage [17], [21]. More recently, U-Net has found new applications in the field of image generation, for further image refinement and denoising [22].

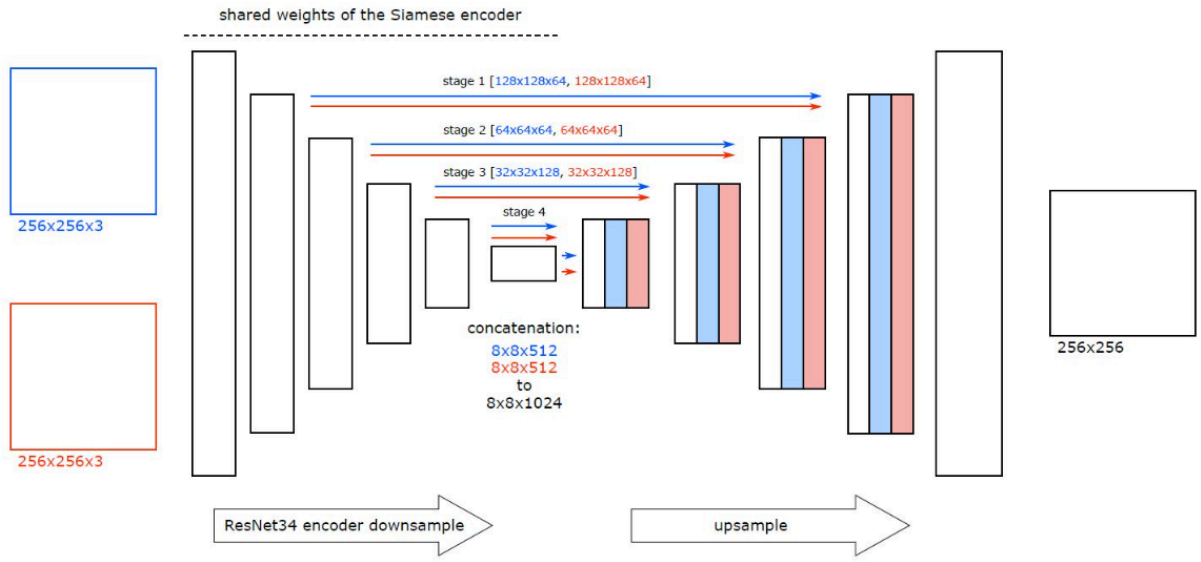


Figure 2.2. Siamese U-Net architecture overview, with ResNet34 model as encoder network [23].

A valuable subset of U-Net implementations are Siamese U-Net networks, pertaining to situations which involve comparing and analyzing similarity between two images, while at the same time preserving spatial information and segmenting relevant features within them [23]. This implementation, as the name suggests, combines two U-Net models in a Siamese architecture, as it

was defined in Section 1.2. In Figure 2.2, a Siamese U-Net example is showcased, which has two 256x256 RGB pictures as inputs and a ResNet34 model as an encoder network.

Some of the most common use cases for Siamese U-Net implementations are medical image analysis, image registration and change detection in satellite images, which is of particular interest for this thesis. For the first two, usefulness pertains to comparing different scans of the same patient over time to detect shrinkages and spatially aligning images that were taken during different sessions. For change detection especially, comparison of before and after satellite imagery can be used to track urban land developments, as well as environmental changes over a certain period of time [23].

YOLO

YOLO (You Only Look Once) is an object detection algorithm developed by Joseph Redmon and his team, that differentiates itself from the rest of the field by introducing a unified neural network model, for all cases of object detection [24]. Additionally, it provides an essential feature for the aims of this thesis, detecting and localizing objects from both image and video inputs in real time. YOLO utilizes a single CNN, that processes the entire image and returns a grid of predictions, which stems from a grid of cells the input is divided into. Every grid cell predicts several bounding boxes, along with coordinates and confidence scores. Then depending on the scores of each bounding box, the most accurate ones are displayed, according to a set confidence baseline.

The YOLO model's architecture consists of three main parts, the backbone, the neck and the head. The backbone extracts features using a convolutional neural network, which was originally based on simpler architectures such as Darknet, while newer versions deploy more sophisticated, deep CNNs such as Darknet-53 or CSPDarknet53. The neck combines multi-scale feature maps through techniques, like Feature Pyramid Networks (FPN) or Path Aggregation Networks (PAN), for enhancing the overall accuracy of the model. The head produces final predictions, including objectness scores, bounding boxes, and class probabilities, using anchor boxes during detections for predicting objects with varying aspect ratios. In Figure 2.3, a visual representation for the YOLOv8 model architecture in particular is showcased, featuring the aforementioned three main parts.

So far, ten versions of YOLO have been released, with each iteration bringing additional features, as well as improvements to the present ones. Since its launch, it has found a wide array of applications including, but not limited to, autonomous vehicles, surveillance systems, robotics and healthcare, use cases for all of which fast detection is crucial. For satellite imagery in particular, the proposal by Stavrakakis et al. [25], suggests a custom-trained YOLO model for real-time observation of marine

and aircraft traffic, along with container detection, highlighting the versatility of the model. Each version features several releases, with the latter ones providing both object detection and image segmentation, featuring various degrees of performance capabilities, as well as resource requirements. The latest release that is available for wider applicability is YOLOv9, which on all five models, that demand ranging levels of computational resources, yields better results than its predecessors, all while being more power efficient [26].

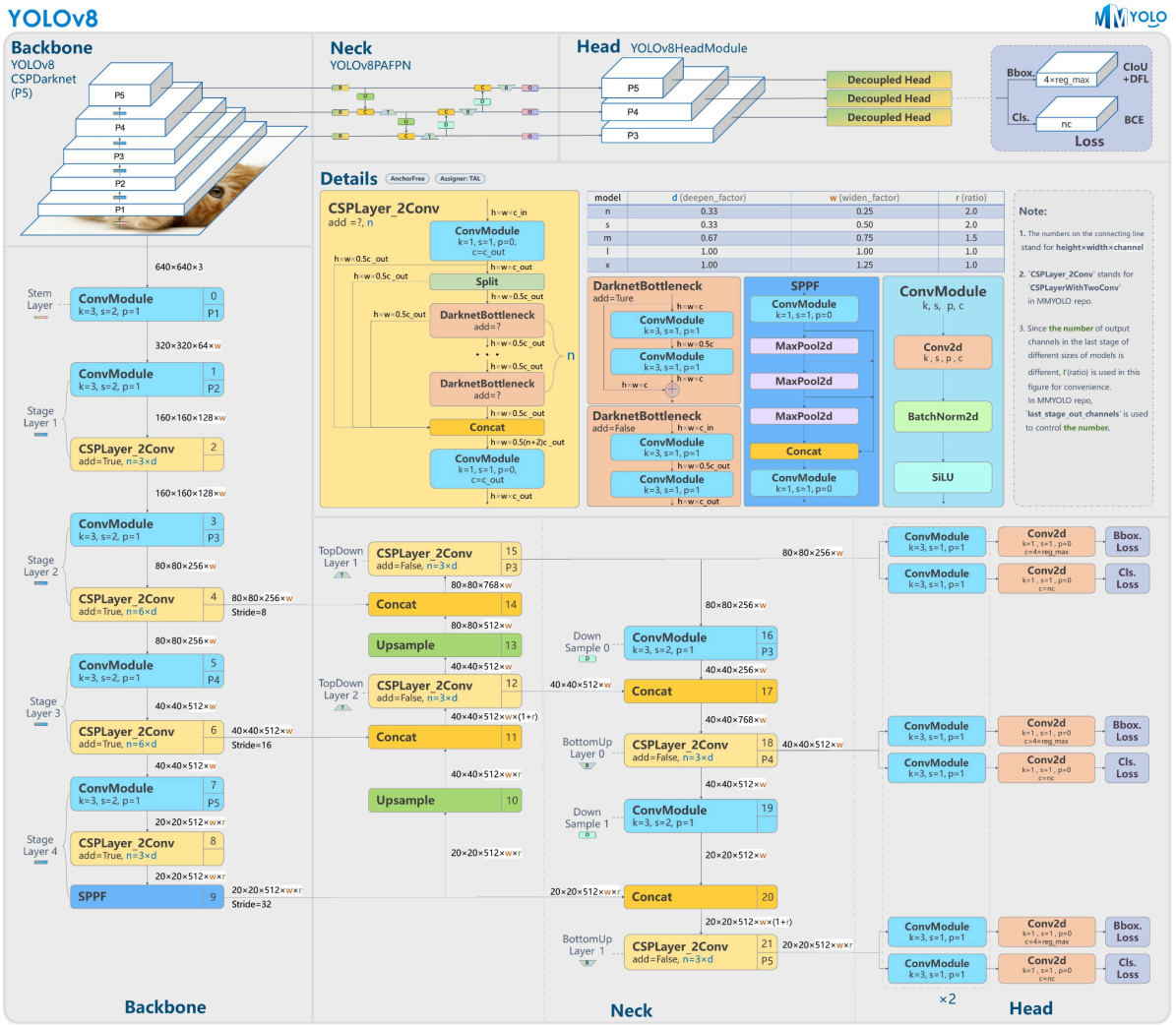


Figure 2.3. YOLOv8 model architecture overview [27].

2.2 - Related Works for Building Damage Assessment

In general, most available previous works focus on Change Detection (CD) for buildings, rather than Building Damage Assessment (BDA). Consequently, most available satellite imagery datasets cover urban areas that were developed in the last 10 to 15 years, so that high-resolution images can be sourced from both before and after instances. Datasets like LEVIR-CD and WHU-CD can assist

greatly in training models for building segmentation. In this field, approaches are split between binary change detection and semantic change detection, for which their primary differentiating trait is the amount of classes incorporated in the predicted output. Whereas binary CD produces a result that has labeled each pixel as either “changed” or “unchanged”, semantic CD has a multi-class map as output, where more information can be derived from each pixel, in terms of land usage changes for example. In this context, BDA can be defined as a focused subdivision of semantic CD, where the multi-class map corresponds to various levels of damage [18].

Available BDA model implementations are fewer and are in general bound within strict directives. Since even on satellite imagery of 0.3 meters Ground Sample Distance (GSD) minute details such as cracks cannot be discerned, building damage has to be evaluated on an arbitrary scale [14], [17]. This in turn means that results cannot be derived from buildings that are simply old or lack ample maintenance. For providing an appropriate sample size of damaged buildings for training and testing purposes, images have to be sourced from natural disaster events [14]. Datasets then either label buildings as “damaged” or “undamaged”, or follow the norm set by the creators of the xBD dataset, which is discussed in more detail in 3.1, where buildings are split based on a unified damage scale (e.g. “no damage”, “minor damage”, “major damage” and “destroyed”). Additionally, some datasets include environmental factors such as water from floods, or smoke from wildfires, elements that are also of substantial importance for response to natural disasters [17].

On the contrary, aerial drone footage provides the opportunity for more comprehensive analysis for an abundance of factors. First of all, more information can be derived from each frame, since UAVs have the ability to approach buildings much closer, especially in the case of a humanitarian crisis [16]. This encompasses particular details of buildings in high resolution, as well as angles that simply cannot be acquired from satellite imagery [17]. Still, available datasets are confined to arbitrary scales similarly to the ones referenced about satellite imagery above, as discerning particular features requires extensive capturing of each individual building’s facades. Pre-trained models follow suit, with most open-source models being provided by the creators of each dataset and therefore limited to use cases within each individual dataset.

2.3 - Overview of Available BDA Models using Satellite Imagery

Building damage assessment using satellite imagery before and after certain events can be broken into two primary tasks, building segmentation and damage classification [14]. Building segmentation in the vast majority of cases is performed independently on pre-event and post-event images, since no data correlation is required for carrying out this task successfully, or optimizing it in any way in

general. Damage classification on the other hand is performed by comparing features extracted from the two images, a task that can be performed with various methods.

The vast majority of existing BDA model implementations using satellite imagery utilize some form of CNN, for which the two main goals are accurate pixel-level image segmentation, along with effective comparison [18]. As noted in 2.1, one of the preferred CNN methods for image segmentation both in terms of overall accuracy, as well as resource requirements is U-Net, while Siamese architecture is deployed for comparing before and after image pairs. CNN implementations are primarily distinguished by whether or not they incorporate some form of attention module. Simpler approaches such as the baseline model by the creators of xBD, of U-Net + ResNet [14], or the Microsoft Model [28], both presented in 3.1, yield F1 scores around 0.7 to 0.8 for building detection and ranging from 0.1 to 0.6 for damage classification.

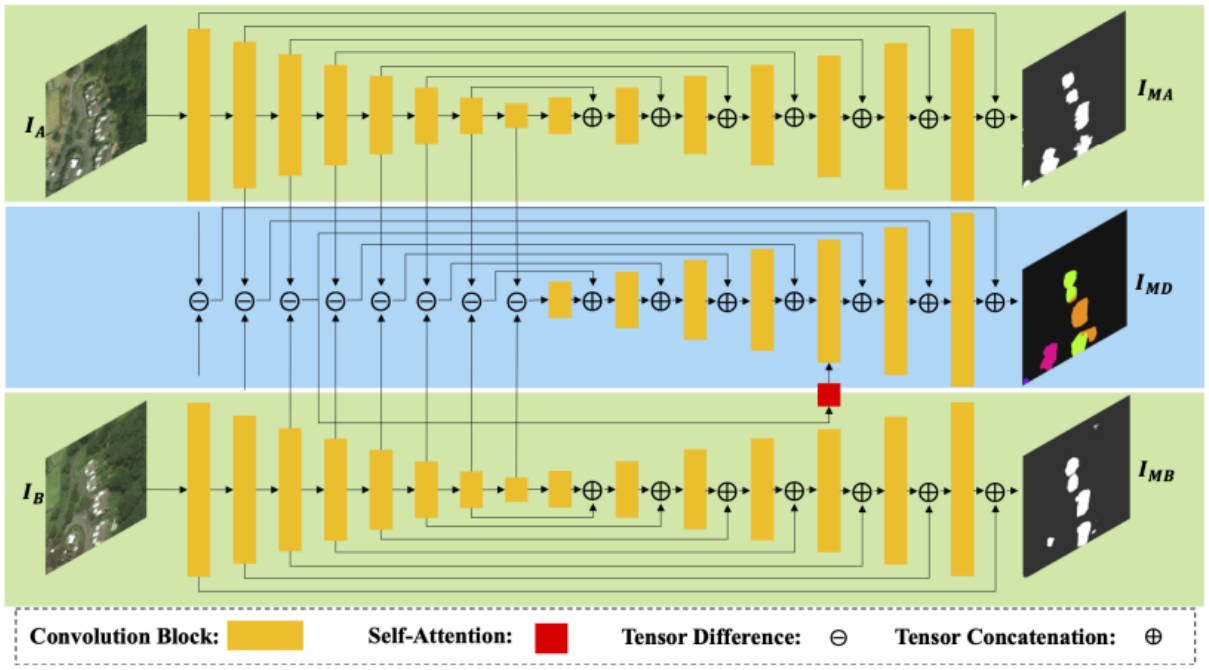


Figure 2.4. Architecture overview of Siamese U-Net Attention-based model for BDA, using satellite imagery [27].

Attention U-Net implementations such as the first place solution in the xView 2 challenge, or the one by Hao et al. [29], illustrated in Figure 2.4 return improved F1 scores, but with additional computational costs. The primary distinguishing factor is a self-attention module that takes into account contextual information, specifically the environmental factors surrounding the buildings in this case. The authors of [29] argue, for example, that buildings' roofs do not solely indicate structural damage and additional inputs such as water around each property. A similar approach by Zheng et al. [30] should be noted, focusing on building damage semantic detection through a task-aware

contextual encoder. This approach, in a comparable manner, provides a framework for gaining a broader understanding of each building's situation. The enhancements the self-attention modules provided, prompted researchers to branch into other architectures entirely for classification tasks.

A newer approach when it comes to computer vision and machine learning in general are Transformers, providing additional capabilities that are simply not feasible with CNN implementations [31]. While initially transformers were utilized for Natural Language Processing (NLP) applications, their ability to model long-range dependencies has prompted research into their applicability for computer vision tasks, leading to the development of Vision Transformers (ViTs). In building damage assessment, their capacity to focus on relevant features while filtering out irrelevant noise makes them particularly well-suited for large datasets, where context and global patterns are essential for accurate classification. While BDA transformer-based methods are minimally examined, not nearly to the same extent as corresponding BCD and SCD methods referenced above, the DamFormer framework stands out as the first BDA implementation utilizing a custom transformer module. As illustrated in Figure 2.5, it comprises two main parts, a Siamese Transformer encoder and a lightweight Dual-Tasks decoder. The authors claim, as evident in Table 2., that the DamFormer architecture outperformed previous CNN based implementations, in both building segmentation and damage classification tasks, showing promise for potential future Transformer based implementations.

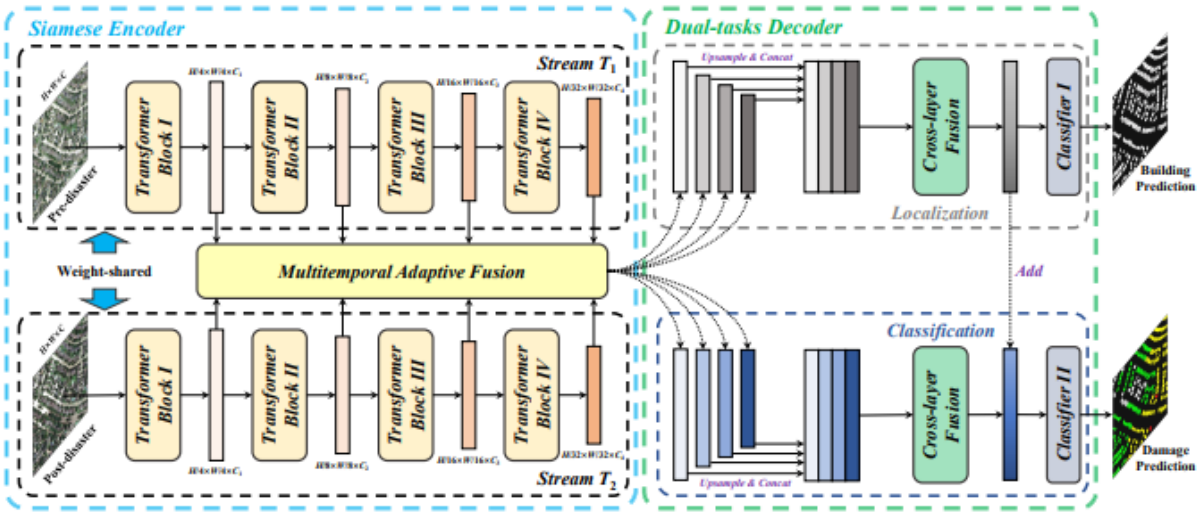


Figure 2.5. DamFormer architecture overview [31].

Recently, the use of State Space Model (SSM) methods has been proposed for BDA [18]. SSMs are predominantly used in dynamic systems requiring real-time monitoring, such as robotics, as well as in assessments involving multitemporal data. The main work implemented, in the field of interest, is ChangeMamba, which performs the binary and semantic CD referenced in 2.2, along with BDA. The authors of the paper claim that it nearly matches Transformer based methods for segmentation tasks, while exceeding them in the most important task that this work focuses on, which is classification. As

of writing this thesis it is the top performing paper for 2D Semantic Segmentation on xBD according to Papers With Code, with a weighted average F1 Score of 0.814 for the MambaBDA-Base model. However, the creators have not made the training weights available for their BDA models as of now, although they plan to release them in the future, which currently prevents the reproduction of their results. In the context of future building damage assessment implementations, SSMs can play a crucial role in monitoring structural integrity and predicting the progression of damage, allowing for proactive maintenance and intervention.

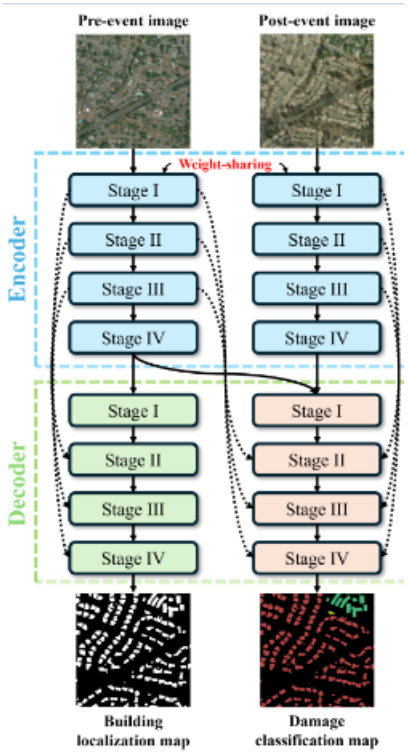


Figure 2.6. MambaBDA-Base network framework [18].

At the same time, datasets remain few with almost all existing implementations being trained, as well as being benchmarked, on the xBD dataset, given the particular challenges in sourcing data from relevant events and annotating it effectively [17]. While smaller datasets are available, they often suffer from inconsistent labeling and focus on individual disasters, making them suitable only for testing on xBD pre-trained models. All of the aforementioned proposals for BDA, therefore, provide performance metrics stemming from evaluation on the xBD dataset, which in turn means that they can be directly compared. A comparison of some of the most prominent models is showcased in Table 2.1, featuring individual F1 scores for building localization (F1 loc) and damage classification, both for each damage class and overall (F1 dam), along with overall F1 score (F1 overall), which prioritizes classification, by computing the weighted sum, according to the following formula [30].

$$F1\ overall\ =\ 0.3\ F1\ loc\ +\ 0.7\ F1\ dam$$

Method	F1 overall	F1 loc	F1 dam	Damage F1 per class			
				No	Minor	Major	Destroyed
xView2 Baseline	26.54	80.47	3.42	66.31	14.35	0.94	46.57
Siamese-UNet	71.68	85.92	65.58	86.74	50.02	64.43	71.68
Mask R-CNN	74.10	83.60	70.02	90.60	49.30	72.20	83.70
ChangeOS	75.50	85.69	71.14	89.11	53.11	72.44	80.79
DamFormer	77.02	86.86	72.81	89.86	56.78	72.56	80.51
MambaBDA-Base	81.41	87.38	78.84	95.94	62.74	76.46	88.58

Table 2.1. Comparison of performance metrics for available BDA models using satellite imagery, on the xBD dataset [14], [18], [30], [31].

In conclusion, Building Damage Assessment utilizing satellite imagery remains a complex endeavor. However, increasingly refined methods are continuously being developed. Each of these three methodologies, CNNs, vision transformers, and state space models, brings distinct advantages to BDA. CNNs are highly effective for extracting localized features from high-resolution images, making them ideal for identifying damage types. Vision transformers offer powerful capabilities for capturing global patterns and complex relationships, making them suitable for detecting subtle structural deformations. State space models excel at tracking damage over time, providing predictive insights that can inform long-term maintenance and prevention strategies [18]. Conversely, training resources are relatively limited, with the xBD dataset being the only one currently offering a sufficiently large sample size for extensive training, due to the challenges in data collection and annotation [14], while major computational demands for developing semantic detection methods in general seem to stall the progress of the field [18].

