

Project Report on

**“RAPID AUTOMATED BUILDING DAMAGE
ASSESSMENT IN POST-NATURAL DISASTER
SCENARIOS USING CONVOLUTIONAL NEURAL
NETWORKS”**

Submitted for the Partial Fulfilment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN CIVIL ENGINEERING

Submitted by

ADITYA MHATRE (201010006)

AKSHANSH DIXIT (201010033)

AMAN SINGH (201010030)

AYUSH BAGUL (201010023)

Under the guidance of,

Dr. VIKAS B. VAREKAR



Civil and Environmental Engineering Department
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE

HR Mahajani Road, Matunga (E),

Mumbai-400019

JUNE 2024

DECLARATION

We declare that this written submission for project report phase one on **“Rapid Automated Building Damage Assessment in Post-Natural Disaster Scenarios Using Convolutional Neural Networks”** represents our ideas in our own words. Where others’ ideas or words have been included, we have adequately cited and referenced the sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented fabricated or falsified any idea/data/fact/source in our submission.

We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been taken when needed.

ADITYA MHATRE
(201010006)

AKSHANSH DIXIT
(201010033)

AMAN SINGH
(201010030)

AYUSH BAGUL
(201010023)

CERTIFICATE

This is to certify that Aditya Mhatre (201010006) , Ayush Bagul (201010023) , Akshansh Dixit (201010033) , and Aman Singh (201010030) the students of the Bachelor of Technology of Civil Engineering, Veermata Jijabai Technological Institute, Mumbai have successfully completed the B. Tech Project on “**Rapid Automated Building Damage Assessment in Post-Natural Disaster Scenarios using Computer Vision and Machine Learning in Civil Engineering**”. During the academic year 2023-24, under the guidance of Dr. Vikas B. Varekar to our satisfaction.

Dr. Vikas. B. Varekar

*Project Guide,
Assistant Professor
Civil and Environmental
Engineering Department*

Dr. Nitin Goyal

*Principal Scientist and Chair,
CSIR-National Environment
Engineering Research Institute (NEERI),
Mumbai Zonal Center, Mumbai*

CERTIFICATE

This is to certify that Aditya Mhatre, Ayush Bagul, Akshansh Dixit, and Aman Singh the students of the Bachelor of Technology of Civil Engineering, Veermata Jijabai Technological Institute, Mumbai have successfully completed the B.Tech Project on “**Rapid Automated Building Damage Assessment and Rehabilitation Recommendations in Post-Natural Disaster Scenarios using Computer Vision and Machine Learning in Civil Engineering**”. During the academic year 2023-2024, under the guidance of Dr. Vikas B. Varekar to our satisfaction.

Dr. Vikas B. Varekar

*Assistant Professor,
Civil and Environmental Engineering
Department*

Dr. A.S. Wayal

*Head,
Civil and Environmental Engineering
Department*

Dr. Sachin Kore

*Director,
Veermata Jijabai Technological Institute*

ACKNOWLEDGEMENT

We would like to thank our project guide, **Dr. Vikas B. Varekar** from **Civil and Environmental Engineering Department of Veermata Jijabai Technological Institute (V.J.T.I)** for helping us throughout the course, guiding us and being a constant source of motivation. We would also like to thank the **Head** and all the **teaching** and **non-teaching staff** from **Civil and Environmental Engineering Department of Veermata Jijabai Technological Institute (V.J.T.I)** for their support. It gives us immense pleasure to place this final year project report first phase with his hearty co-operation and we would like to extend our sincere thanks to him.

ADITYA MHATRE

(201010006)

AKSHANSH DIXIT

(201010033)

AMAN SINGH

(201010030)

AYUSH BAGUL

(201010023)