2.4 - Overview of Available BDA Models using Aerial Footage

BDA pre-trained models utilizing aerial footage are fairly limited, with most implementations developed by the creators of the datasets themselves. This in turn means that each application is fairly purpose-built, in order to accentuate the particular features of each dataset and not for creating a substantial trained model for BDA. Moreover, most implementations were found to use outdated dependencies, as well as being based on simple CNN methods, without exploring further advancements in ML/ DL presented in the past five years. This dissertation examines, among others, potential UAV BDA models leveraging developments in the field of AI, along with improvements in drone technology, both in terms of range capabilities and camera hardware.

Nevertheless, some existing implementations have to be highlighted for showing where the field is at currently, as well as where it can potentially move towards. While some demo pre-trained models on popular object detection models, such as the YOLO series seem to be available on platforms like Roboflow, there are not any corresponding publications for verifying their efficacy. Therefore, this thesis examines two approaches presented by the creators of the ISBDA dataset [16] and the RescueNet dataset [17], which are further showcased in 3.2. Both papers propose implementations for pixel-level image semantic segmentation, albeit with differing methods. The creators of ISBDA created their own model, called MSNet, with an architecture of four major components, as shown in Figure 2.7, performing both object detection and instance segmentation [16]. More precisely, MSNet consists of a Pyramid Backbone Network, a Hierarchical Region Proposal Network, a Score Refinement Network and a Mask R-CNN Head, which allude to a simplified YOLO implementation, even if it performs segmentation as well. The authors claimed it attained markedly improved results compared to state-of-the-art methods of the time, such as Mask R-CNN, however the process was deemed not replicable as the pipeline included deprecated dependencies.

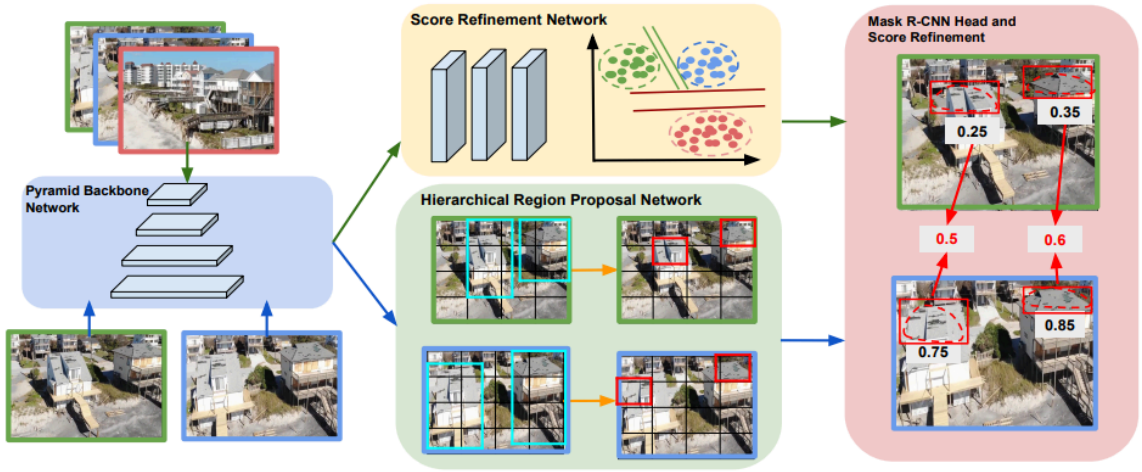


Figure 2.7. MSNet architecture overview [16].

The creators of RescueNet followed a different approach by testing the dataset on various existing semantic segmentation methods, with the aim of showcasing its applicability on future natural disaster damage assessment projects [17]. An Attention U-Net model proved superior to non-attention based methods, whilst Vision Transformer Segmenters returned substantial results as well. Collectively, the authors' results from the testing process are provided in Table 2.2, split according to building damage level, along with the mean IoU (mIoU) %.

Method	No Damage	Minor Damage	Major Damage	Destroyed	mIoU %
DeepLabv3+	61.6	49.6	47.3	57.2	57.43
PSPNet	95.16	94.36	96.91	98.95	95.67
Attention U-Net	99.46	99.59	99.56	99.89	98.47
Segmenter (ViT-Tiny)	65.18	44.41	53.16	89.54	68.49
Segmenter (ViT-Small)	80.74	72.42	79.67	95.85	85.78

Table 2.2. Results of each individual class from RescueNet testing set, along with mIoU % [17].

Conclusively, while both of these proposals provide substantial damage assessment results from analyzing images from their corresponding datasets, they are built upon resource intensive infrastructure, meaning they mostly lack real-time detection capabilities. Since it is regarded as an important aspect of disaster response scenarios, it will be addressed in greater detail in the subsequent chapters.

Chapter 3 - External Resources Utilized in Proposed Pipeline Development

3.1 - Resources for Satellite Imagery Process

xBD Dataset

For the xView 2 Challenge, a competition for automating building damage assessment and overall change detection, held in 2019 by the Defence Innovation Unit (DIU), a new large-scale dataset was built from a collection of high-resolution satellite images, before and after natural disasters [14]. The images were primarily collected from the Maxar/DigitalGlobe Open Data Program [15], which is further discussed in the following subsection. They do not have a fixed GSD, but an upper bound of 0.8 meters per pixel was set, to provide the ability to distinguish particular minute details. As the full release description states, prior to the compilation of xBD, the available resources for training purposes were fairly limited, restricted to a binary distinction of "undamaged"/"damaged" buildings and more often than not focused on a single natural disaster, producing biased results.

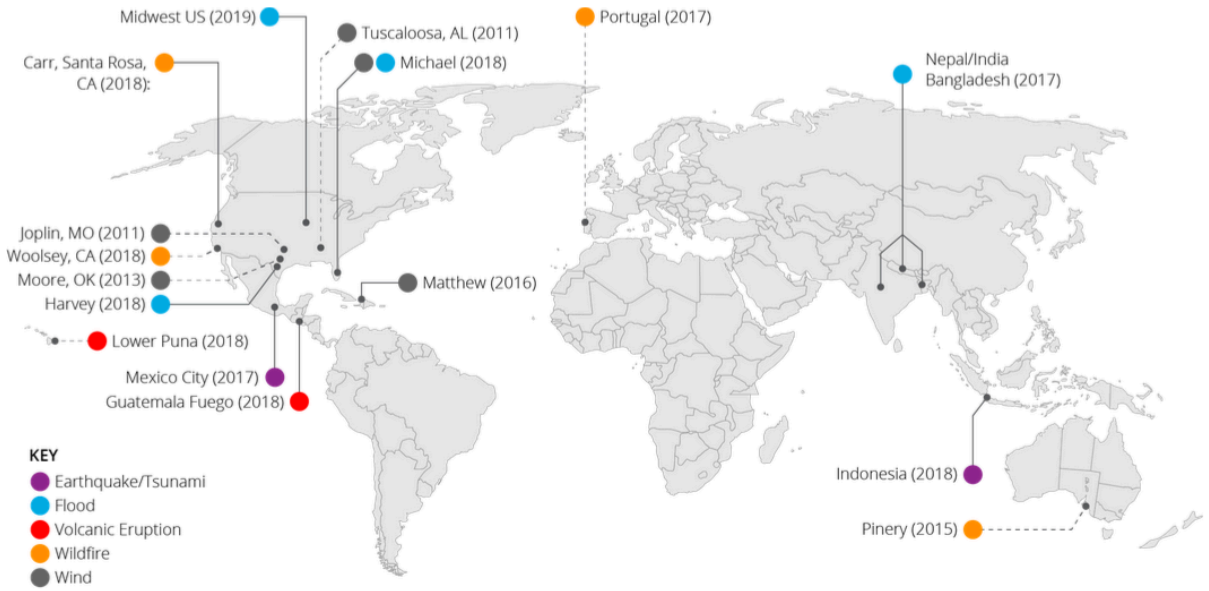


Figure 3.1. World Map featuring events covered in the xBD Dataset, according to disaster type [14].

The following nineteen events, highlighted in Figure 3.1, are covered in the dataset, ranging between six disaster types. Tier 1 events include all imagery sourced from the Open Data Program, while Tier 3 events refer to 8 additional ones, outside the program, once again in collaboration with Maxar.

Disaster Event Name	Event Dates	Tier	Env. Factors
Guatemala Fuego Volcano Eruption	Jun 3, 2018	1	Yes
Lower Puna Volcanic Eruption	May 23 - Aug 14, 2018	3	Yes
Hurricane Michael	Oct 7-16, 2018	1	No
Hurricane Florence	Sep 10-19, 2018	1	Yes
Hurricane Harvey	Aug 17 - Sep 2, 2017	1	No
Hurricane Matthew	Sep 28 - Oct 10, 2016	1	No
Moore, OK Tornado	May 20, 2013	3	No
Joplin, MO Tornado	May 22, 2011	3	No
Tuscaloosa, AL Tornado	Apr 27, 2011	3	No
Pinery Fire	Nov 25 - Dec 2, 2018	3	No
Woolsey Fire	Nov 9-28, 2018	3	No
Carr Wildfire	Jul 23 - Aug 30, 2018	1	No
Santa Rosa Wildfires	Oct 8-31, 2017	1	Yes
Portugal Wildfires	Jun 17-24, 2017	3	No
Midwest US Floods	Jan 3 - May 31, 2019	1	Yes
Indonesia Tsunami	Sep 18, 2018	1	Yes
Sunda Strait Tsunami	Dec 22, 2018	3	No
Monsoon in Nepal, India, Bangladesh	Jul - Sep, 2017	1	Yes
Mexico City Earthquake	Sep 19, 2017	1	No

Table 3.1. Disasters covered in the xBD Dataset [14].

Several events contain further annotations considering environmental factors, indicating smoke, fire, flood water, pyroclastic flow and lava, labeled with corresponding polygons.

The xBD dataset imposes a common labeling method among events, introducing a joint damage scale, ranging from 0 (“No Damage”) to 3 (“Destroyed”), as demonstrated in Table 3.2 below.

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

Table 3.2. Joint Damage Scale according to which the xBD Dataset was annotated [14].

For the damage characterization of buildings, several entities were consulted, including NASA, CAL FIRE, FEMA and the California Air National Guard, as well as past literature, such as HAZUS, FEMA’s Damage Assessment Operations Manual, the Kelman Scale and the EMS-98 scale.

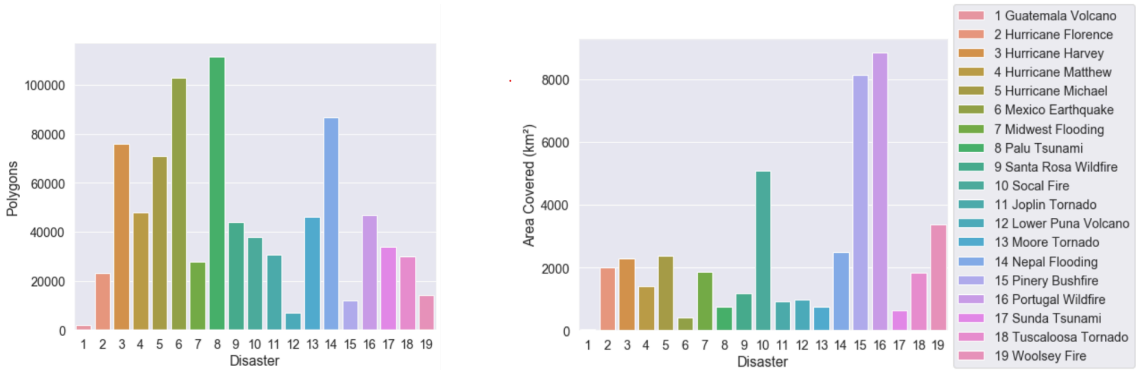


Figure 3.2. Polygons and area covered (km^2) per disaster [14].

Overall, as mentioned above, six disaster types are covered in this dataset, amounting to 22,068 images, with 850,736 labeled building polygons and a total land area of 45,361 square kilometers. The building polygons appearing in each individual disaster, as well as the percentage of positive imagery

are not explicitly correlated with the respective area covered. This illustrates that some disasters are much denser in polygons and positive examples present, such as the Mexico City earthquake or the Palu tsunami, as is evident in Figure 3.2.

Additionally, even if compared to other available datasets xBD is more focused on providing positive examples, meaning showing some degree of damage, the vast majority of polygons are labeled as having no damage. While, negative imagery is essential to the training and validation process, so that the final models return objective results, it has to be noted that even for a highly curated dataset in xBD, there are eight times the polygons with no damage, as there are for all three damage cases combined.

Disaster Level	Number of Polygons
0 (No Damage)	313,033
1 (Minor Damage)	36,860
2 (Major Damage)	29,904
3 (Destroyed)	31,560
4 (Unclassified)	14,011

Table 3.3. xBD dataset number of annotation polygons per disaster level [14].

The available files have been split into three sections, which are notably not common for ML image datasets, “train”, “test” and “holdout”, in a ratio of 80/10/10% respectively.

Split	Images	Polygons
Train	18,336	632,228
Test	1,866	109,724
Holdout	1,866	108,784

Table 3.4. xBD Dataset image splits and their corresponding annotation polygons [14].

This division served the point of the xView 2 Challenge, which in essence was to have a closed-off testing package (“holdout”) to validate each entry’s model, after it was ranked on the open leaderboard. Since this thesis was written after the conclusion of the competition, all splits were

readily available for download and reallocated among the lines of the more conventional train/ validation/ test structure.

The creators of the dataset also present a baseline model, set as a bare minimum proposal for localisation and classification, regarding the xView 2 challenge. For automatically deriving polygons, to later serve as inputs for the classifier, a fork of Motoki Kimura's "SpaceNet Building Detection" was used, which is a modified U-Net approach. These results are then fed into a ResNet50 model, pretrained on ImageNet, coupled with a smaller network (Shallow CNN) with random weights, showcased in Figure 3.3. The process returns a one-hot encoded vector, with each element representing the probability of each individual damage class.

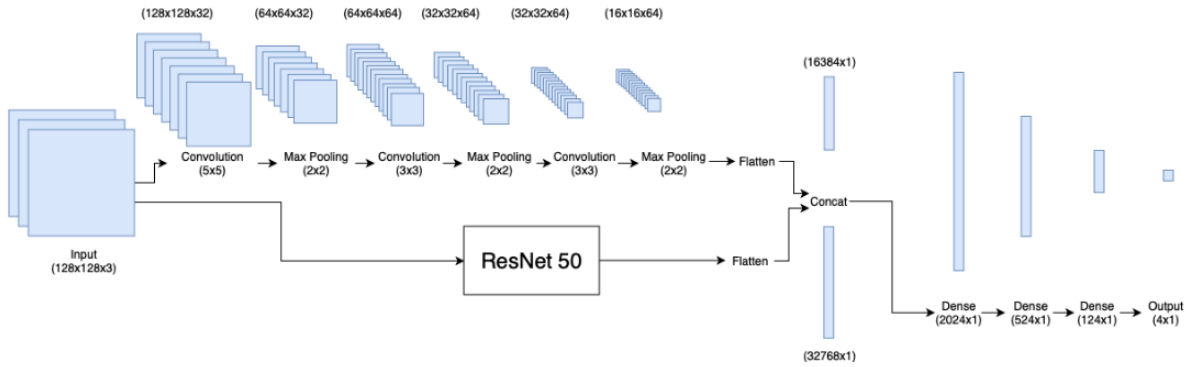


Figure 3.3. Architecture of xBD baseline model [14].

The creators of the model claim to have achieved an overall weighted F1 score of 0.2654, while designating it as the most important metric regarding imbalanced datasets, like xBD. As demonstrated in Table 3.5, both precision and recall drop considerably in the intermediate classes of "Minor Damage" and "Major Damage", since the sample size for either class is significantly lower.

Damage Type	F1 Score	Precision	Recall
No Damage	0.6631	0.8770	0.5330
Minor Damage	0.1435	0.1971	0.1128
Major Damage	0.0094	0.7259	0.0047
Destroyed	0.4657	0.5050	0.4321

Table 3.5. xBD baseline model performance metrics [14].

Concurrently, the discerning features between those classes are fairly minimal as well, leading the model to misclassify “Major Damage” scenarios as “Minor Damage” and in turn substantially lowering the recall. In general, this issue will be explored, with further examples provided in the subsequent chapters.

Maxar Open Data Program

To assist in the humanitarian response to natural disasters, Maxar provides high-resolution satellite imagery before and after such events [15]. These resources are an indispensable part of developing tools to aid in the critical efforts required to assess specific situations, along with preventing further damages. High-resolution satellite imagery, in general, refers to images with a GSD of at least 1 meter, which means that each pixel in an image represents an area of 1 square meter. For the purposes of this thesis, mostly very high-resolution images were sourced, as referenced in our analysis of the xBD dataset, ranging from 0.8 meters GSD, all the way up to 0.3 meters. These images are mostly sourced from Maxar’s WorldView-2 satellite, which provides high-resolution commercial satellite imagery, with 0.46 spatial resolution for panchromatic images. Some pre-event remote sensing data is from the QuickBird satellite, that was in orbit between 2001 and 2014 and had 61 cm panchromatic resolution, while some more recent imagery came from the WorldView-3 satellite, which as of now has the highest available panchromatic resolution at 31 cm.

Seeing that most, if not all, imagery coming from Maxar does not incorporate GSD in the metadata, another method had to be implemented for deriving an estimate of this fundamental metric. A process for calculating an approximation of GSD that follows standard geospatial practices as described in 'Remote Sensing and Image Interpretation' by Thomas Lillesand, Ralph W. Kiefer, and Jonathan Chipman [1], incorporating principles from the geospatial software of ESRI ArcGIS documentation, was deployed. The attributes that are standard among all GeoTIFFs and are helpful for calculating a GSD approximation are “Extent”, “Width” and “Height”. Extent contains the values of minimum and maximum latitude in degrees, for both X and Y axes, whereas Width and Height values refer to the amount of pixels on each axis of the image. For streamlining the calculating process, a simple Python script was written, which takes the aforementioned values as inputs and returns the GSD estimate.

It works as follows:

1. Calculation of geographic coverage in degrees, in both X and Y directions, by subtracting the minimum extent values, from the maximum ones.

2. Calculation of GSD estimate in degrees per pixel, in both X and Y directions, by dividing the previously calculated coverage, by the width and by the height respectively.
3. Conversion of GSD estimate to meters per pixel, in both X and Y directions, by multiplying the previously calculated GSD approximation, with the appropriate conversion factor. In WGS 84 for example, which is the geographic coordinate system referenced in Maxar's satellite imagery metadata, 1 degree of latitude is about 111,320 meters, while 1 degree of longitude varies between different latitudes, resulting in a conversion factor of 111,320 multiplied by the cosine of latitude.
4. Calculation of final GSD estimate, based on the average of the two values of GSD in meters per pixel, for X and Y axes respectively.

Microsoft Model

The main model chosen as a baseline for BDA inference using satellite imagery, is a Siamese U-Net CNN approach, developed by Microsoft's AI for Good Research Lab in 2021 [28]. As mentioned before, barring opting for the use of transformers in an implementation, which has, in the majority of cases, been proven to be remarkably resource intensive, some variation of a Siamese U-Net CNN is considered optimal. The deep learning model utilized, contrary to other proposed methods, uses a single pipeline for both building segmentation and damage classification. In essence, it is a slightly simplified version of the one proposed by "An Attention-Based System for Damage Assessment Using Satellite Imagery" by Hanxiang Hao et al. [29], the architecture of which is illustrated in Figure 2.4. The key difference is that no attention mechanism was incorporated and overall fewer convolutional layers were used for the segmentation arm. The architecture resembles the one presented in 2.1, with two U-Net networks, one for the pre-disaster image and one for the post-disaster image. As showcased in Figure 3.4 below, each of the individual U-Net networks provides the pixel-level building segmentation map separately for the pre-disaster and the post-disaster imagery. Damage classification is the part of the process where extracted features from both networks are combined, for the purpose of producing a single classification output for each pair of images. It follows a Siamese approach equivalent to the one also highlighted in 2.1, utilizing shared weights, in which another decoder performs damage classification on the subtracted embedding layers, with several convolutional layers. Loss functions are utilized for segmentation and classification, with weighted binary cross-entropy loss for the first and multi-label cross-entropy loss for the latter [28].

Although accuracy is always regarded as an essential metric for the superior functionality of a model, when pertaining to humanitarian responses, efficient time management is to a similar extent, if not

greater of a priority. Satellite imagery, in general, can be harnessed for a primary estimation of the extent of damage imposed by a particular disaster, however it is not possible to discern intricate elements. Additionally, even though remote sensing is one of the first tools deployed for assessing an urgent situation, there still has to be an interval between 2 to 6 hours, depending on satellite availability, for the images to become available for emergency response agencies. Therefore, the time strain in conjunction with the often vast amount of land area that has to be evaluated, render time-saving measures as an imperative feature. The publication associated with the dataset provides a less commonly utilized metric for ML implementations, which is inference time. More specifically, the authors measured the inference time of the top five ranking models in the xView 2 challenge, apart from the fourth placed one which was not reproducible, against their own proposal, as demonstrated in Table 3.6 below.