CONTENTS

DECLARATION	ii
CERTIFICATE.....	iii
CERTIFICATE.....	iv
ACKNOWLEDGEMENT.....	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
<u>1. Introduction</u>	1
1.1 Need of the study	4
1.2 Scope of the work	4
1.3 Objectives	5
<u>2. Literature Review</u>	6
<u>3. Nepal Floods 2017: An Overview</u>.....	10
3.1 Overview and Background Information	10
3.2 Locations of the floods	11
3.3 Causes of the floods.....	11
3.4 Types of Impacts	13
3.4.1 Environmental Impacts	13
3.4.2 Economic Impacts	13
3.4.3 Infrastructural Impacts.....	14
3.5 Response and Mitigation:	15
<u>4. Methods and Materials</u>	16
4.1 Technologies Used	16
4.2 List of Libraries	17
4.3 Exploring Neural Networks	18
4.4 Methodology of Assessment	19
4.4.1 Data Collection	19
4.4.2 Data Pre-Processing.....	22
4.4.2 Data Augmentation.....	23
4.4.3 Prediction of damage via Machine Learning Model	25

4.4.3.A Multi-Class Classification.....	25
4.4.3.B Two-Encoder U-Net Model	26
4.4.3.C Components of a Convolutional Neural Network	28
4.4.3.D Training of the machine learning model.....	33
<u>5.</u> Test Results and Findings	35
Conclusions.....	39
Web References	40
References.....	41

LIST OF TABLES

Table 1.1	Types of Disasters	1
Table 4.1	Disaster Information used in the training dataset	20
Table 4.2	Disaster Information used in the case study test dataset	21
Table 4.3	Multi-Class Classification	25
Table 4.4	Training Parameters	33
Table 4.5	Localization Scores	34
Table 4.6	Class-wise comparison with xBD baseline dataset	34
Table 5.1	Outputs of Images	35

LIST OF FIGURES

Fig 1.1	Regional Distribution of disasters by Type	2
Fig 1.2	Post Disaster Needs Assessment Procedure (Acc. To. Government of India PDNA SOP)	3
Fig 1.3	Procedure for Collecting/Collating/Reporting of Disaster Damage Data	3
Fig 3.1	Topographic map of Nepal	10
Fig 3.2	Terai Region map of Nepal	11
Fig 3.3	Affected regions in 2017 Nepal Flooding	12
Fig 4.1	Google Colaboratory Interface	16
Fig 4.2	Anaconda Prompt IDE	17
Fig 4.3	Visual Studio Code Interface	17
Fig 4.4	Jupyter Notebook	18
Fig 4.5	Disasters and their locations in the xBD dataset	21
Fig. 4.6	Pre and Post disaster images in the xBD Dataset	21
Fig. 4.7	(a), (b) Pre-Disaster Image, Pre-Disaster Labels (c), (d) Post-Disaster Image, Post-Disaster Labels	22
Fig. 4.8	Examples of Data Augmentation used on the xBD Dataset	24
Fig. 4.9	Two-Encoder U-Net Model architecture	26
Fig. 4.10	Training and Validation loss and accuracy curves across 10 epochs	33
Fig. 5.1	Area of Testing Imagery. (Source: Google Maps)	34

Chapter 1

Introduction

A disaster is defined as a catastrophe, mishap, calamity or grave occurrence in any area, arising from natural or man-made causes, or by accident or negligence which results in substantial loss of life or human suffering or damage to, and destruction of, property, or damage to, or degradation of, environment, and is of such a nature or magnitude as to be beyond the coping capacity of the community of the affected area.

Natural disasters, such as earthquakes, hurricanes, and floods, frequently result in devastating damage to buildings and infrastructure, often posing immediate risks to public safety and infrastructure functionality. In the field of civil engineering, assessing the extent of damage accurately and swiftly is of paramount importance for effective disaster response and recovery.

General types of disasters include:

Table 4.1 Types of Disasters

No.	Category of Disaster	Disaster
1	Water & Climate related disasters	1. Floods 2. Cyclones 3. Tornadoes 4. Snow Avalanches 5. Cloud Burst
2	Geological related disasters	1. Earthquakes 2. Landslides 3. Minor Fires
3	Accidental disasters	1. Forest Fires 2. Building Collapses 3. Industrial Disasters 4. Nuclear Disasters 5. Biological Disasters 6. Chemical Disasters