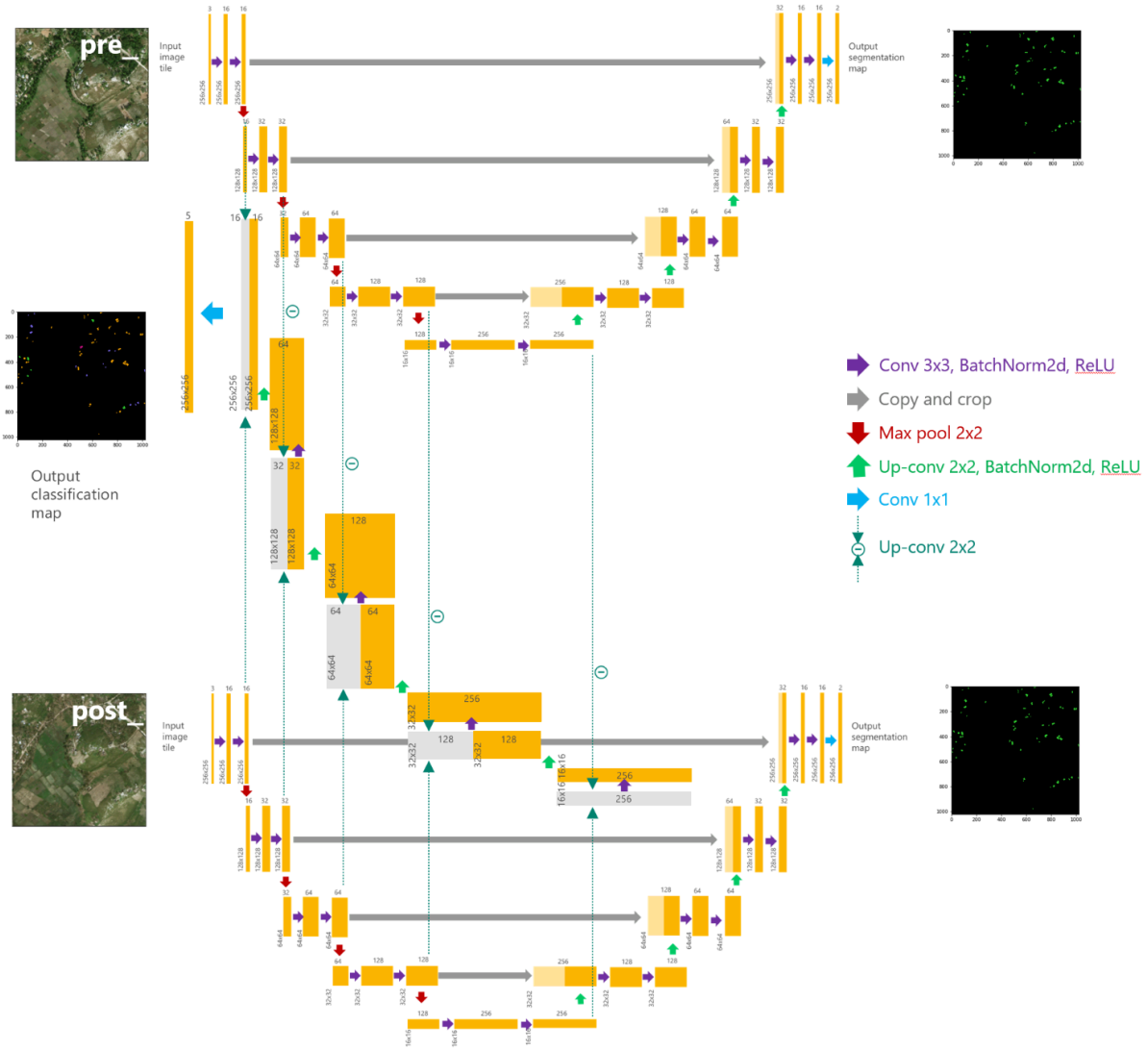


Figure 3.4. Microsoft model Siamese U-Net architecture overview [32].

Method	Inference time (s)	km^2 / h
xView 2 1st place	245.75 (0.73)	1.38
xView 2 2nd place	121.03 (0.36)	2.81
xView 2 3rd place	108.21 (0.6)	3.14
xView 2 5th place	10.94 (0.06)	31.07
Microsoft Model	3.8 (0.02)	89.35

Table 3.6. Inference time on a single 1024x1024 pixel tile (w/ 0.3 m/pixel resolution) from top ranking implementations from the xView 2 competition and the Microsoft Model, using a single TESLA K80 GPU [28].

Under the same circumstances, the disparity in inference time on xBD imagery is significant. The structure of the xView 2 competition encouraged participants to forego temporal efficiency entirely, in the interest of leaderboard performance, which was calculated solely according to F1 score and subsequent metrics. This led to most of the standout implementations, as well as several models trained on xBD, that were released in the years following the conclusion of the challenge, to use larger overall models, at the expense of inference speed. Whereas the majority of the field, which is fairly limited to begin with, outputs similar approaches for BDA, only a minority of said publications is pertaining in any way throughout the whole process to inference time. This drives up resources' cost, whether referring to initial hardware investments, or cloud computing fees, along with hindering the humanitarian response capabilities of the organizations responsible.

Since most of the top placing implementations include some form of attention mechanism, the overall inference latency is increased substantially. While this approach returns improved results, its adaptability in real world scenarios is limited. The authors claim for example, that for Hurricane Ida the Maxar Open Data program released around 20,000 km^2 of pre and post disaster imagery [15], [28]. For the first placed solution in particular, which performs inference with a 12 model configuration that is run 4 times per input, on a single TESLA K80 GPU available in Azure, BDA would require around 300 days. In contrast, the present model would require 4.7 days, eliciting discussion regarding where the line can be drawn between accuracy and temporal efficiency in the trade-off [28].

Performance-wise the Microsoft Model exhibited similar results to the baseline proposal of the creators of xBD, in terms of building segmentation, which is to be expected since both utilize modified U-Net approaches without attention mechanisms. The vast improvement recorded is

regarding damage classification, where the relevant model outperforms the baseline one by a wide margin. More precisely, the baseline model had a 0.03 weighted F1 score for solely the damage classes, while the Microsoft model recorded 0.58 for the same metric. Overall, the model returned positive results for various splits, encompassing different potential scenarios, with the F1 scores considered adequate for the task at hand. Some examples provided by the authors are featured in Table 3.7, while Figure 3.5 showcases three distinctive instances of 1024x1024 tiles, with varying degrees of damage, along with their corresponding label masks and model predictions.

Experiment	Train	Test	BLD-1	DMG-0	DMG-1	DMG-2	DMG-3	DMG-mean
Random splits	80% at random	10% at random	0.74	0.89	0.43	0.54	0.73	0.60
Joplin held out	90% of non - Joplin	Joplin only	0.76	0.89	0.50	0.36	0.81	0.56
Joplin held out (wind only damage classifier)	90% of non - Joplin	Joplin only	0.74	0.89	0.42	0.54	0.77	0.60
Nepal held out	90% of non - Nepal	Nepal only	0.63	0.42	0.17	0.23	0.02	0.06
Nepal held out (flood only damage classifier)	90% of non - Nepal	Nepal only	0.64	0.54	0.12	0.27	0.07	0.14

Table 3.7. Pixel-wise F1 Score on various splits of the xBD dataset [28].

In general, this demonstrates the model's potential generalizability, which is another advantage of choosing a non-attention layer implementation, since it can return results for different disaster types, without extensive training for each one individually. This way overfitting scenarios are avoided, especially for underrepresented classes and event categories in the case of the xBD dataset, along with

performing model training and inference with considerably less computational demands. Finally, the model presents predictions with mixed damage labeling for each building, providing a more substantial overview for the given assignment.

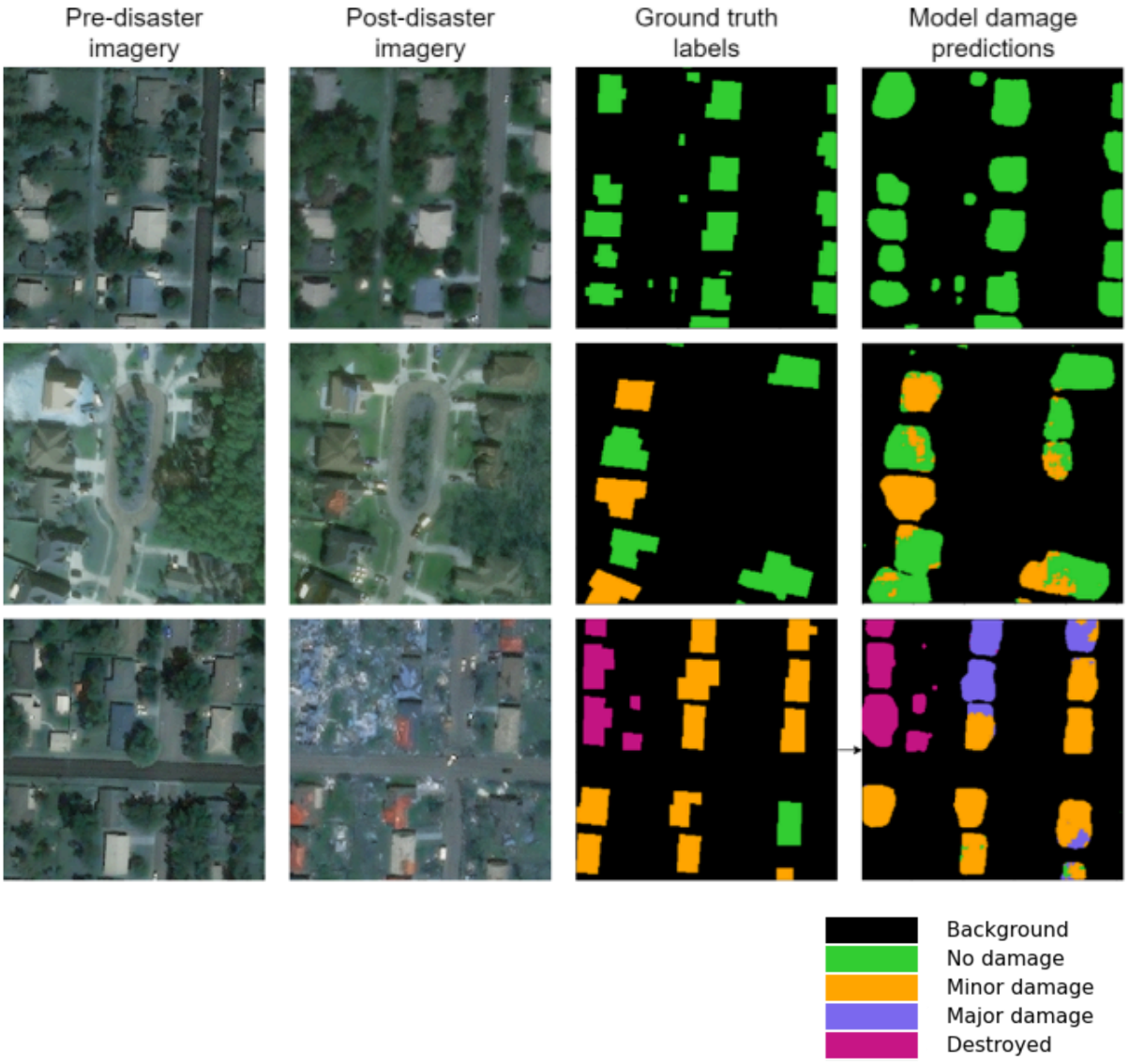


Figure 3.5. Model prediction examples from the xBD dataset, with variations in damage scale [28].

3.2 - Resources for Aerial Footage Process

ISBDA Dataset

In 2020, a new BDA dataset was produced by Xiaoyu Zhu, Junwei Liang and Alexander Hauptmann, stemming from drone footage of various natural disasters [16]. The first of its kind, as drone imagery recently became more widely available, it provides an opportunity to discern damage features in buildings, which are simply not possible from satellite imagery exclusively. The images are sourced from user-generated aerial videos from social media, covering areas affected by Hurricane Harvey in 2017, Hurricane Michael and Hurricane Florence in 2018, as well as three tornadoes occurring between 2017 and 2019. The footage consists of mostly areas located in the American south (Florida, Texas, Alabama, North Carolina) and the midwestern US (Illinois, Kansas, Missouri). From videos adding up to 84 minutes in aggregate, individual frames were singled out as long as they complied with the following guidelines.

Collectively, 1030 images from 10 videos overall are annotated with instance-level building masks, along with damaged part masks. These damage masks amount to 2961 in total, which are divided into a damage scale of three levels “Slight”, “Severe” and “Debris”, in accordance with the damage assessment practice of “Joint Damage Scale”. As previously defined by the creators of the xBD dataset, Slight means there are visible cracks or appearance damages, Severe means that there is partial wall or roof collapse and Debris means that the building has completely collapsed.

Adhering to Microsoft COCOn’s size definition, the overall damaged part instances in different sizes were calculated, for each corresponding damage level, resulting in Table 3.8, where an area is defined as “Small” if it covers less than 32x32 pixels in the segmentation mask, “Medium” if it covers between 32x32 and 96x96 pixels and “Large” if it covers more than 96x96 pixels.

Damage Scale	Small	Medium	Large	Total
Slight	204	1169	746	2119
Severe	-	120	440	560
Debris	-	54	228	282

Table 3.8. Damaged part instances in ISBDA dataset, divided according to scale [16].

The creators of the dataset proposed their own model called MSNet, for demonstrating the capabilities of their work, along with providing the first widely available implementation for aerial footage BDA, which is presented in further detail, in 2.4.

DoriaNET Dataset

In 2021, a visual dataset for post-disaster building damage assessment was produced by Chih-Shen Cheng, Amir H. Behzadan and Arash Noshadravan, consisting of aerial footage from Hurricane Dorian, which occurred in the Bahamas in August of 2019 [33]. It builds upon the data provided by ISBDA, by containing global coordinates, meaning latitude and longitude of each building, local coordinates with pixel-level masks in each video frame, wind-induced damage state, building characteristics and an ordinal score representing damage annotation effort for each building. For this dissertation, only damage masks were utilized, since the proposed process does not use image metadata, such as coordinates in some way. In Table 3.9, the six total damage categories and their corresponding descriptions are showcased, expanding on the Joint Damage Scale from the xBD dataset.

Damage State	Quantitative Damage Description
0	No Damage or Very Minor Damage Little or no visible damage from the outside. No broken windows. or failed roof deck. Minimal loss of roof over. with no or very limited water penetration
1	Minor Damage Maximum of one broken window, door or garage door. Moderate roof cover loss that can be covered to prevent additional water entering the building Marks or dents on walls requiring painting or patching for repair
2	Moderate Damage Major roof cover damage, moderate window breakage Minor roof sheathing failure Some resulting damage to interior of building from water
3	Severe Damage Major window damage or roof sheathing loss. Major roof cover loss. Extensive damage to interior from water.
4	Destruction Complete roof failure and/or, failure of wall frame. Loss of more than 50% of roof sheathing
5	Buildings under Construction Completed destroyed or under construction

Table 3.9. Description of DoriaNET dataset damage scale, according to FEMA guidelines [33].

Overall, the DoriaNET dataset includes 271 images, which include 147 annotated buildings, stemming from three 5-minute long 1280x720 pixel UAV videos recorded at 30 frames per second (FPS).

RescueNet Dataset

In late 2023, a new high-resolution UAV semantic segmentation dataset for natural disaster management, called RescueNet, was created by Maryam Rahnemoonfar, Tashnim Chowdhury & Robin Murphy [17]. It is the most substantial widely-available dataset for aerial footage pertaining to the given assignment, as of the time of writing this dissertation. RescueNet features aerial imagery from Hurricane Michael, which occurred in an area near Mexico beach, Florida in the US, on the 10th of October 2018.

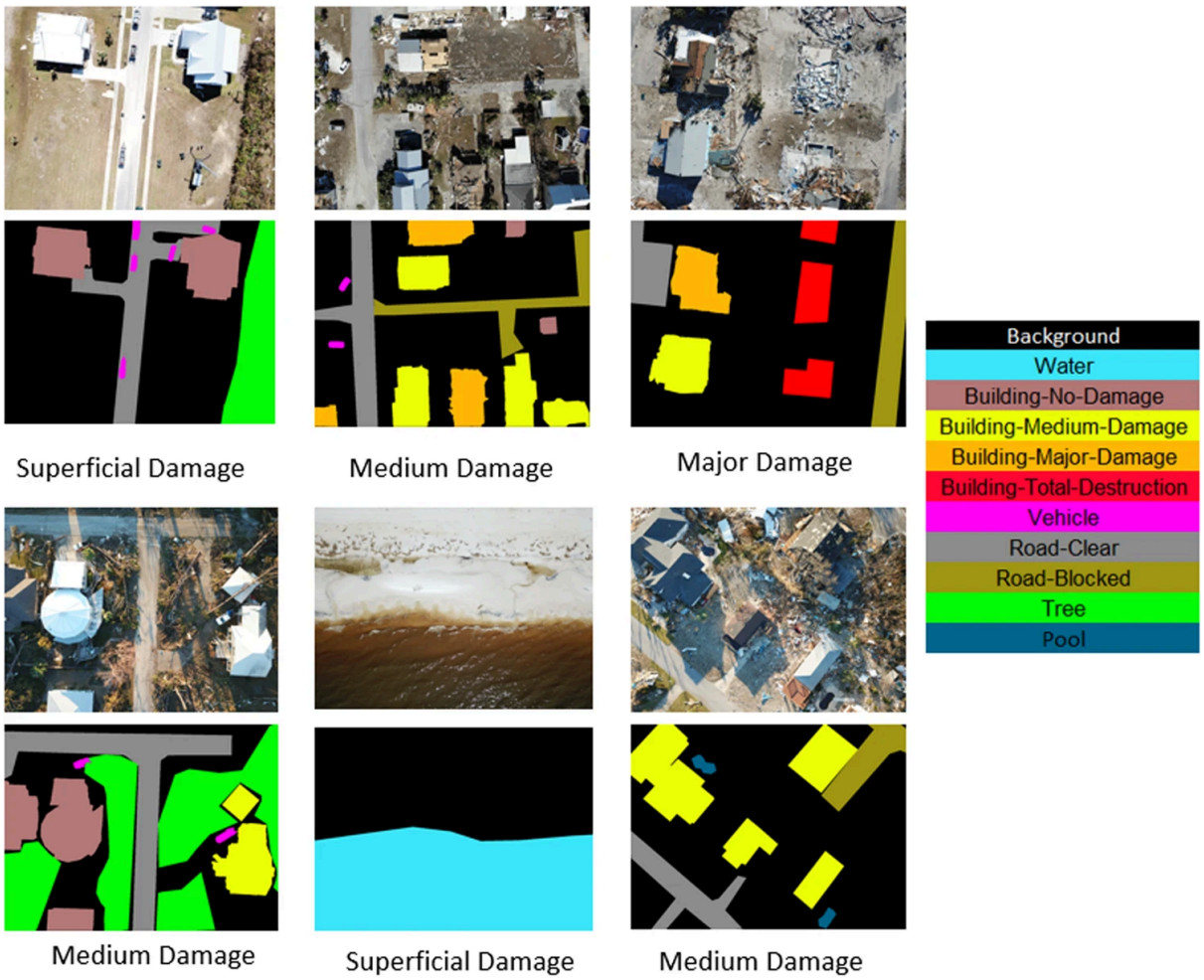


Figure 3.6. Visual representation of RescueNet sample images, along with their corresponding annotations [17].

It includes annotations for 10 different classes, as illustrated in Figure 3.6, including building damage according to the Joint Damage Scale, along with road status and environmental factors, such as vegetation and water presence. The authors justify the inclusion of extended annotations, by claiming that environmental and road data compliment structural damage information provide a more complete understanding of each scene. Since supplemented imagery is solely using an orthographic view, meaning all images are taken using a top-down angle, RescueNet is planning to be utilized for scenarios similar to those of satellite imagery, while capturing images at considerably higher resolution. In total, it includes 4494 annotated images, with a fixed resolution of 3000x4000 pixels and a more balanced representation of building damage labels per class than the xBD dataset, as evident in Table 3.10.

Damage Level	Number of Polygons
No Damage	4011
Minor Damage	3119
Major Damage	1693
Destroyed	2080

Table 3.10. RescueNet dataset number of annotation polygons per disaster level [17].

YOLO v9

The YOLOv9 model was the release chosen for implementing the proposed method for this dissertation, since it was the latest one to have wide adaptability in terms of dataset annotations and overall generalizability. Architecture-wise YOLOv9 introduces several innovations, enhancing both detection accuracy, along with inference speed, all while integrating state-of-the-art techniques , mainly pertaining to the neck module [26]. As showcased in Figure 3.7, the Reversible Column (RevCol) in the backbone improves memory efficiency, an advanced Path Aggregation Network (PAN) and Programmable Gradient Information and (PGI) in the neck enhance multi-scale feature fusion and global context interaction, while Deep Supervision spans all across the network, for helping with training stability. Concurrently, dynamic anchor boxes and refined bounding box regression improve accuracy in the head module, while advanced augmentation techniques boost generalization during training.

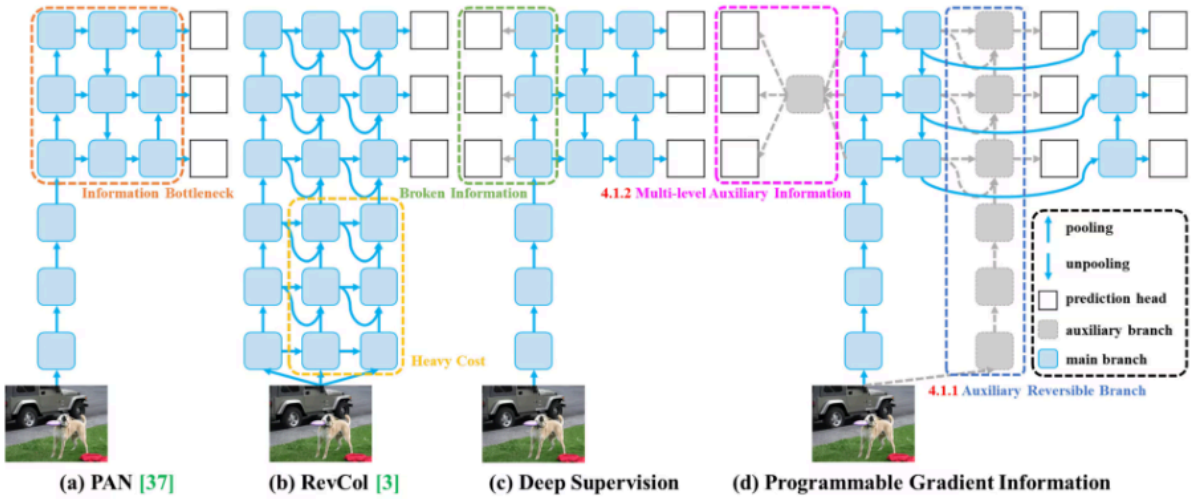


Figure 3. PGI and related network architectures and methods. (a) Path Aggregation Network (PAN)) [37], (b) Reversible Columns (RevCol) [3], (c) conventional deep supervision, and (d) our proposed Programmable Gradient Information (PGI). PGI is mainly composed of three components: (1) main branch: architecture used for inference, (2) auxiliary reversible branch: generate reliable gradients to supply main branch for backward transmission, and (3) multi-level auxiliary information: control main branch learning plannable multi-level of semantic information.

Figure 3.7. YOLOv9 components architectures and methods [26].