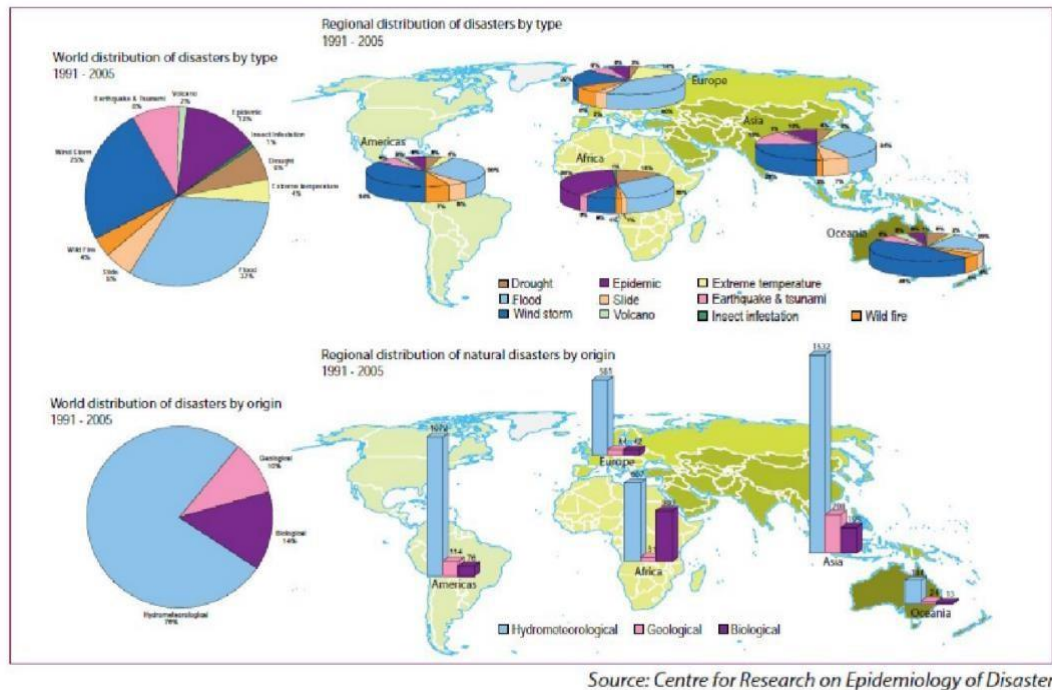


Fig.1 Regional Distribution of disasters by Type

Post a natural disaster, as a major stakeholder, the first action that takes place is a post-disaster assessment. In this assessment the stages include.

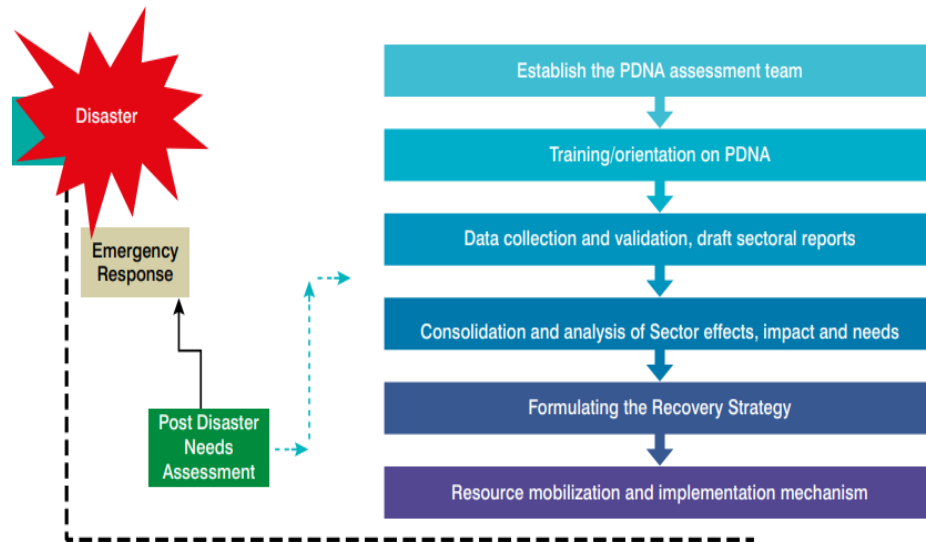
- Stage 1: Rapid Damage and Needs Assessment
- Stage 2: Humanitarian Needs Assessment
- Stage 3: Critical Infrastructure Assessment
- Stage 4: Health and Sanitation Assessment
- Stage 5: Shelter and Housing Assessment
- Stage 6: Environmental Impact Assessment

After these assessments, data is delivered to required stakeholders and then further action can be taken accordingly. This includes repair and rehabilitation of damaged facilities, reconstruction of structures, which is part of the reconstruction and planning aspect of this post-disaster damage control.