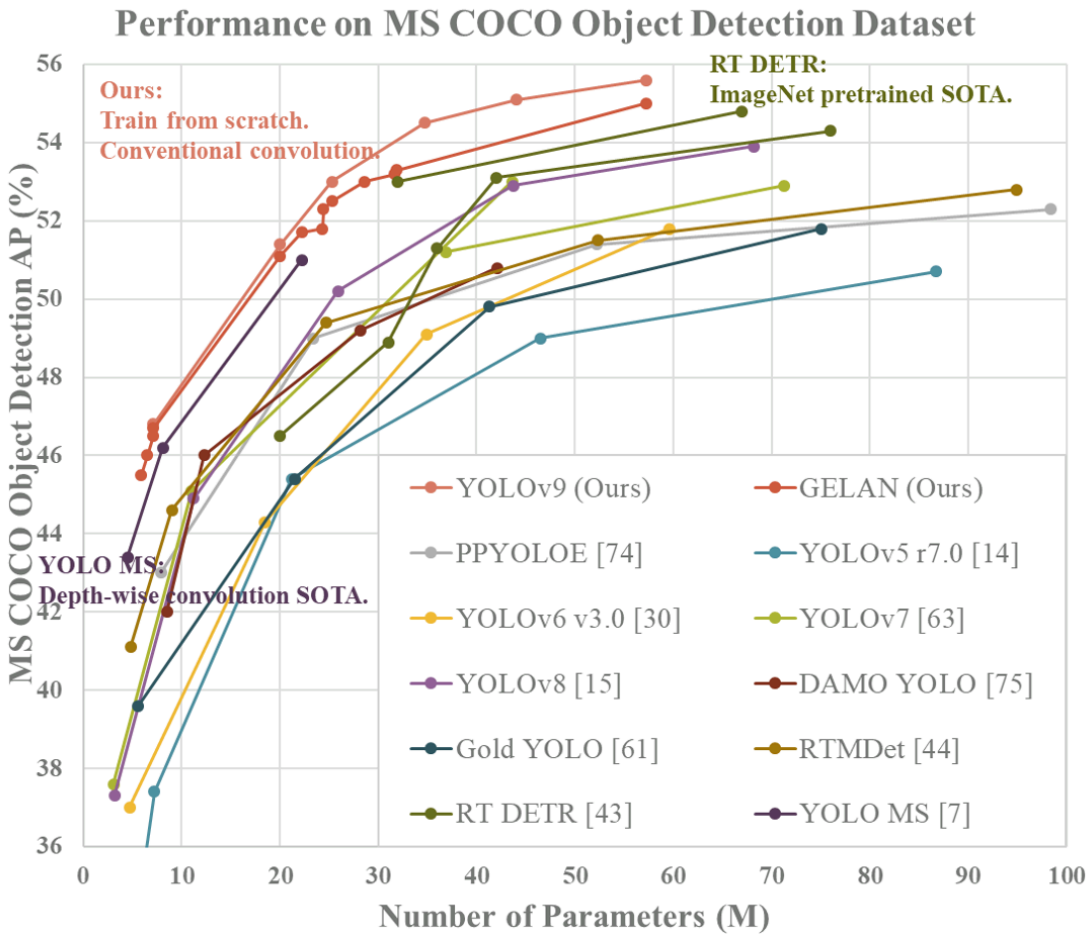


Figure 3.8. YOLOv9 performance on the MS COCO dataset, compared to other popular object detectors [26].

YOLO models are typically pre-trained on the MS COCO Object Detection Dataset, as is the case with YOLOv9, providing a baseline model for general object detection tasks, which can then be fine-tuned further depending on the demands of each individual application. Figure 3.8 presents the advancements noted by YOLOv9, in terms of accuracy, along with computational demands, compared to other object detection algorithms, on the MS COCO dataset.

Below, in Table 3.11 and Table 3.12, are the available releases for YOLOv9 for detection and segmentation respectively, along with some performance metrics on the MS COCO dataset generated from Ultralytics.

Model	**size** (pixels)	**mAP^{val}** 50-95	**mAP^{val}** 50	**params** (M)	**FLOPs** (B)
YOLOv9t	640	38.3	53.1	2.0	7.7
YOLOv9s	640	46.8	63.4	7.2	26.7
YOLOv9m	640	51.4	68.1	20.1	76.8
YOLOv9c	640	53.0	70.2	25.5	102.8
YOLOv9e	640	55.6	72.8	58.1	192.5

Table 3.11. Performance metrics for Object Detection YOLOv9 model releases [34].

Model	**size** (pixels)	**mAP^{box}** 50-95	**mAP^{mask}** 50-95	**params** (M)	**FLOPs** (B)
YOLOv9c-seg	640	52.4	42.2	27.9	159.4
YOLOv9e-seg	640	55.1	44.3	60.5	248.4

Table 3.12. Performance metrics for Image Segmentation YOLOv9 model releases [34].

As evident from the two tables, both lightweight and medium to large model releases are available for object detection, with large variation in computational needs, from just 7.7 billion FLOPs for the t (tiny) model release to 192.5 for the e (extensive) one. Meanwhile, real-time instance segmentation remains a tremendously resource intensive process, despite notable advancements in the sector, with the lightest model requiring 159.4 billion FLOPs for a single forward pass. However, even lighter models for both processes provide substantial results in terms of accuracy, showcasing improvements

from earlier versions, as noted by the mAP metrics,as well as efficiency and adaptability, rendering YOLO as one of the most valuable tools for object detection [34].

In terms of the objectives of this dissertation, bar some older projects on Roboflow, no BDA implementations with newer YOLO models were found. However, as referenced above, annotated datasets are more oriented towards object detection, than pixel segmentation, so a model with real-time detection and classification capabilities was deemed optimal. Ultimately, YOLO provides a framework that can both detect buildings, as well as classify them on the arbitrary Joint Damage Scale, while also providing results in real-time, therefore rendering it as an essential tool for disaster response situations.

Chapter 4 - Methodology

4.1 - Process Overview and Graphical User Interface Implementation

The proposal in its entirety comprises a unified pipeline for BDA, encompassing both types of remote sensing data analysis, meaning from both satellite and aerial imagery and deriving damage reports from both scenarios. The final implementation is presented through a graphical user interface (GUI), developed using the standard, built-in Python library Tkinter, in accordance with the themes of scalability and compatibility, highlighted throughout this project. The user is presented with an initial choice of running inference for BDA, on either satellite images or drone footage. Afterwards, an input of two satellite images or a folder of drone image frames is requested, which is then checked for compliance with the corresponding proccess's requirements, as they are outlined in the following segments. Ultimately, the program returns a colored damage map that can be further analyzed in GIS software, or the UAV images with bounding boxes highlighting the YOLO-based model's predictions.

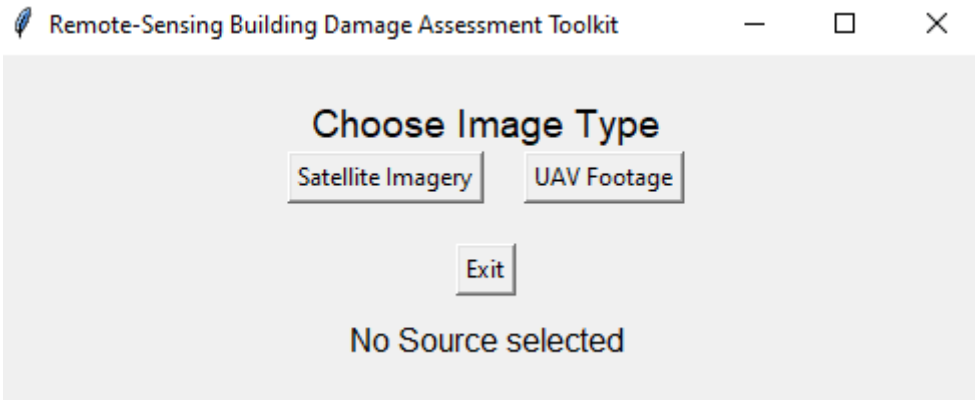


Figure 4.1. Screen capture of GUI main menu.

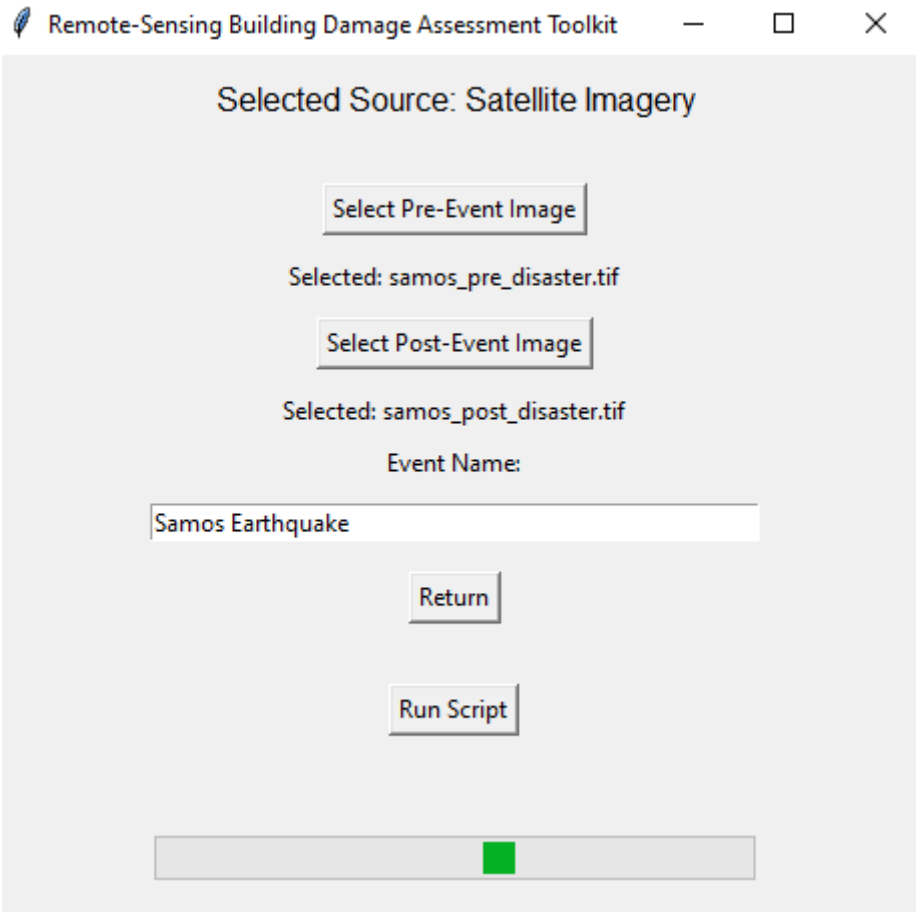


Figure 4.2. Screen capture of GUI, running inference on satellite imagery from Samos earthquake.

4.2 - Satellite Imagery Process

Despite the fact that, as claimed by the research paper accompanying Microsoft's implementation, their approach is an end-to-end model, several modifications were made to the inference pipeline, for facilitating the applicability to imagery beyond the xBD dataset. Since inference requires, aside from the standard input of an imagery folder path, the mean and standard deviation of each image, along with a JSON file referencing all image file paths, the Python code provided had to be altered. The objective of these adjustments was to provide a framework that only has two pre-aligned GeoTIFF images as inputs, in order to further streamline the overall process.

The first step, nonetheless, is preparing the GeoTIFF files to be in the appropriate format for deriving results from their analysis. Essentially, this implies that the pre-disaster and post-disaster images need to both refer to the exact same geographical area, as well as having the same amount of pixels. In order to uncomplicate the experience for the end user, alongside it generally being the most competent solution, any GIS software can be used for refactoring the images to the desired output. Given that

QGIS is a free and open-source project, together with it being cross-platform, all trials were conducted utilizing it.

After the images are formatted accordingly, the required files for the inference process are generated by deploying several Python scripts for splitting the large GeoTIFF images into 256x256 PNG tiles, along with generating the aforementioned JSON files containing referential info on the PNG tiles.

Firstly, the two images are divided into tiles and the last tile of each line is white padded if necessary. No information is lost in this process since there is no compression occurring, just splitting based on pixel count. Then, the appropriate naming format, alluding to the xBD dataset's naming convention, is added, separating pre and post disaster images in the process. As explained in Chapter 1, mean and standard deviation are important metrics for normalizing and preprocessing input data, leading to better model performance, as well as aiding the main point of this part of the dissertation, which is generalization beyond xBD images. Therefore, another Python script was implemented for calculating these statistics for each individual image as is highlighted in 1.3. Additionally, a JSON file indicating the file paths of each pair of tiles is generated, with the possibility to also divide the image pair paths into train/ validation/ test splits for further model refinement, through extra training. As expected, building segmentation and damage masks have to be supplemented for attaining improvements in the model.

With all of these resources set in place, all that is left is for model inference to be carried out, by running the corresponding Python file, utilizing the following inputs. The image directory, the two generated JSON files, an additional JSON file for mapping damage level labels, the folder containing the pretrained model and a directory for the outputs to be stored. This process returns a folder of 256x256 grayscale PNG images with pixel values ranging from 0 to 4, where 0 means background and 1 to 4 aligns with the Joint Damage Scale indicators, where 1 equates to a building part with zero damage and 4 to a building part that has been destroyed. Some complementary evaluation metrics are provided in the form of comma-separated values (CSV) files, showcasing how many pixels of each damage level are found in every picture. Moreover, true/ false positive and negative evaluation metrics are supplemented, on the condition that building and damage masks are made available.

However, an apparent problem that arises with these results is that they cannot be displayed using conventional software, since the variations in pixel values are practically indiscernible. In the same way, even with the damage labeling color map applied, observing every single 256x256 tile separately, does not allow the user to form a coherent overview of the situation at hand, especially considering that there might be thousands of tiles to inspect. Bearing that in mind, another Python script was written to produce a "mosaic" of all damage map tiles and apply transparent background.

This has the damage grayscale 256x256 PNG tiles as inputs, as well as the number of tiles in a single line of the original GeoTIFF images. It returns a single PNG image, with RGB colors according to the corresponding damage labels applied, featuring the length and width of the original large images. From then onwards, this PNG image can be imported in any GIS software as a raster layer and by adding the appropriate georeferencing information, it can be overlaid precisely on the source imagery.

After performing these modifications, evaluation metrics were derived from inference on the entirety of the xBD dataset. The subsequent results in Table 4.1 and Table 4.2 showcase the capabilities of the model in all types of disasters, with overall improved results from the ones presented by the baseline model.

Damage Class	Precision	Recall
Building Segmentation (0.0 - 4.0)	0.47174560470701377	0.8969967284980053
1.0	0.90120913955129	0.8302602395572306
2.0	0.4006328189110755	0.2218038170521118
3.0	0.4267668106325847	0.4776434564739261
4.0	0.6758474766592797	0.6138595591618061

Table 4.1. Precision and Recall from model inference on the entirety of the xBD dataset.

Damage Class	F1 Score	Accuracy
Building Segmentation (0.0 - 4.0)	0.6183110638722704	0.9667699259870193
1.0	0.8642810841740255	0.7961483309927403
2.0	0.2855291071783234	0.9240021783371831
3.0	0.4507741275230238	0.8915882069631587
4.0	0.6433638377704219	0.9746595717169235
Harmonic Mean of All (1.0 - 4.0)	0.47438120446765264	-

Table 4.2. F1 Score and Accuracy from model inference on the entirety of the xBD dataset.

The only damage classes that returned subpar metrics were the intermediate ones, as is the case with all models trained on xBD, because as discussed in 3.1, they are both underrepresented and it is fairly challenging to discern features between them. Since building predicted polygons are connected at some instances and calculated true positive numbers are underestimated, pixelwise F1 score is regarded as the most reliable metric overall. This provides an important overview, unavailable for most models, given that results in many publications are difficult to reproduce accurately. In Chapter 5, results from various events, including disaster types that were not included in the xBD dataset, are examined more thoroughly.

To summarize, using a pretrained model that yielded results exclusively from the xBD dataset, that were difficult for the end user to interpret and after performing several necessary modifications, a refined program for satellite imagery BDA is presented. A simplified process that solely requires two satellite images covering the same area as inputs and outputs a dynamic mosaic that immediately accentuates sections that demand swift humanitarian response. While similar solutions for BDA on areas of smaller scale may have been available, this work is unique in terms of providing a streamlined process for larger scale areas, that features great generalizability, coupled with superior performance when it comes to speed, as outlined in Chapter 3.

Inputs	2 GeoTIFF satellite images of identical resolution, captured before and after the events.
Outputs	Transparent PNG damage map of buildings, with resolution of input images.
	CSV files containing statistics for building and damage instances.
	Individual grayscale 256x256 tiles of model predictions.

Table 4.3. Inputs and outputs of satellite imagery process.

4.3 - Aerial Imagery Process

For the second part of the methodology of this dissertation, a model was trained on the aforementioned ISBDA and DoriaNET datasets, to perform BDA on imagery acquired from aerial drone footage, after natural disasters. This additional pipeline provides the capability to further scout

areas highlighted from the previous satellite imagery process, which seem to be affected to a greater extent by such events. It can be the final step for remote sensing before deploying coordinated groups to assist in the humanitarian response, with ample knowledge of the situation at hand.

The model is trained using the YOLOv9 object detection algorithm, which was presented in 3.2, and performs both object detection and classification on a single image input. The process's essential feature is its real-time detection, provided by YOLO, which further speeds up the entire process. In terms of this thesis, YOLO leverages its high-accuracy, real-time object detection capabilities for BDA, from either satellite or aerial imagery, especially in disaster response efforts, where efficient time management is essential for allocating resources and managing rescue operations. On top of that, YOLO's lightweight architecture makes it optimal for scalability, since it has the ability to be deployed on multiple platforms, such as drones and mobile phones, rendering it a key aspect of on-site damage assessment, alongside remote assessment efforts.

Imagery sourced from ISBDA and DoriaNet datasets was used for training, with 10,354 total annotations across all four classes, since at the time this dissertation was produced they were the only two datasets with consistent labeling and easily adaptable to the already established process for satellite imagery.

Damage Level	Number of Annotations
No Damage (0)	4,927
Minor Damage (1)	3,362
Major Damage (2)	1,358
Destroyed (3)	707

Table 4.4. Aerial imagery dataset annotations per damage level for ISBDA and DoriaNet imagery used in the proposed pipeline.

As noted in 3.2, they cover several tornadoes that affected suburban areas of the United States, during the past decade. Additional fine-tuning was provided by utilizing the RescueNet dataset, also presented in 3.2, which features top down aerial imagery from Hurricane Michael, in Florida, USA. The final dataset annotations amount to the class splits shown in Table 4.5, with 13,298 total annotations across all four classes. It features a different distribution of annotations per damage level class, compared to popular datasets like xBD, by providing more intermediate examples, for the purpose of testing whether it affects model performance.

Damage Level	Number of Annotations
No Damage (0)	6,242
Minor Damage (1)	3,362
Major Damage (2)	2,155
Destroyed (3)	1,539

Table 4.5. Aerial imagery dataset annotations per damage level aggregate.

As explained in 3.2, each YOLO iteration features several models for object detection with varying computational demands, ranging from light-weight to large models. To provide a more comprehensive understanding of the capabilities of YOLOv9 for BDA using aerial footage, training and testing were conducted on multiple models, yielding notable results. Overall, distributions ranging from YOLOv9t to YOLOv9c, as illustrated in Table 3.11, were trained using the aforementioned dataset imagery. In the subsequent pages, figures produced during the training and validation processes are presented.

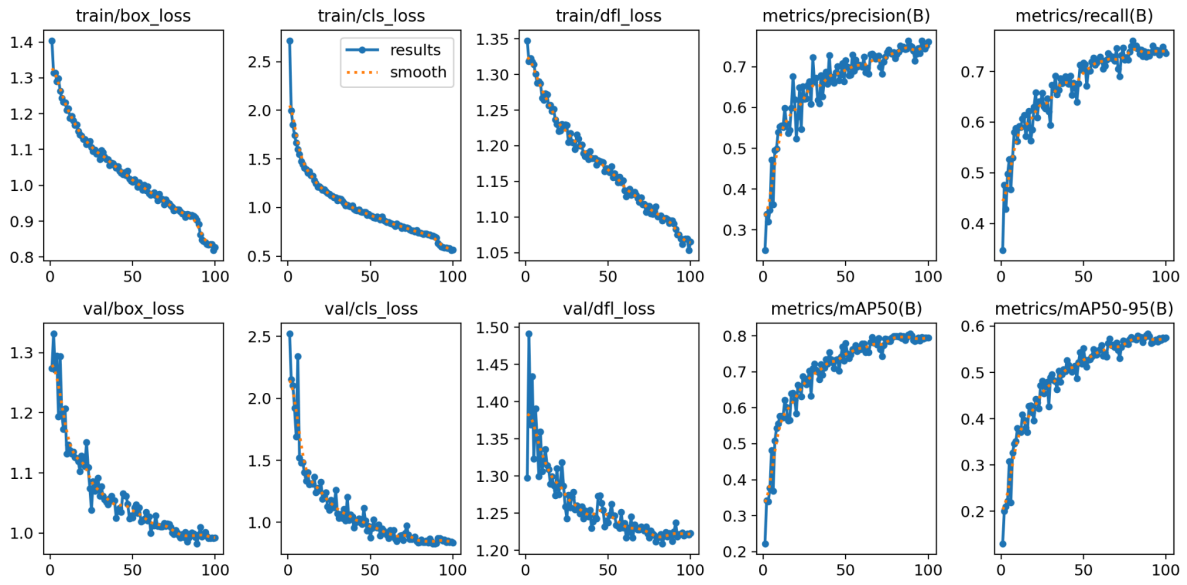


Figure 4.3. Training and validation losses, along with performance metrics with YOLOv9t model, for 100 epochs.

In Figure 4.3, an assortment of graphs produced during training and validation with the YOLOv9t model is illustrated, where the train/box_loss graph indicates the model's ability to predict the correct bounding box coordinates for detected objects, while the train/class_loss graph measures how well the model classifies objects into their respective categories. The train/dfl_loss (Distribution Focal Loss)

focuses on enhancing the model's ability to handle difficult-to-classify instances by adjusting the loss function based on the quality of predictions. Similarly, validation losses, represented by val/box_loss, val/class_loss, and val/dfl_loss, provide insights into how well the model generalizes to unseen data during validation. Additionally, the precision-recall graph assesses the model's ability to maintain high precision while maximizing recall, with mAP@0.50 (mean Average Precision at IoU threshold of 0.50) and mAP@0.50:0.95 (mean Average Precision averaged across multiple IoU thresholds) serving as comprehensive metrics for evaluating overall performance. For comparison, the corresponding graphs on the more intensive YOLOv9c model yielded similar results, which can be attributed to the size and diversity of the dataset. Moreover, while most performance metric curves stopped improving in less overall epochs, with the best weights being produced after the 87th epoch concluded, the final metrics were almost identical to the lighter model ones, as evident in Figure 4.4.