During various natural disasters, like earthquakes, floods, taking place in urban/densely populated/heavy infrastructure zones, there is a major chance of damage occurring to these structures like buildings, apartment complexes, bridges, community centers, flyovers, etc.

This damage can be captured as images by satellites & aerial imagery, unmanned aerial vehicles, drones & LiDAR, in the form of raster data.

These images according to the resolution of the imagery, show us the finer details and changes that take place after the disaster, and can be used for further processing and interpretation of the disaster.

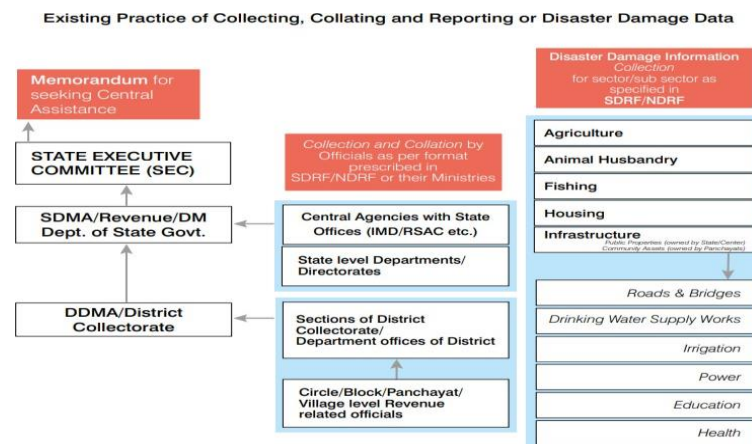


*Fig.2 Post Disaster Needs Assessment Procedure
(Source: To. Government of India PDNA SOP)*

Typically, satellites like Landsat-8 and organizations like USGS, ISRO, NASA keep taking regular photos of the surface of earth at various resolutions according to the requirements and the specifications of the satellites, which typically find applications in diverse fields, including agriculture, forestry, environmental science, urban planning, disaster management, and climate monitoring.

These satellite imagerys are also used to analyse trends, make informed decisions, and respond to emerging challenges.

By the use of Computer Vision as well as Convolutional Neural Networks and or other Machine Learning and Artificial Intelligence technologies, one can predict the amount of destruction and damage to the various structures after a disaster has taken place in that region.



*Fig.3 Procedure for Collecting/Collating/Reporting of Disaster Damage Data
(Source: Government of India PDNA SOP)*

This dissertation explores an innovative approach to building damage assessment and rehabilitation recommendation processes by harnessing the potential of computer vision and machine learning technologies.

1.1Need of the study

Typically, when a Rapid Post Disaster Assessment takes place, images of the disaster are almost immediately taken, and are typically processed manually or with some algorithms in parallel with manual surveys and assessments on the disaster-stricken field.

The major need of this study is to decrease the time required for first-action assessments that are done. This will reduce the time and labour required for the manual assessment of buildings that is done before any field-based observations and other reconstruction/rehabilitation plans are taken, and can be used as a baseline for further measures.

Further, this technology will be useful in the fields of not only disaster management, but also town and city planning, as a tool to be used in a disaster-based scenario.

The study also gives focus on damage done by some Natural Disasters in India, also giving us a case-study based approach which can be used to verify our results from the technology as well.

1.2Scope of the work

A lot of field investigations for damage to buildings are carried out after large earthquakes. In concordance with the variety of investigations, the building damage descriptions are really incoherent because a well-coordinated damage scale has not been prepared yet. In order to achieve satisfactory results in building damage detection, various damage grading schemes have been defined according to the type of the remote sensing imagery. The walls of buildings are difficult to detect from the satellite images.