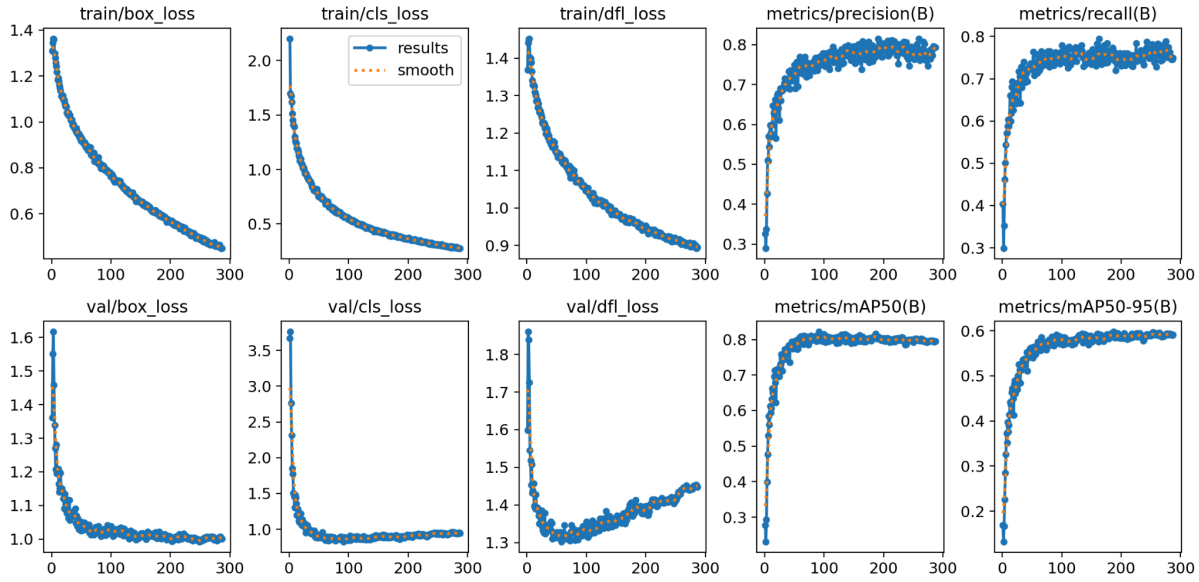


Figure 4.4. Training and validation losses, along with performance metrics with YOLOv9c model, for 300 epochs.

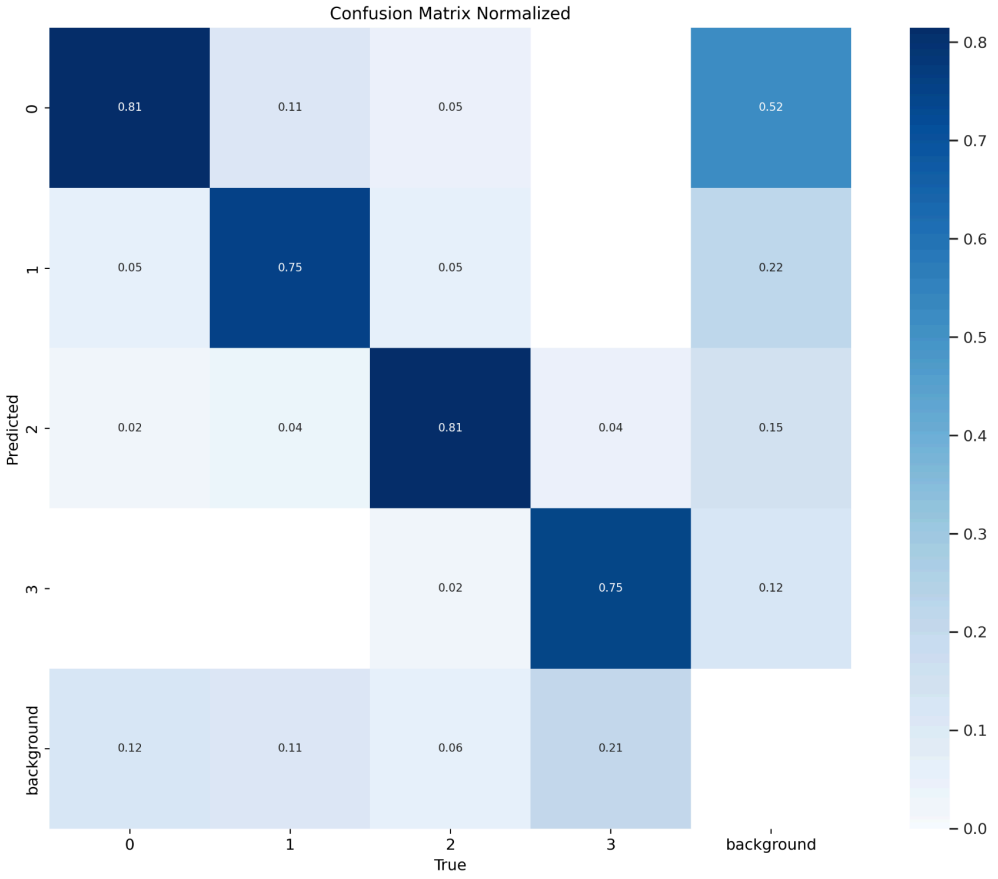


Figure 4.5. Normalized confusion matrix from YOLOv9c training.

YOLO also provides an NxN normalized confusion matrix, where N represents the number of classes in the training dataset. The higher the values in the main diagonal, the more accurately the model identifies both positive and negative instances, resulting in improved overall performance. Conversely, lower values in the off-diagonal elements, associated with False Positives and False Negatives, indicate fewer misclassifications, which is essential for a robust classification model. Figure 4.5 illustrates the efficacy of the proposed model, showing generally high values across the main diagonal. The only noticeable misclassification arises from background elements incorrectly labeled as buildings, analogous to the class splits, which is expected due to some disparity in the bounding boxes.

Additionally, YOLO produces four curves, namely F1-Confidence, Precision-Confidence, Precision-Recall, and Recall-Confidence, which collectively evaluate the trade-offs and relationships between precision, recall, and confidence levels in the object detection performance of the YOLO model. The F1-Confidence curve (Figure 4.6) illustrates the relationship between the F1 score and the confidence threshold of predictions, helping assess how changes in confidence affect the balance between these metrics. The Precision-Confidence curve (Figure 4.7) plots precision against

confidence scores, providing insights into how precision varies with different confidence levels, where high precision at a specific threshold indicates accurate predictions. The Precision-Recall curve (Figure 4.8) depicts the trade-off between precision and recall, with the area under this curve (AUC-PR) serving as a measure of the model's overall performance. Finally, the Recall-Confidence curve (Figure 4.9) shows the relationship between recall and confidence scores, helping visualize how the model's ability to identify true positives varies with the confidence threshold.

The workflow has a folder of images as input and returns the same images with bounding boxes indicating houses, damage level, as well as a confidence score for each individual prediction. Overall, the proposed model analyzes more specific cases identified during the satellite imagery analysis and generates damage-classified bounding boxes for each individual instance. As each image requires individual attention, no general assessment is provided and evaluations are made separately for each case. Ultimately, the designed methodology employs YOLO for the efficient analysis of aerial footage, enabling real-time identification and classification of building damage through automated processing, thus facilitating timely decision-making in disaster response efforts. To conclude, this represents the final step in a unified remote sensing building damage assessment pipeline utilizing Artificial Intelligence techniques, prior to any required human intervention. This automation streamlines a significant portion of the process and enhances efficiency by allowing experts to focus on analyzing results rather than on time-consuming preliminary assessments.

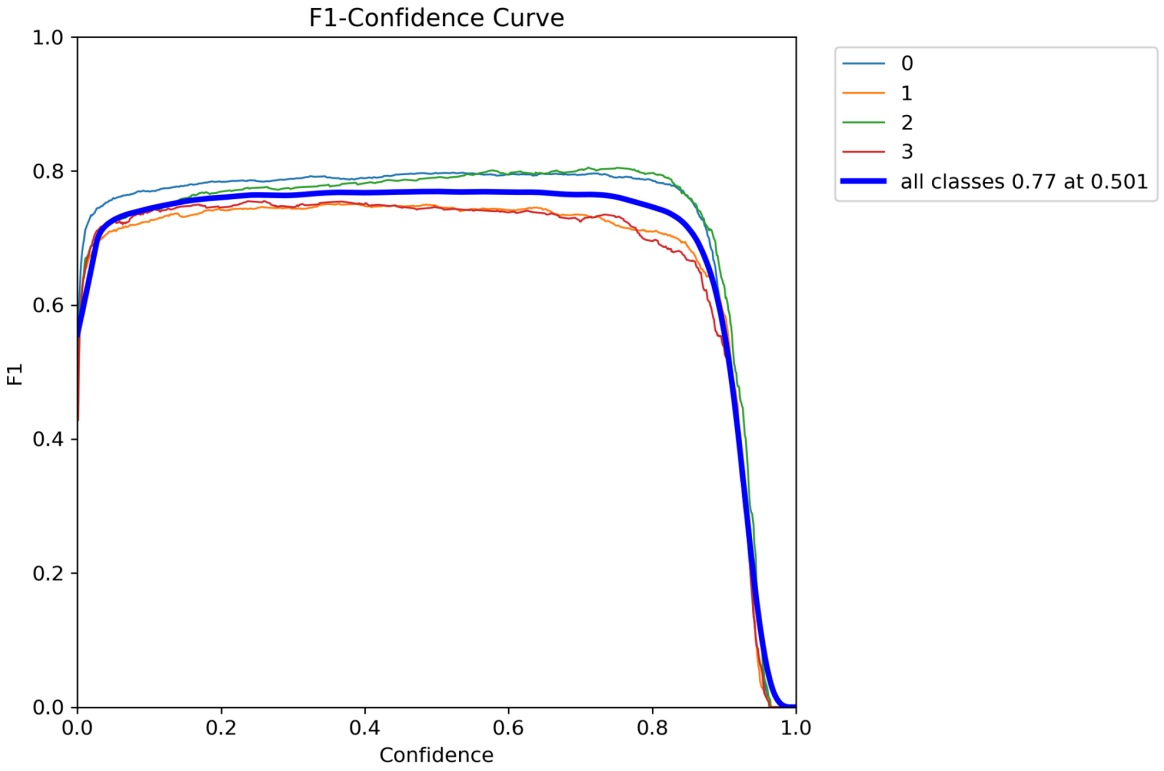


Figure 4.5. F1 - Confidence curve.

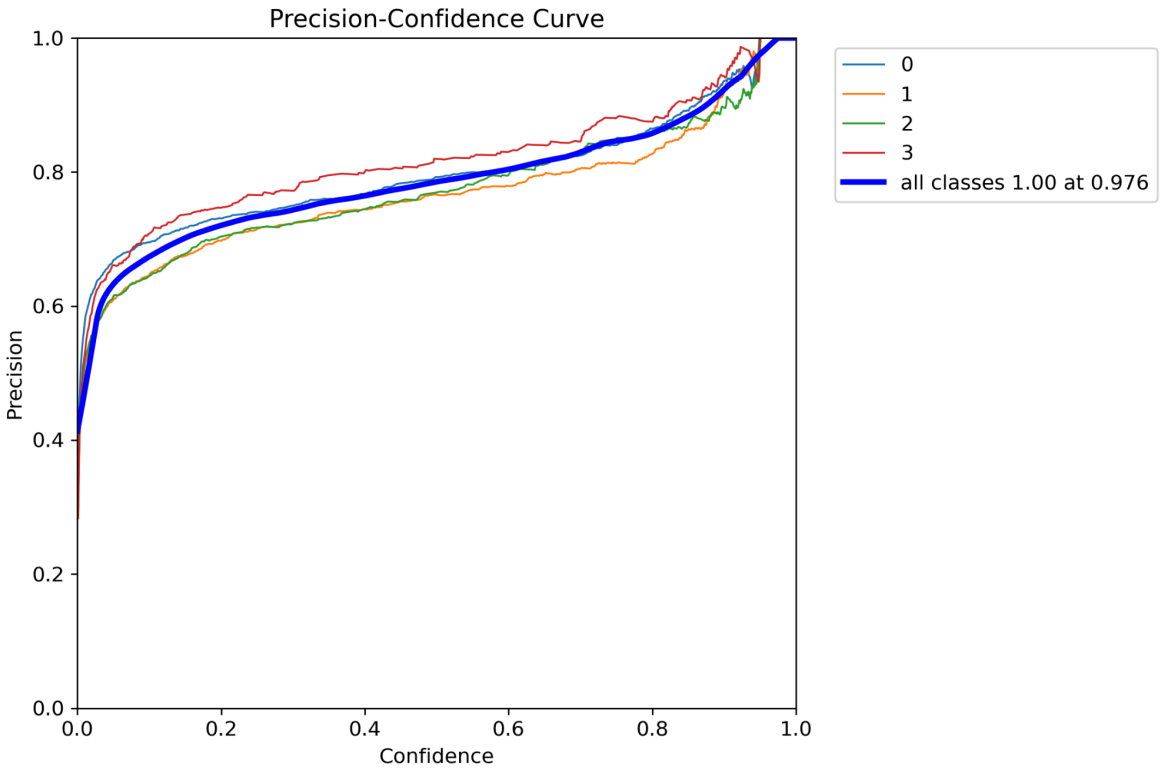


Figure 4.5. Precision - Confidence curve.

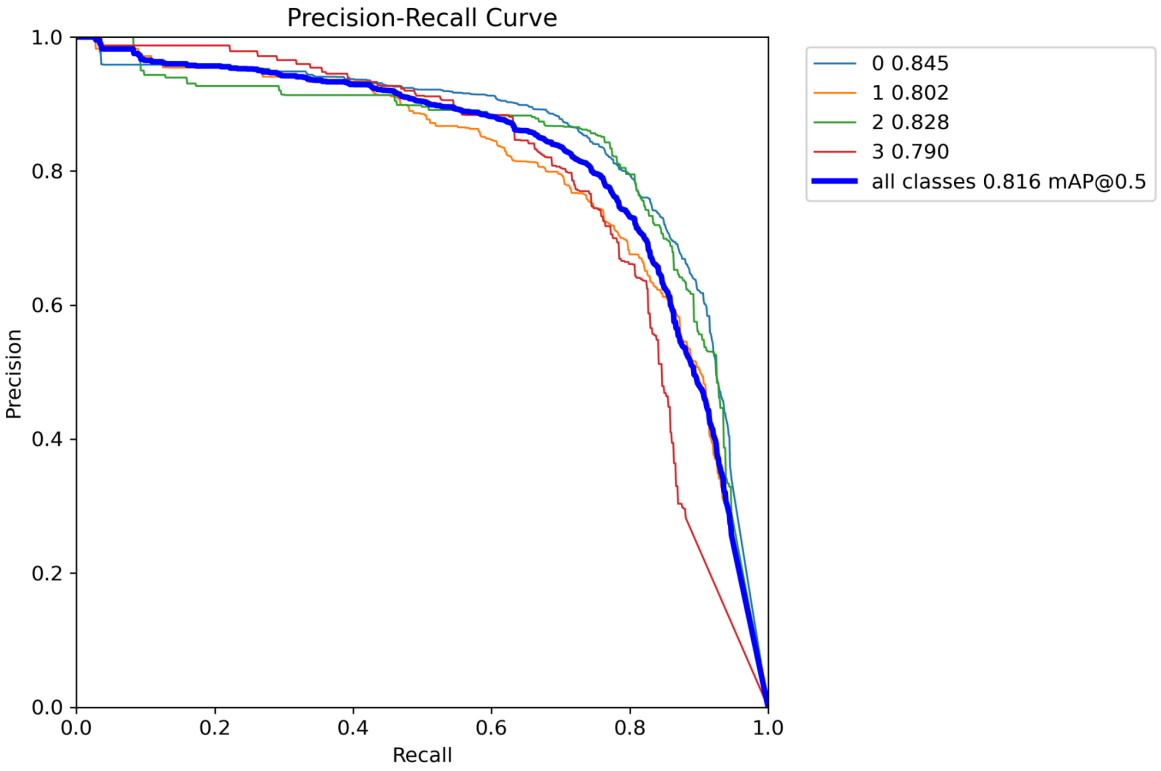


Figure 4.6. Precision - Recall curve.

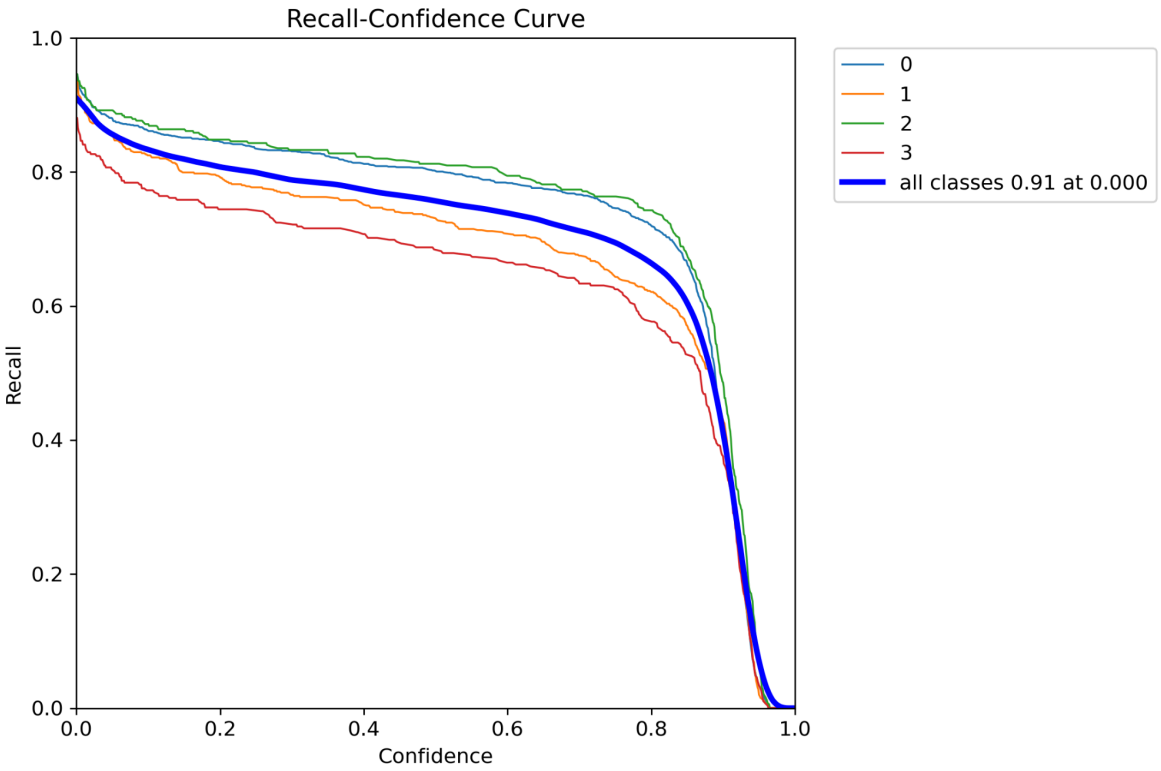


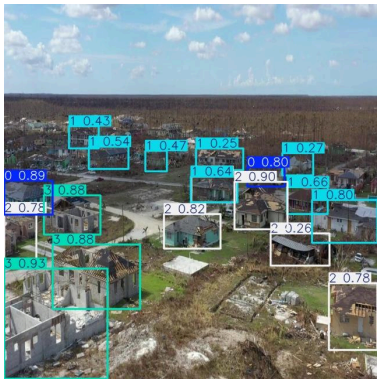
Figure 4.7. Recall - Confidence curve.

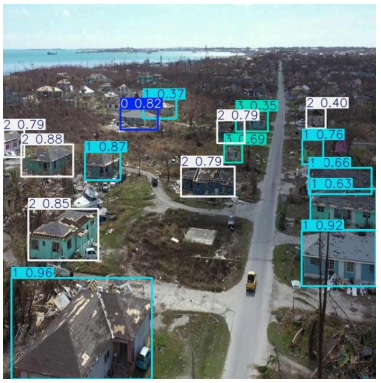
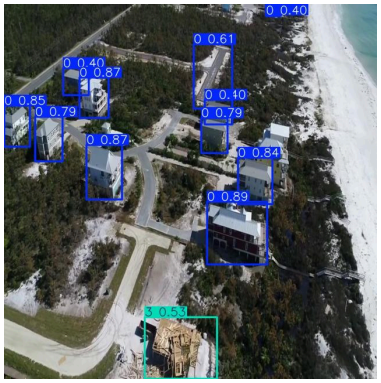
As noted in Figure 4.3, since most performance metrics showed slight improvements utilizing the custom-trained YOLOv9c model, the best.pt weights returned from the aforementioned combined dataset training process were chosen. Overall, inference times were fairly comparable among the various models, with vast differences in resource requirements being observed solely during training. Concurrently, since wanted results were reached with less epochs, alterations in dataset splits and several augmentation options were explored. Namely, geometric transformations (e.g. flipping, rotating), pixel-level transformations (e.g. noise injection, blurring) and cropping and padding techniques, which however did not yield substantially different metrics from the training process using the original unaugmented dataset.

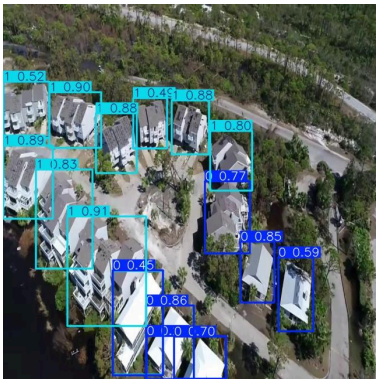
In summary, multiple YOLOv9 models were tested by training them on the combinatory custom dataset created for the purposes of this thesis, with resources pulled from three pre-existing works and adjusted annotations based on the Joint Damage Scale. After varied testing, the top performing training weights were chosen for the BDA inference, stemming from a YOLOv9c model implementation. In Figure 4.8 below, several test set inference results are presented, verifying the accuracy of the proposed model. Overall, this ensures optimal functioning capability for BDA of natural disaster scenarios, similar to the ones featured in the dataset, which is presented further in Chapter 5.

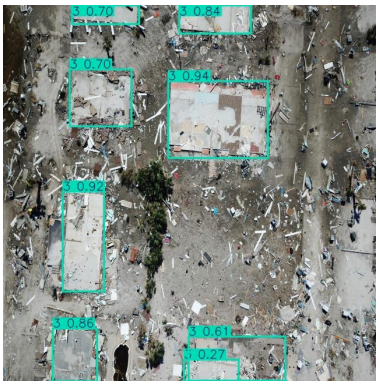
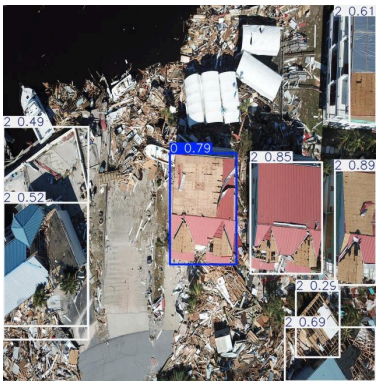
Inputs	Folder of either JPG or PNG aerial images.
Outputs	Folder of either JPG or PNG aerial images, featuring predicted damage class bounding boxes with individual confidence scores.

Table 4.6. Inputs and outputs of aerial imagery process.





Video courtesy Sean Terry, Its About Times
KMBC NEWS



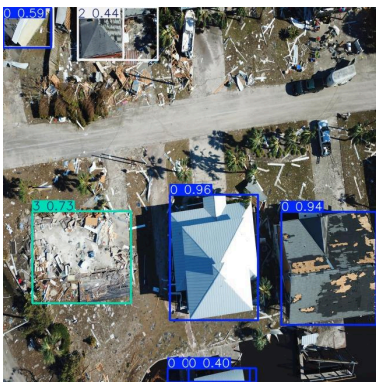



Figure 4.8. Test set inference visual results from the custom dataset, with the best available weights.

Chapter 5 - Results from Examining Individual Cases

5.1 - Satellite Imagery Building Damage Assessment

After building the generalized inference pipeline, results were derived from several real-world events, based upon images sourced from Maxar's Open Data Program, which were then modified accordingly. Both pre and post disaster imagery used for BDA is provided, as well as the damage map generated by the whole process, with individual colors representing joint damage scale values as shown in the corresponding graphic below.

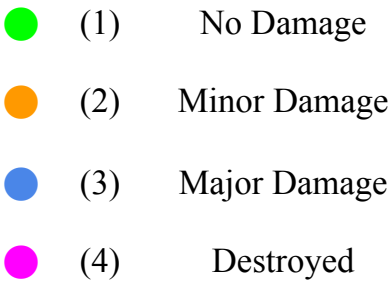


Figure 5.1. Visual representation of damage scale in the implemented pipeline.

The cases examined range from man-made to natural disasters and from minor impact events to major impact ones. Beneath, in Table 5.1 all events analyzed are noted, as well as being highlighted in the map in Figure 5.2.

Event Name	Event Date
Haiti Earthquake	Jan 12, 2010
Beirut, Lebanon Explosion	Aug 4, 2020
Aegean Sea Earthquake	Oct 30, 2020
Bata, Equatorial Guinea Explosions	Mar 7, 2021
Shovi, Georgia Landslide	Aug 3, 2023
Libya Floods	Sept 10-11, 2023

Table 5.1. Events examined by the implemented satellite imagery pipeline, sorted by date.

Imagery was provided in the form of raw data, consisting of multiple large GeoTIFF files per date for each specific event. In this form, the images were unsuitable for BDA in the proposed pipeline, as they either partially covered different areas or had varying resolutions due to different satellite sources. Therefore, as showcased in 4.2, imagery was first refactored using GIS software to fit the aforementioned criteria. The final images were created by combining multiple pre- and post-event GeoTIFFs, primarily focusing on areas of particular interest while excluding land outside urban centers. Ultimately, the end goal is to present several examples that are composed of diverse types of scenarios, both featured on the xBD dataset, along with some that were omitted, demonstrating the capabilities of the proposal in the process.

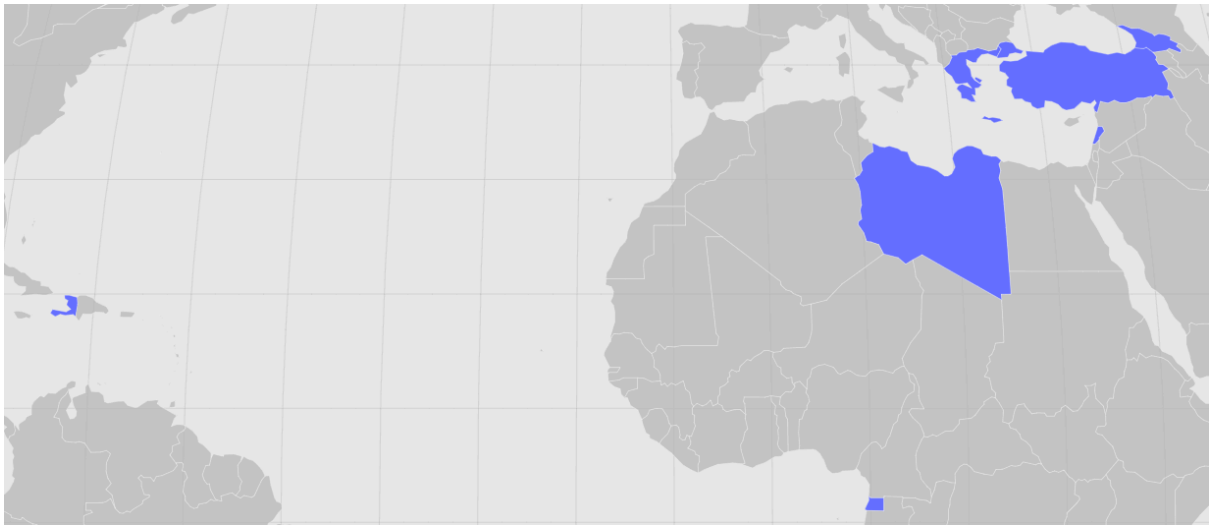


Figure 5.2. Map highlighting countries featuring events examined by the proposed pipeline, using satellite imagery.

The following test cases showcase the expected strength of the proposed process, while also revealing some potential limitations. Inference speed per kilometer of input images is higher than relevant implementations, both for scenarios similar to the xBD dataset, as well as additional ones. Concurrently, predictions for separate building parts, instead of a unified label for each building, provides the opportunity for a more nuanced perception of the situation at hand, something that proved particularly useful for the two explosions covered in this dissertation. Further testing, however, also revealed some of the proposal's weaknesses. In particular, the noted statistics were observed to be skewed in favor of some degree of damage instead of no damage. This can mostly be attributed to the overlapping predicted polygons, which are returned in the CSV files as one instance, rather than multiple. While this occurs for all classes, it is more prevalent in the most common class, which is by far the 'No damage' one. Overall, this entails that undamaged cases are significantly underrepresented in the following provided statistics, though it is not an issue of concern, considering that damaged buildings are of higher importance. Finally, the model seems to produce false positive examples that

nonetheless are widespread in available implementations and can be primarily credited to satellite image issues, such as differences in cloud cover, angle of capture and geometric distortions, among others.

2010 Haiti Earthquake

On Tuesday, 12 January 2010 an earthquake of magnitude 7.0 Mw struck Haiti, with its epicenter being near the town of Léogâne, located about 25 km west of the nation's capital, Port-au-Prince [35]. Estimated casualties were in the range of 160,000 lives, while due to substantial damage to infrastructure, from both the initial earthquake, as well as several aftershocks in the following days, hundreds of thousands of people were displaced. In total, as many as 3 million people were reported to be affected from the disaster, according to the International Federation of Red Cross and Red Crescent Societies. Buildings and infrastructure around the nation were already on a massive strain before the earthquake, exacerbating the destruction even further. Approximately 250,000 residences and 30,000 commercial buildings were either severely damaged or had collapsed, while communication systems, electrical networks and transport facilities, along with hospitals suffered considerable impact as well.

The imagery was sourced from Maxar's WorldView-2 satellite, with a spatial resolution of 0.5m per pixel [15].

Figure 5.3. Port-au-Prince, Haiti before earthquake (09-01-2010) [15].

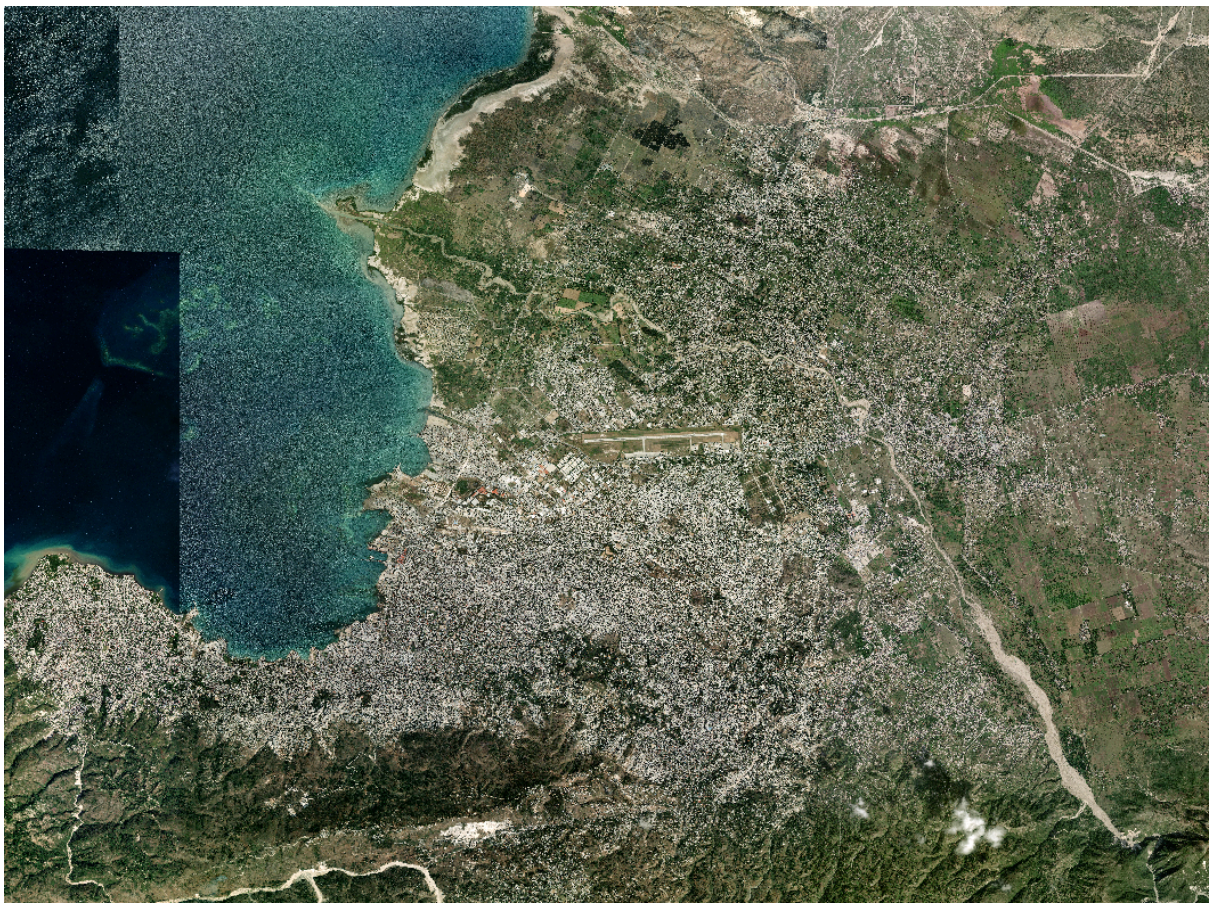


Figure 5.4. Port-au-Prince, Haiti after earthquake (15-01-2010) [15].

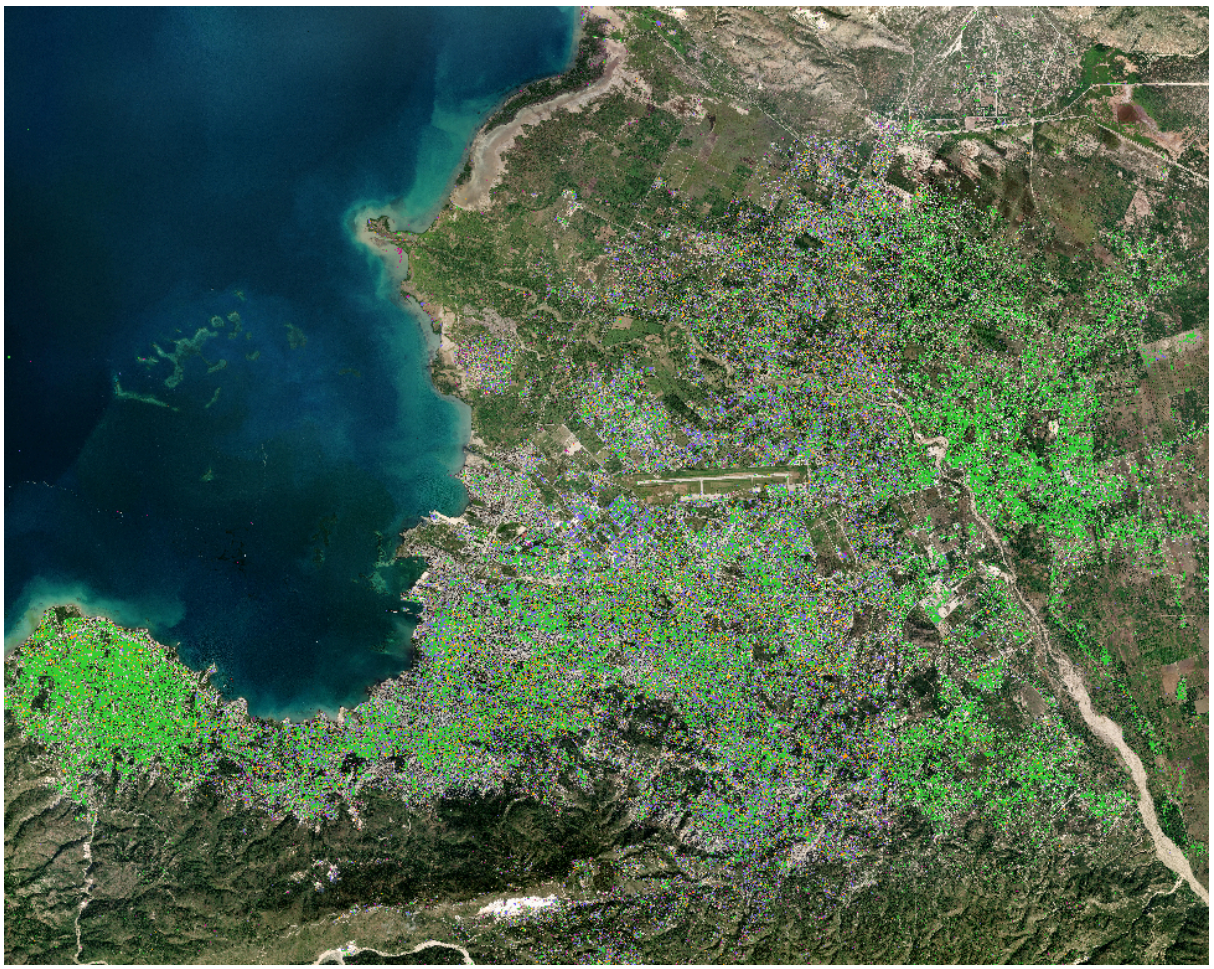


Figure 5.5. Building damage map of Port-au-Prince, Haiti generated by the proposed pipeline.

No Damage	Minor Damage	Major Damage	Destroyed	Overall
12082	3608	6125	3339	25154

Table 5.2. Building parts per damage class of Port-au-Prince, Haiti generated by the proposed pipeline.

In Port-au-Prince, as reported, large parts of the city suffered major damage, as highlighted by the damage map. While Léogâne, which was closer to the epicenter, had the most collapsed buildings, the analysis of Haiti's capital showcases the extent of the disaster, with buildings all around the city being labeled as having either minor or major damage. Taking into account the bias favouring damaged instances, the results appear close to the expected ones.

In total, inference time was approximately 10 hours and 22 minutes, for images that were 53518 pixels wide and 32261 pixels tall. Given a spatial resolution of 0.5 meters per pixel, this corresponds

to a total land area of approximately 431.63 square kilometers. The analysis speed was around 11565.81 square meters per second.

2020 Beirut Explosion

On August 4th 2020, an explosion occurred in the port of the capital of Lebanon, when a fire broke out in a warehouse holding confiscated cargo [36]. The blast was the largest single-fired ammonium nitrate explosion in history, registering as an earthquake of magnitude 3.3 and is considered one of the most powerful non-nuclear explosions ever recorded. The explosion caused 218 deaths, more than 7,000 people being injured, while rendering at least 300,000 homeless and leading to upwards of 15 billion US dollars in property damages.

The imagery was sourced from Maxar's WorldView-3 satellite, with a spatial resolution of 0.31m per pixel [15].

Figure 5.5. Beirut, Lebanon before port explosion (31-07-2020) [15].

Figure 5.6. Beirut, Lebanon after port explosion (05-08-2020) [15].

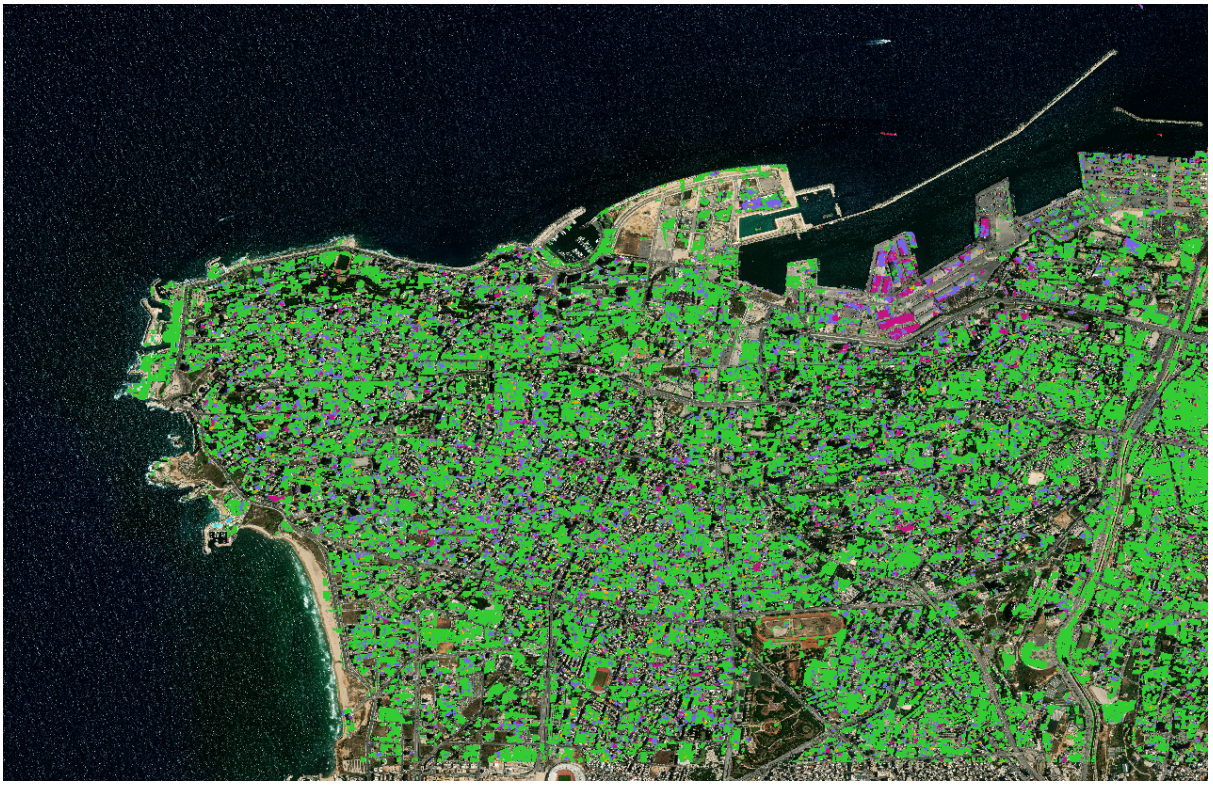


Figure 5.7. Building damage map of Beirut, Lebanon generated by the proposed pipeline. Area of explosion is highlighted in the top right corner.

No Damage	*Minor Damage*	*Major Damage*	*Destroyed*	*Overall*
3850	*344*	*1172*	*644*	*6010*

Table 5.3. Building parts per damage class of Beirut, Lebanon generated by the proposed pipeline.

As evident in the building damage map, all buildings in the port of Beirut surrounding the warehouse where the explosion happened were completely destroyed, while several partially damaged buildings showed up in the surrounding areas.

In total, inference time was approximately 1 hour and 21 minutes, for images that were 28449 pixels wide and 12969 pixels tall. Given a spatial resolution of 0.31 meters per pixel, this corresponds to a total land area of approximately 35.45 square kilometers. The analysis speed was around 7295.59 square meters per second.

2020 Aegean Sea Earthquake

On October 30th 2020, an earthquake of magnitude 7.0 was recorded roughly 14 km northeast of the Greek island of Samos [37], [38]. While the island was closer to the earthquake's epicenter, the region of İzmir Province, Turkey was most heavily affected. In İzmir in particular, which is located 70 km from the epicenter, 117 lives were lost and 1,034 people were injured, while more than 700 buildings were majorly damaged, or destroyed [37]. On the Greek side the impact was less severe, 2 people died, while an additional 19 sustained injuries. Buildings, meanwhile, suffered mostly minor damage that was for the most part completely repairable and no further relief response was required [38].

The imagery was sourced from Maxar's WorldView-3 satellite, with a spatial resolution of 0.31m per pixel [15].

Figure 5.8. İzmir Province, Turkey before the earthquake (27-04-2020) [15].

Figure 5.9. İzmir Province, Turkey after the earthquake (03-11-2020) [15].

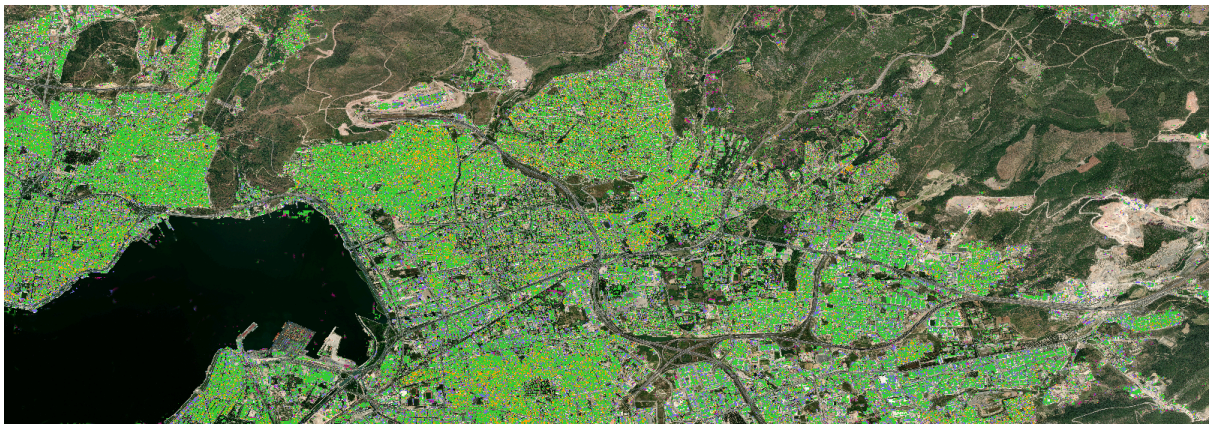


Figure 5.10. Building damage map of İzmir Province, Turkey generated by the proposed pipeline.