For example, if there are some cracks in the walls, we cannot detect them with satellite images. It is also difficult to construct an explicit correspondence between the building damage grades and their appearance in remote sensing imagery.

Therefore, the building damage classification based on remote sensing imagery is difficult to reach the same level of accuracy as a field survey.

This research is trying to bridge the following gaps observed in the previously carried out work

1. To give us an idea of damage caused to the structures and to place it on an interpretable scale.
2. To increase the speed of processing of data, which will be used to get the final results of destruction of buildings
3. To carry out a proper study on latest Natural Disasters, and try to match the data obtained manually by surveys
4. To utilise localized data from Natural Disasters and use it on older CNNs to improve accuracy of the model.

1.3 Objectives

5. To collect information and pre- and post-event imagery of various natural disasters across various regions and various time periods for training the model
6. To perform a Rapid Building Damage Assessment with the help of Deep Learning Model with Convolutional Neural Networks and determine the impact of the Natural Disaster and damage done to the structures.
7. To analyze the results of the model and interpret the findings as a Rapid Assessment which can be further utilized for future actions.
8. To apply this model to pre- and post-event imagery obtained from the Nov. 2017 Nepal Floods, and to compare consequences to ground data, as a part of the case-study.

Chapter 2

Literature Review

Lary et al. (2015) and Tuia and Camps-Valls (2009) explored neural networks and their applications in remote sensing. Focusing on Artificial Neural Networks, which imitates the structure of our brain's neural networks. A neural network comprises numerous interconnected neurons that generate a series of neural activations, which is if a neuron is activated or not. These neurons receive input from sensors that perceive things through 'weighted' connections, that signify the connective strength between two neurons, originating from earlier activated neurons.

The use of a subclass of neural networks, which are the Convolutional Neural Networks (CNNs), excel in image analysis and geospatial problems, as they align towards object recognition and image recognition. CNNs integrate three major Neural Network architectural concepts, which are local receptive fields, shared weights and spatial or temporal subsampling, which in turn adds a lot of depth to the types of data that might be used and predicted in the same.

CNNs possess multi-layer interconnected channels with high capacity for learning features and classifiers, where they also adjust parameters simultaneously and classify them together. Spectral and Spatial information can be encoded and classified automatically.

With CNNs, there is also use of multiple Image Classification techniques where Xin and Wang (2019) and Kattenborn et al. (2021) also gave more insight on implementation of CNNs on remote sensing imagery, typically when used for three major types of learning algorithms, which are supervised, unsupervised and semi-supervised learning, where the model can be trained with varied levels of training data, as well as transfer learning algorithms, which are typically explored along with pre-trained CNNs like ImageNet, or YOLO. Further elaboration is done on aspects like Feature Selection and Extraction of data from larger datasets. This is taken into consideration as datasets made from satellite imagery have datasets in higher dimensions, causing increase in computational time, which can be reduced with use of SVM- based feature eliminators.

For any type of large datasets with multiple types of data classes within it, Alsafy et al. (2020) explores a technique known as 'multi-class classification'. Classification is basically a data mining attribute which specifies classes to groups of data in order to help with further analysis. Methods like Support Vector Machine, K-Nearest Neighbors Classifier, Naïve Bayes, Decision Trees, Neural Networks, Fuzzy Logics, K-Means Classifier, etc. are utilized to make use of this multi-class classification technique.

In an image-based dataset, Hussain et al. (2013) utilized Pixel Based Classification Techniques. Pixel-based techniques analyze the spectral characteristics of surface features in multiple or hyperspectral images, with the pixel serving as the basic unit of analysis. To evaluate individual pixels within images containing thousands or millions of them, statistical operators are used, and spatial context is often less important.

There are two main categories of pixel-based classification: unsupervised and supervised. Unsupervised classification involves grouping pixels into classes based on their natural associations without prior knowledge or training data of the study area. Common unsupervised classification algorithms include k-means and their variants. K-means clustering, a popular unsupervised classification algorithm, starts with initializing cluster centroids. Then, each pixel is assigned to its nearest centroid, and the algorithm recomputes centroids based on these assignments. This process is repeated until the centroids no longer change significantly.