No Damage	Minor Damage	Major Damage	Destroyed	Overall
9120	2379	4785	3390	19674

Table 5.4. Building parts per damage class of İzmir Province, Turkey generated by the proposed pipeline.

This overview of the metro area of İzmir, Turkey highlights mostly buildings that showcase minor and major damage. Though from a larger sample size than Samos, Greece, a greater percentage of buildings appear to feature some form of damage, as expected from reports of the event.

In total, inference time was approximately 1 hour and 51 minutes, for images that were 28449 pixels wide and 12969 pixels tall. Given a spatial resolution of 0.31 meters per pixel, this corresponds to a

total land area of approximately 35.45 square kilometers. The analysis speed was around 5323.81 square meters per second.

Figure 5.11. Samos, Greece before the earthquake (31-07-2020) [15].

Figure 5.12. Samos, Greece after the earthquake (05-11-2020) [15].

Figure 5.13. Building damage map of Samos, Greece generated by the proposed pipeline.

No Damage	Minor Damage	Major Damage	Destroyed	Overall
1974	335	1344	426	4079

Table 5.5. Building parts per damage class of Samos, Greece generated by the proposed pipeline.

As mentioned above, on the Greek side most buildings showed minor damage, either shear cracks or generally damage features that are not discernable from high-resolution satellite imagery. While several building examples are labeled as damaged, none of them were destroyed, with most of the instances of destroyed buildings attributed to false positives in non-urban areas.

In total, inference time was approximately 4 hours and 10 minutes, for images that were 27548 pixels wide and 23552 pixels tall. Given a spatial resolution of 0.31 meters per pixel, this corresponds to a total land area of approximately 62.35 square kilometers. The analysis speed was around 4156.71 square meters per second.

2021 Bata Explosions

In the port city of Bata in Equatorial Guinea, on March 7th 2021, 4 successive explosions occurred at some military barracks, supposedly from negligently stored explosives [39]. The explosions led to 108 casualties, while at least 600 more people were injured. Infrastructure around the city suffered greatly, with many people being displaced, leading to about 150 families staying in temporary shelters in Bata and others moving in with relatives. It was reported that almost all buildings and residences in the city showed signs of major damage, with 243 structures characterized as either “heavily damaged or completely destroyed”, according to the United Nations Institute for Training and Research.

The imagery was sourced from Maxar’s WorldView-3 satellite, with a spatial resolution of 0.31m per pixel [15].

Figure 5.14. Bata, Equatorial Guinea before barracks explosions (07-08-2020) [15].

Figure 5.15. Bata, Equatorial Guinea after barracks explosions (09-03-2021) [15].

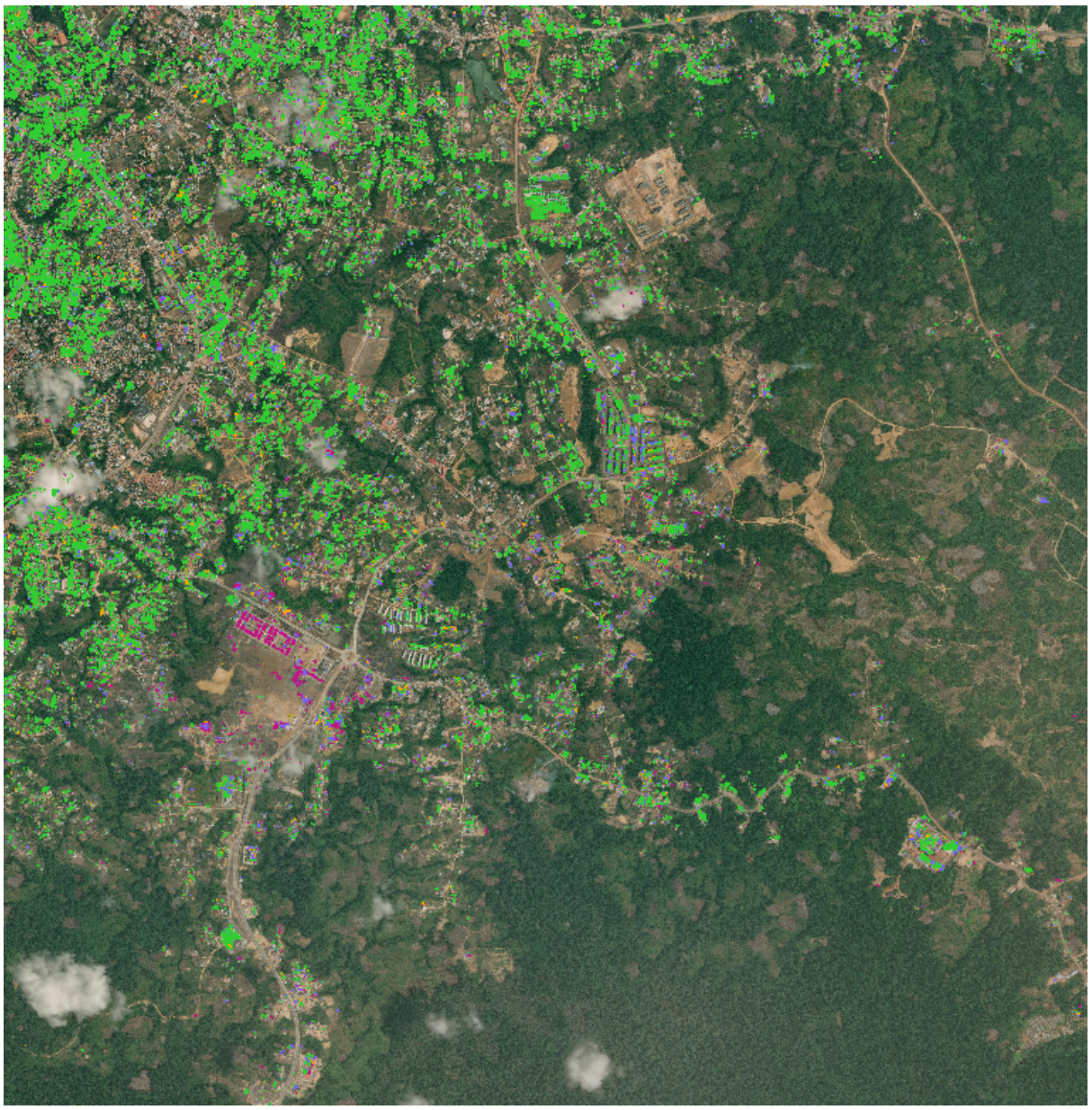


Figure 5.16. Building damage map of Bata, Equatorial Guinea generated by the proposed pipeline. Area of explosion is highlighted in the bottom left corner.

No Damage	Minor Damage	Major Damage	Destroyed	Overall
632	78	459	351	1520

Table 5.6. Building parts per damage class of Bata, Equatorial Guinea generated by the proposed pipeline.

The produced building damage map confirms the complete destruction of the military barracks, as well as the structures surrounding them. Furthermore, various cases of majorly damaged buildings showed up which is to be expected, based on the reports following the explosions.

In total, inference time was approximately 32 minutes for images measuring 14,468 pixels wide by 12,704 pixels tall. Given a spatial resolution of 0.31 meters per pixel, this corresponds to a total land area of approximately 17.66 square kilometers. The analysis speed was around 9,199.65 square meters per second.

2023 Shovi Landslide

On August 3rd 2023, a landslide happened in the mountain resort village of Shovi, Georgia, located at the southern foothills of the Greater Caucasus Mountains [40]. The disaster was attributed to subglacial waters being released after an initial rock mass collapse. The disaster led to 32 people losing their lives, while a large part of the resort's infrastructure was destroyed.

The imagery was sourced from Maxar's WorldView-3 satellite, with a spatial resolution of 0.31m per pixel.

Figure 5.17. Shovi, Georgia before landslide (27-07-2017) [15]. Area of the landslide is highlighted in the right side.

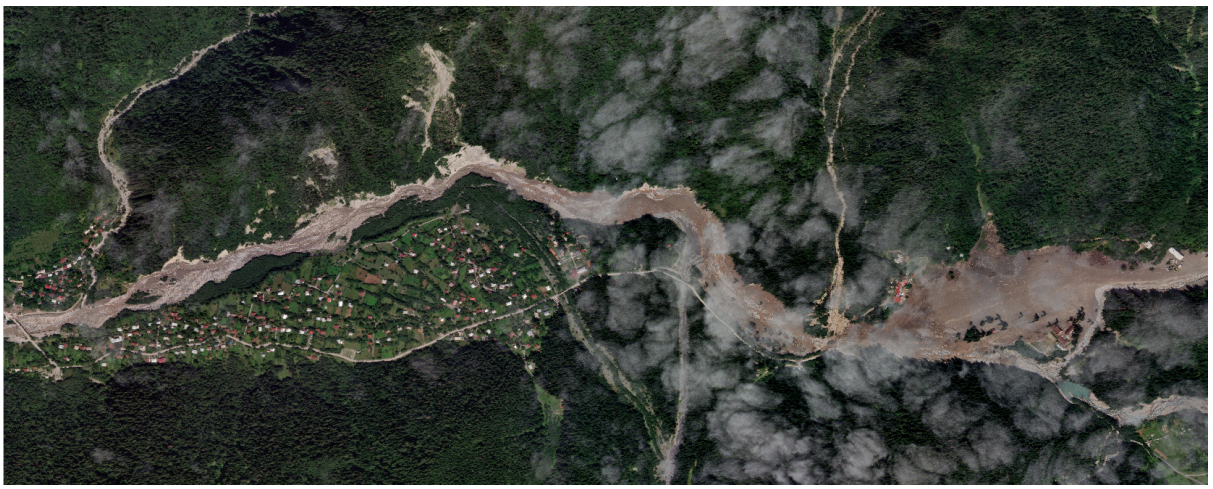


Figure 5.18. Shovi, Georgia after landslide (08-08-2023) [15].

Figure 5.19. Building damage map of Shovi, Georgia generated by the proposed pipeline. Area of the landslide is highlighted on the right side.

No Damage	Minor Damage	Major Damage	Destroyed	Overall
632	78	459	351	1520

Table 5.7. Building parts per damage class of Shovi, Georgia generated by the proposed pipeline.

In accordance with the reports, almost all buildings on the east side of the analyzed area were destroyed, while several others showed major damage. Some false positives showed up in the village on the west side, mainly attributed to cloud coverage and land use changes, along with some older buildings.

In total, inference time was approximately 36 minutes for images measuring 28,170 pixels wide by 7,698 pixels tall. Given a spatial resolution of 0.31 meters per pixel, this corresponds to a total land

area of approximately 20.84 square kilometers. The analysis speed was around 9,647.94 square meters per second.

2023 Libya Floods

In early September of 2023, Storm Daniel affected Greece, Bulgaria and Turkey with extensive flooding, before moving further south to the coast of Libya [41]. Libya, already lacking in critical infrastructure due to the recent civil war, suffered the most with more than five thousand casualties and tens of thousands of people recorded missing. One of the places hit hardest by the tropical storm was the city of Derna and its surrounding areas, with the collapsing of dams (Derna dam and Mansour dam) on the outskirts of the city leading to the release of an estimated 30 million cubic meters of water. Overall, it led to at least 40,000 residents being displaced, more than 2,200 buildings flooded and substantial parts of the city dragged out to the Mediterranean Sea, leading to about a quarter of it being destroyed.

The imagery was sourced from Maxar's WorldView-3 satellite, with a spatial resolution of 0.31m per pixel [15].

Figure 5.20. Derna, Libya before Storm Daniel (01-07-2023) [15].

Figure 5.21. Derna, Libya after Storm Daniel (13-09-2023) [15].

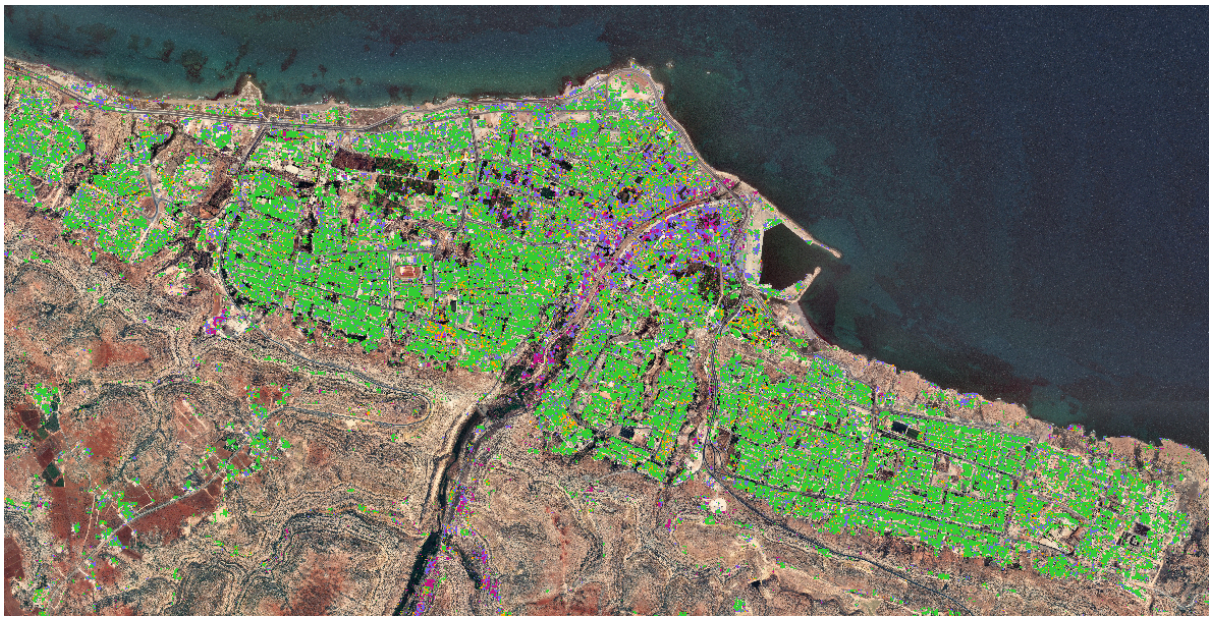


Figure 5.22. Building damage map of Derna, Libya generated by the proposed pipeline. Area primarily affected by flooding is highlighted on the river running through the center of the city.

No Damage	Minor Damage	Major Damage	Destroyed	Overall
8442	1998	3699	2502	16641

Table 5.8. Building parts per damage class of Derna, Libya generated by the proposed pipeline.

As shown in the building damage map, urban areas on both sides of the waterway past the collapsed Mansour Dam were primarily affected, with buildings labeled as having major damage or being destroyed, as well as many buildings in surrounding areas showing water damage.

In total, inference time was approximately 11 hours and 8 minutes for an image measuring 48,932 pixels wide by 38,404 pixels tall. Given a spatial resolution of 0.31 meters per pixel, this corresponds to a total land area of approximately 182.04 square kilometers. The analysis speed was around 4,505.73 square meters per second.

5.2 - Aerial Footage Building Damage Assessment

After training the YOLOv9c model, several images were sourced from natural disaster coverage, for the purpose of testing the process's efficacy in examples outside the datasets utilized. As noted in Chapter 3, the most prominent available datasets, namely ISBDA and DoriaNET, feature predominantly suburban single-family homes, sourced from natural disasters in the continental United States. Therefore, most examples used for testing feature similar building types, since as also noted below the few examples outside said structures analyzed, yielded subpar results. In general, along with several examples from hurricane induced damage in suburban homes in the US, a local natural disaster, the 2022 floods in Thessaly, Greece were covered.

Bounding Box Class Label	Building Damage Level
0	No Damage
1	Minor Damage
2	Major Damage
3	Destroyed

Table 5.9. Bounding box class labels and building damage level correlation legend.

Event Name	Event Date
Range of Hurricanes in the United States of America	2017-2023
Greece Floods	Sept 5, 2023

Table 5.10. Events examined by the implemented aerial footage pipeline, sorted by date.

Range of Hurricanes in the United States of America

As highlighted in the former section, the vast majority of available training data covered areas of the United States affected by hurricanes. Therefore, images were obtained from multiple online sources, covering a wide array of post-hurricane scenarios, for proving the efficacy of the proposed method. The United States is located in an area particularly prone to hurricanes, which, along with the widespread availability of aerial footage, provides a basis for a greater sample size of such footage. From a larger set of online-sourced images, selected examples are presented in the following pages to demonstrate the effectiveness of the proposed pipeline.

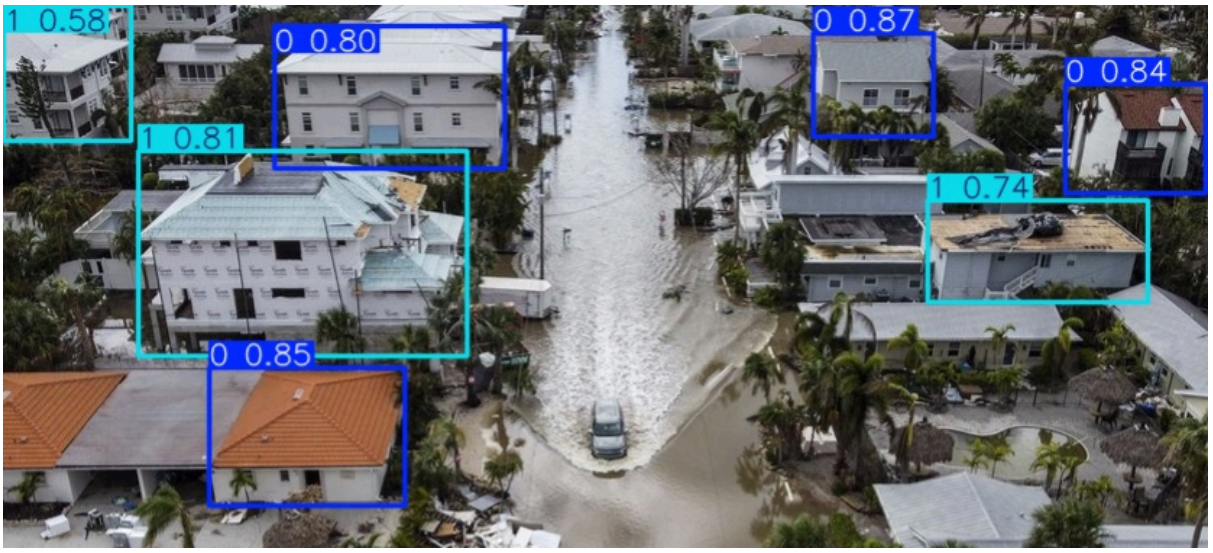


Figure 5.23.a. Building damage bounding boxes on imagery from the United States by the proposed pipeline.

Figure 5.23.b. Building damage bounding boxes on imagery from the United States by the proposed pipeline.

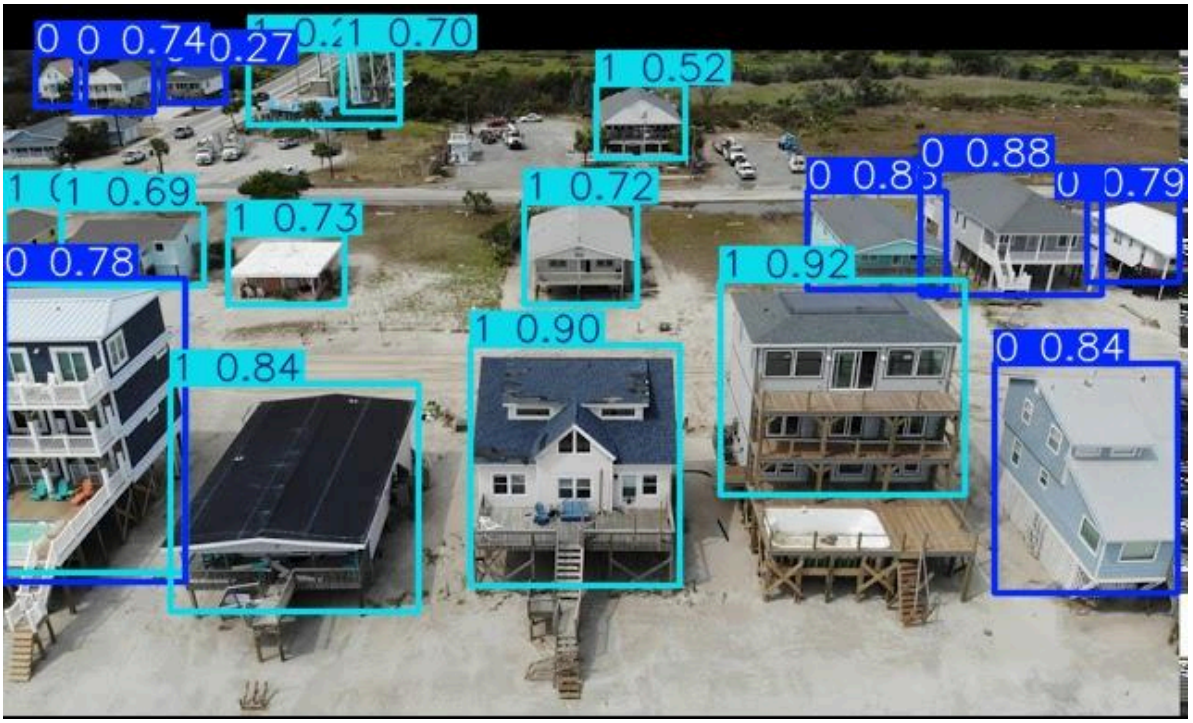


Figure 5.23.c. Building damage bounding boxes on imagery from the United States by the proposed pipeline.

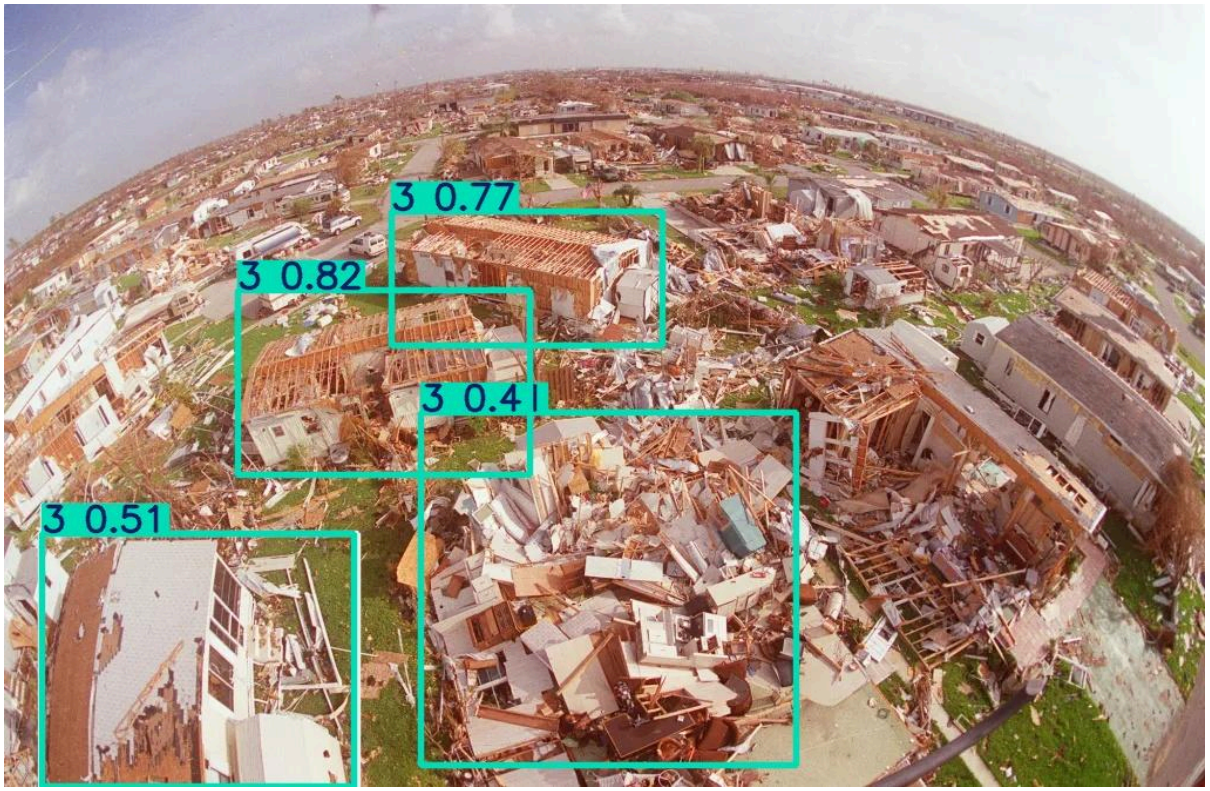


Figure 5.23.d. Building damage bounding boxes on imagery from the United States by the proposed pipeline.

Figure 5.23.e. Building damage bounding boxes on imagery from the United States by the proposed pipeline.

Figure 5.23.f. Building damage bounding boxes on imagery from the United States by the proposed pipeline.

Overall, the model defines building bounding boxes effectively, although confidence scores regarding damage classes seem to vary. From these scenarios, the highest confidence scores seem to be returned from minor damage cases, followed by undamaged buildings and then destroyed and majorly damaged ones. This confirms the expected outcome, since more undamaged, along with intermediate damage examples were featured compared to xBD, so the model performs optimally on said scenarios.

2023 Greece Floods

On September 5th 2023, Storm Daniel led to mass flooding in the region of Thessaly in central Greece, leading to 17 deaths and more than 2 billion dollars worth of damage [42]. Data from the Sentinel-1 satellite following the Storm estimated that an area around 180,000 acres had been flooded. Several hundred people were displaced and a substantial part of the agricultural production was hindered, in large parts inflicting long-term damage. Similarly to the previous subsection, selected

examples are presented in the following pages to demonstrate the effectiveness of the proposed pipeline.

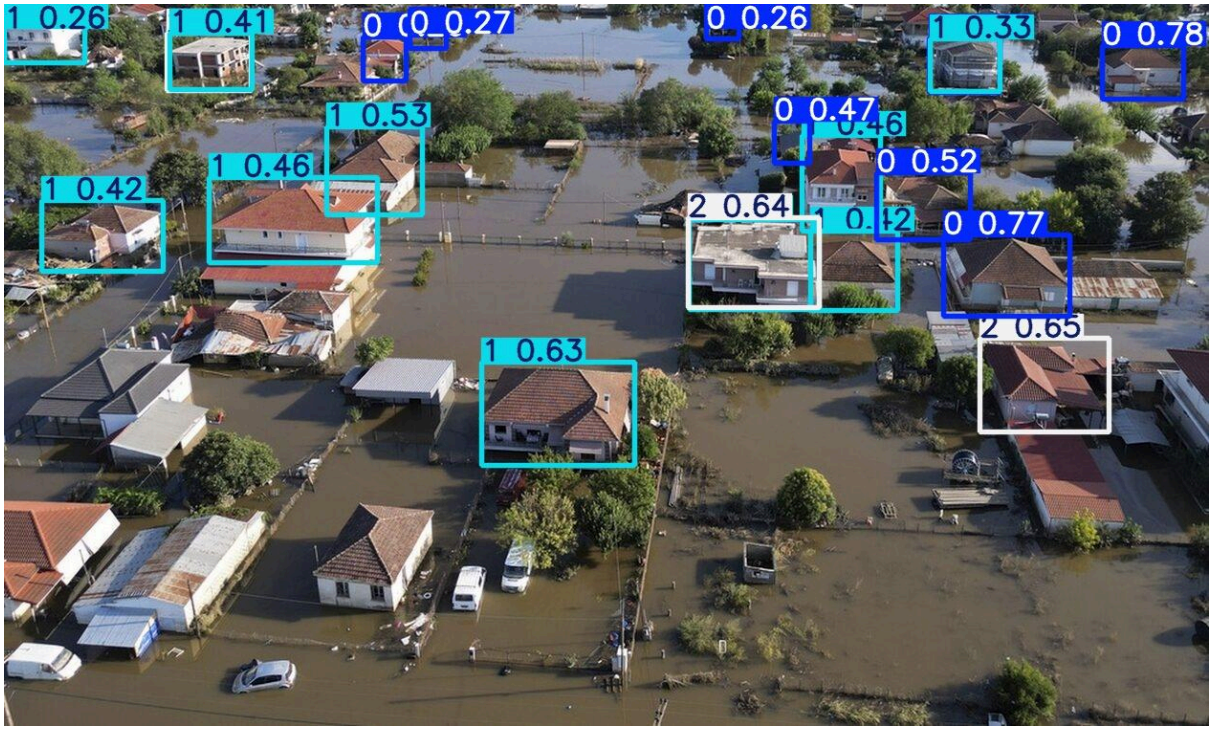


Figure 5.24.a. Building damage bounding boxes on imagery from Thessaly, Greece by the proposed pipeline.

Figure 5.24.b. Building damage bounding boxes on imagery from Thessaly, Greece by the proposed pipeline.

Figure 5.24.c. Building damage bounding boxes on imagery from Thessaly, Greece by the proposed pipeline.

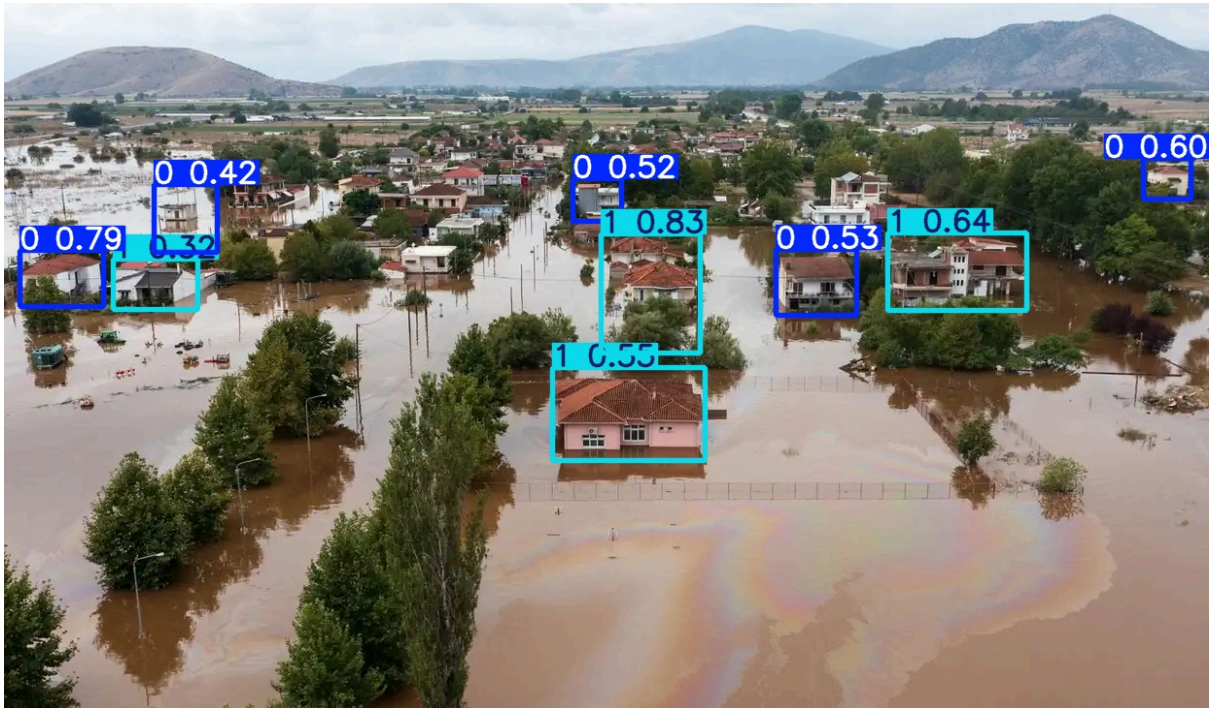


Figure 5.24.d. Building damage bounding boxes on imagery from Thessaly, Greece by the proposed pipeline.

Results observed from analyzing these images returned similar results performance-wise, to the ones pertaining to hurricane structural damage. Since no collapsed buildings are featured, just ones featuring varying degrees of water damage, the model predicted the aforementioned damage levels fairly precisely. As with the hurricane scenarios, the model appears to perform better for minor damage cases, compared to more major ones.

Conclusions and Future Work

Natural disaster response is a critical sector that involves rescuing countless lives each year and mitigating the impact of damaged buildings and surrounding infrastructure. Over the past decade, efforts from research groups, as well as larger humanitarian organizations, have provided resources that assist in that response in a remarkable way. In the same manner, advancements in the field of AI allow for the possibility of developing tools that automate and further speed up the whole process, with CNN and Transformer based models held in particularly high regard.

Throughout this dissertation, all present forms of remote-sensing Building Damage Assessment, utilizing Machine Learning techniques were examined, along with their applicability in a unified pipeline. The main goal of this thesis was to provide a substantial analysis on available BDA with ML methods, in the interest of providing tools necessary for responding to future disasters more effectively. Firstly, using satellite imagery provides a comprehensive overview of the conditions, offering analysis-ready data that can be delivered with increasing speed. With progressively more efficient methods for automating the analysis of this data, thousands of square kilometers can be assessed more accurately and quickly than any manual approach. Following this streamlined process, instead of deploying coordinated teams for further assessment, a subsequent automated procedure utilizing aerial footage is carried out. With the collaboration of the two aforementioned methods, humanitarian response teams have ample information pertaining to the situation at hand, therefore enabling them to address it accordingly. By leveraging advancements in CNNs for pixel-level image segmentation and classification, alongside state-of-the-art object detection models like the YOLO series, an accurate assessment has been made of what can be achieved for BDA with contemporary resources.

Moreover, this work examined the balance between the potential models' assessment accuracy and the speed at which they return results, to evaluate their applicability in real-world situations. Ultimately, this led to a potential method that automates a significant portion of the process for effectively addressing urgent cases. The toolkit implemented for this thesis is innovative in that it provides a suite of BDA resources, which work in unison, bringing together functionality from several other projects. In total, this end-to-end unified pipeline with a user-friendly interface yields results, from which important information can be derived in a timely manner, directing emergency response groups appropriately. Considering future possibilities, the aggregation of available data and proven assessment methods is expected to strengthen urban resilience against pressing challenges such as climate change. Consequently, a more nuanced analysis of past events enables the opportunity to

better inform pertaining entities, along with the general public, mitigating the socioeconomic impact of future disasters.

On the other hand, since the satellite imagery BDA model was trained on the xBD dataset, generalizing the whole process beyond xBD showed some limitations. As highlighted in 3.1, the damage classes of “minor damage” and “major damage” are underrepresented, a fact which renders it difficult to differentiate between them, as well as discern each individual class. Unless more substantial datasets are created in the future, BDA on even a set damage scale is bound to return results of limited accuracy, even with the aforementioned advancements in transformer and SSM based models. Furthermore, necessary features for determining structural stability of buildings, such as cracks or discoloration, cannot be detected on even the highest possible resolution images, at 30 cm per pixel. This poses challenges for more refined damage classification and confines the potential analysis between fairly strict margins. Finally, optimisation of model performance remains a major hurdle that needs to be overcomed, both in terms of resource requirements and overall inference speed, since the main usage for these types of models is response to humanitarian crises. The whole process in total, from acquiring the raw satellite imagery, to deriving meaningful feedback post-inference, takes an inordinate amount of time. Temporal efficiency, which is crucial in such situations, has to be the primary aspect to improve upon moving forward.

In the context of aerial imagery, limitations related to BDA have also arisen. Similarly to satellite imagery, the extent to which images and videos can be analyzed effectively is bound to the datasets the corresponding models are trained on. The main issue that was noted, when pertaining to the model trained for this thesis, was the limited variation between building types. Since the sample drone footage was exclusively from natural disasters in suburban or rural areas of the United States, the model proved to be potent almost solely when it had to perform BDA on typical American single-family homes. A dataset covering a wider range of disasters among different regions around the world is a necessity for training more competent models in the future. Along the same lines, as cameras and image processing improve, UAVs can be utilized to discern particular damage features in buildings, without demanding extensive recording of each individual facade. This will expand tremendously on the already available tools that are restricted on an arbitrary damage scale, provided that experts build new datasets featuring the necessary generalizability for real world applications.

References

[1] T. Lillesand, R. Kiefer, and J. Chipman, Remote Sensing and Image Interpretation, 5th ed. New York, NY, USA: Wiley, 2004.

[2] Esri, "Fundamentals of Pan Sharpening," ArcGIS Online, https://doc.arcgis.com/en/arcgis-online/analyze/fundamentals-of-pan-sharpening-pro.htm.

[3] Open Geospatial Consortium (OGC), "GeoTIFF Standard," [Online]. Available: https://www.ogc.org/standard/geotiff/.

[4] Soori, M., Arezoo, B., & Dastres, R. "Artificial Intelligence, Machine Learning and Deep Learning in Advanced Robotics: A Review," Cognitive Robotics, vol. 3, pp. 54-70, 2023. doi: 10.1016/j.cogr.2023.04.001.

[5] S. A. Sayed, Y. Abdel-Hamid, and H. A. Hefny, "Artificial intelligence-based traffic flow prediction: a comprehensive review," Journal of Electrical Systems and Information Technology, vol. 10, no. 13, 2023. doi: 10.1186/s43067-023-00081-6.

[6] A. Yamashita, "A Comprehensive Guide to Convolutional Neural Networks — The ELI5 Way," Towards Data Science, Dec. 15, 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[7] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," arXiv preprint arXiv:1511.08458, 2015. [Online]. Available: https://arxiv.org/abs/1511.08458.

[8] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Llados, and U. Pal, "SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification," arXiv, 2017. [Online]. Available: https://arxiv.org/abs/1707.02131.

[9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Upper Saddle River, NJ: Pearson, 2018.

[10] V. S. Shaik Althaf, S. Wang, G. Zhai, and B. Spencer, "CNN based data anomaly detection using multi-channel imagery for structural health monitoring," *Smart Structures and Systems*, vol. 29, no. 1, pp. 181-193, Jan. 2022. doi: 10.12989/sss.2022.29.1.181.

[11] M. Vakili, M. Ghamsari, and M. Rezaei, "Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification," arXiv preprint arXiv:2001.09636, 2020. [Online]. Available: https://arxiv.org/abs/2001.09636.

[12] R. Padilla, S. Netto, and E. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," in *Proc. IEEE Int. Workshop on Signal Processing and Applications (IWSSIP)*, 2020, doi: 10.1109/IWSSIP48289.2020.

[13] A. Rosebrock, "Intersection Over Union (IoU) for Object Detection," PyImageSearch, Nov. 7, 2016. [Online]. Available: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

[14] R. Gupta, R. Hosfelt, S. Sajeev, N. Patel, B. Goodman, J. Doshi, E. Heim, H. Choset, and M. Gaston, "xBD: A dataset for assessing building damage from satellite imagery," arXiv, Nov. 2019. [Online]. Available: https://arxiv.org/abs/1911.09296.

[15] Maxar Technologies, "Maxar Open Data," [Online]. Available: https://www.maxar.com/open-data.

[16] X. Zhu, J. Liang, and A. Hauptmann, "MSNet: A multilevel instance segmentation network for natural disaster damage assessment in aerial videos," arXiv, Jun. 2020. [Online]. Available: https://arxiv.org/abs/2006.16479.

[17] M. Rahnemoonfar, T. Chowdhury, and R. Murphy, "RescueNet: A High Resolution UAV Semantic Segmentation Dataset for Natural Disaster Damage Assessment," Scientific Data, vol. 10, no. 1, Dec. 2023, doi: 10.1038/s41597-023-02799-4.

[18] H. Chen, J. Song, C. Han, J. Xia, and N. Yokoya, "ChangeMamba: Remote sensing change detection with spatiotemporal state space model," IEEE Trans. Geosci. Remote Sens., vol. 62, pp. 1-20, 2024, doi: 10.1109/TGRS.2024.3417253.

[19] M. Tani, M. Terao, N. Sogi, T. Shibata, K. Senzaki, and R. Rodrigues, "Disaster Damage Assessment Using LLMs and Image Analysis," *Special Issue on Revolutionizing Business Practices with Generative AI*, *NEC Technical Journal*, vol. 17, no. 2.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," arXiv, May 2015. [Online]. Available: https://arxiv.org/abs/1505.04597.

[21] P. Kramarczyk and B. Hejmanowska, "UNET NEURAL NETWORK IN AGRICULTURAL LAND COVER CLASSIFICATION USING SENTINEL-2," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLVIII-1/W3-2023, pp. 85–90, 2023. doi: 10.5194/isprs-archives-XLVIII-1-W3-2023-85-2023.

[22] E. Schonfeld, B. Schiele, and A. Khoreva, "A U-Net Based Discriminator for Generative Adversarial Networks," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[23] V. Růžička, S. D'Aronco, J. D. Wegner, and K. Schindler, "Deep active learning in remote sensing for data efficient change detection," arXiv, Aug. 2020. [Online]. Available: https://arxiv.org/abs/2008.11201.

[24] R. Varghese and S. M., "YOLOv8: A novel object detection algorithm with enhanced performance and robustness," in Proc. 2024 Int. Conf. Advances Data Eng. Intelligent Comput. Syst. (ADICS), Chennai, India, 2024, pp. 1-6, doi: 10.1109/ADICS58448.2024.10533619.

[25] K. Stavrakakis, D. Pacios, N. Papoutsakis, N. Schetakis, P. Bonfini, T. Papakosmas, B. Charalampopoulou, J. L. Vázquez-Poletti, and A. Di Iorio, "EYE-Sense: empowering remote sensing with machine learning for socio-economic analysis," in *Ninth International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2023)*, vol. 12786, K. Themistocleous, D. G. Hadjimitsis, S. Michaelides, and G. Papadavid, Eds., SPIE, 2023, pp. 127860D. [Online]. Available: https://doi.org/10.1117/12.2681739.

[26] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," arXiv preprint arXiv:2402.13616, 2024. [Online]. Available: https://arxiv.org/abs/2402.13616.

[27] OpenMMLab, "MMYOLO: A unified framework for object detection," GitHub repository, https://github.com/open-mmlab/mmyolo/tree/main/configs/yolov8.

[28] S. Gholami et al., "On the Deployment of Post-Disaster Building Damage Assessment Tools using Satellite Imagery: A Deep Learning Approach," 2022 IEEE International Conference on Data Mining Workshops (ICDMW), Orlando, FL, USA, 2022, pp. 1029-1036, doi: 10.1109/ICDMW58026.2022.00134.

[29] H. Hao, J. Zhang, W. Li, and X. Liu, "An attention-based system for damage assessment using satellite imagery," in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Brussels, Belgium, 2021, pp. 4396-4399, doi: 10.1109/IGARSS47720.2021.9554054.

[30] 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 Sens. Environ., vol. 265, p. 112636, 2021, doi: 10.1016/j.rse.2021.112636.

[31] H. Chen, E. Nemni, S. Vallecorsa, X. Li, C. Wu, and L. Bromley, "Dual-Tasks Siamese Transformer Framework for Building Damage Assessment," arXiv preprint arXiv:2201.10953, 2022. [Online]. Available: https://arxiv.org/abs/2201.10953.

[32] Microsoft, "Building Damage Assessment CNN Siamese," GitHub, May 26, 2023. [Online]. Available: https://github.com/microsoft/building-damage-assessment-cnn-siamese.

[33] C.-S. Cheng, A. Behzadan, and A. Noshadravan, "DoriaNET: A visual dataset from Hurricane Dorian for post-disaster building damage assessment," Designsafe-CI, 2023. [Online]. Available: https://www.designsafe-ci.org/data/browser/public/designsafe.storage.published/PRJ-3278v2. doi: 10.17603/DS2-GQVG-QX37.

[34] Ultralytics, "YOLOv9 Benchmarks," *Ultralytics YOLO Docs*. [Online]. Available: https://docs.ultralytics.com/models/yolov9/#yolov9-benchmarks.

[35] R. Desroches, M. Comerio, M. Eberhard, W. Mooney, and G. Rix, "Overview of the 2010 Haiti earthquake," Earthquake Spectra, vol. 27, pp. S1–S21, 2011, doi: 10.1193/1.3630129.

[36] S. Sivaraman and S. Varadharajan, "Investigative consequence analysis: A case study research of Beirut explosion accident," J. Loss Prev. Process Ind., vol. 69, p. 104387, 2021, doi: 10.1016/j.jlp.2020.104387.

[37] H. E. Demirci, M. Karaman, and S. Bhattacharya, "A survey of damage observed in Izmir due to 2020 Samos-Izmir earthquake," Nat. Hazards, vol. 111, pp. 1047–1064, 2022, doi: 10.1007/s11069-021-05085-x.

[38] I. Kalogeras, N. S. Melis, and N. Kalligeris, "The earthquake of October 30th, 2020 at Samos, Eastern Aegean Sea, Greece," Prelim. Rep. v2., Nat. Obs. Athens, Inst. Geodyn., Greece, 2020.

[39] Wikipedia contributors, "2021 Bata explosions," Wikipedia, The Free Encyclopedia, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=2021_Bata_explosions&oldid=1220168703.

[40] Wikipedia contributors, "2023 Shovi landslide," Wikipedia, The Free Encyclopedia, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=2023_Shovi_landslide&oldid=1238741497.

[41] J. P. Rafferty, "Libya flooding of 2023," Encyclopedia Britannica, Sep. 3, 2024. [Online]. Available: https://www.britannica.com/event/Libya-flooding-of-2023.

[42] "Storm Daniel," *Wikipedia*, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Storm_Daniel.