Pixel grouping is achieved by an image segmentation process which can be classified into three major image segmentation areas, named spatial clustering, thresholding, and region-growing. (Haralick and Shapiro, 1985)

Dong et al. (2013) various types of pre- and post-earthquake data imagery, their sources, and methods of acquisition. Building damage detection heavily relies on remote sensing and GIS data, including optical, SAR, LiDAR, and vector maps. These sensors can be airborne or spaceborne, with resolutions ranging from 10 to 0.3 meters. Each data type has its unique strengths and is useful for different scenarios and purposes.

Most studies suggest that a resolution of at least 1 meter is required to detect individual damaged buildings, with 0.5 meters providing a more reliable outcome. Different types of remote sensing data complement each other in building damage detection. Optical imagery, for example, is a valuable source of pre-event reference data due to its large image archives, and it can be easily interpreted to assist first responders after an earthquake. SAR imagery, on the other hand, can be acquired immediately after an earthquake regardless of weather conditions. LiDAR data is especially helpful in identifying pancake-type collapses and estimating debris volume, as it detects building damage by evaluating elevation changes or patterns.

The role of Geomatics in disaster monitoring and mitigation, specifically highlighting aerial photogrammetry facilitated by UAV platforms for rapid mapping during emergency response operations. The classification of damage, referencing methodologies such as Copernicus EMS, the BAR Methodology, and UNOSAT can be taken into consideration (Calantropio et al. 2021).

In this, they have explored and created 'DEEP' (Digital Engine for Emergency Photo-analysis), an open-source engine for analysis of UAV based images and classify buildings as 'damaged' or 'not-damaged'.

It also contains a case-study of the model on the 2016 Central Italy Dataset, which showcases its Segmentation, Classification and Damage Verification capabilities. DEEP utilizes Convolutional Neural Network, and a different image classification model and trains the data based on the same technology. The first test concerned the segmentation model of DEEP that can be used (in addition to the existing cartography) to automatically detect the building footprint, which will be later classified by the classification module.

During the development of the tests related to this research, it has been observed that DEEP works well with orthoimages that have a GSD of $9 \text{ cm} \pm 3 \text{ cm}$; this is a good compromise in terms of the orthoimages' quality, the number of images required for its generation, flight altitude, and time of flight. For example, a DJI Phantom 4 RTK can obtain a 9 cm/pixels GSD with a flight altitude of about 330 m, while the SenseFly eBeeX can get a 9 cm/pixels GSD with a flight altitude of approximately 390 m.

It has also been observed that higher resolutions (lower GSD) introduce false positive detections during the segmentation phase and increase the overall processing time; lower resolutions (higher GSD) instead do not make the classification model working correctly, making it difficult or impossible to detect the level of damage accurately.

In earlier models, like the one created by Hao et al. (2021), a combination of the UNet model and the Siamese architecture was used. In this setup, the UNet model was responsible for learning the semantic segmentation of buildings, and the Siamese network focused on damage classification by comparing the segmented outputs. Building on this, C. Wu et al. (2021) enhanced the model by incorporating an attention-based UNet. In this modified architecture, attention is applied to the incoming layers from the encoder module before being fused with the up-sampled features in the decoder.

Xia et.al (2022) serves as a benchmark solution for the detection of building damage on a dual- class scale on the xBD dataset. This solution uses a self-supervised learning approach, with the use of a Student-Teacher Contrastive Learning model, with the use of DINO (No Negative Samples) and UPerNet and ImageNET for pre-training the Neural Network, for various segmentation tasks.

These architectures work by leveraging global features learned from both pre- and post-disaster images. The key idea is to merge these features to understand the changes (specifically, damage) at the final stage of the model. However, there's a challenge in these models – the majority of parameters are geared towards learning global features for individual images. This focus makes it challenging for the model to effectively learn the changes, especially when considering multi- task supervision.

Gerke et al. (2011) showcased the use of Pictometry, Inc. data for post-earthquake imagery in Port-Au-Prince, Haiti, which is known for the seismically active zone called the 'Ring of Fire'. This corresponds to a flying height above ground of approximately 1,000 m, leading to a ground sampling distance (GSD) ranging between 10 cm (foreground) and 16 cm (background). For every point on the ground, at least one image per viewing direction is available; however, stereo overlap from a single direction is not guaranteed since the along-track overlap is smaller than 50 per cent.

The study investigated feature importance in a building classification task using dense matching and stereo information. Disabling color features significantly impacted Tree classification, dropping from 83% to 32%. Omitting optical texture reduced accuracy for all classes (except Trees) by about 10%, emphasizing texture importance. Removing 3D points, straight lines, and stereo-dependent features resulted in decreased accuracy, highlighting the role of 3D orientation and planarity in classification. Façade classification was notably affected by the absence of straight lines. The analysis of E- and W-looking images without stereo-dependent features showed that stereo information, particularly derived from stereo matching, significantly contributed to accurate façade classification. The study suggests the importance of considering various features, especially 3D information, for accurate building classification in urban environments.

When taking into consideration about Natural Disasters like Earthquakes, early disaster responses are very important, and hence a quick estimation of damage is required (Miura et al. 2023). Multiple solutions like use of earthquake intensity-bearing ShakeMap system, use of LIDAR data, and use of Satellite/Aerial/UAV imagery is taken into consideration to form this rapid estimation and assessment of damage. Seismic activities can be observed by both physical visualization as well as seismic intensity visualization, where the impact of an earthquake on a region depends on both Shear Wave velocity as well as various urbanized city aspects, like population density, buildings, infrastructure, etc. Rao et al. (2023) also gave an pre- and post- event Synthetic Aperture Radar imagery-based as well as an integrated building inventory data solution for the building data assessment, with an average accuracy of around 50%.

Tu et al. (2021) presents a flood risk assessment for Shanghai, China, which provides an indication of what buildings (including residential, commercial, office, and industrial) will be exposed to flooding and its damage. This paper gives us an idea of how post flood, building damage assessment actually takes place and gives us various indicators on where the data is required and what the use case of the data is, especially in an urban environment. It also provides us with various measures that might be taken in post-flood scenario in terms of reconstruction and rehabilitation.

Melamed et.al (2021) gave us more insight on the xBD dataset, which is typically used to pre- train the CNN models, and also on how to make it more useful by the use of new baselines that included 4PS-Local, 4PS-Contour and Mask-RCNN. These baselines serve as better improvements in terms of corresponding to lesser losses and more optimized f1-values (the target metric). Major aspects of the paper include the optimization of the xBD/xFBD dataset's detectability of isolated damaged buildings, as a lot of individual chipped but not majorly damaged buildings are also seen in images of the dataset, but not detected as such.

Chapter 3

Nepal Floods 2017: An Overview

3.1 Overview and Background Information

Nepal, a country characterized by its diverse topography ranging from the flat plains of the Terai to the rugged Himalayas, experiences significant seasonal flooding, particularly during the monsoon season. This case study aims to analyze the causes, impacts, and responses to the major flooding events that occurred in Nepal from 2017 to 2021. By examining these events, we seek to understand the recurring nature of these floods, the underlying factors contributing to their severity, and the effectiveness of the mitigation and response efforts.

Nepal is a landlocked Himalayan country in South Asia. It is home to the world's highest mountains along its northern borders but also has relatively extensive floodplains in the southern part of the country. It stretches between the latitudes of 26°22'N and 30°27'N, and longitudes of 80°4'E and 88°12'E and is roughly aligned southeast-northwest.

The total physical area of Nepal is approximately 147,181 square kilometers of which about 143,351 square kilometers constitutes land, and about 3,830 square kilometers constitutes permanent inland water. The country has an almost rectangular shape, extending from the southeast to northwest over a distance of approximately 900km and along the north-south direction with distance varying between 80 to 250km with elevation ranging from about 70m in the south to 8,848m above sea level at the peak of Mount Everest.

Despite being a small country, the landscape of Nepal is diverse; it ranges from the humid plains in the south to the lofty Himalayas in the north. Thus, Nepal can be divided into three different geographical regions and each region extends from east to west across the country. The three different regions are: the Terai, the Middle Hills, and the Himalayas. Figure taken from the Inception Report shows a rough delineation of the three zones based on elevation and approximate extents of glaciated areas, key cities and major floodplains.

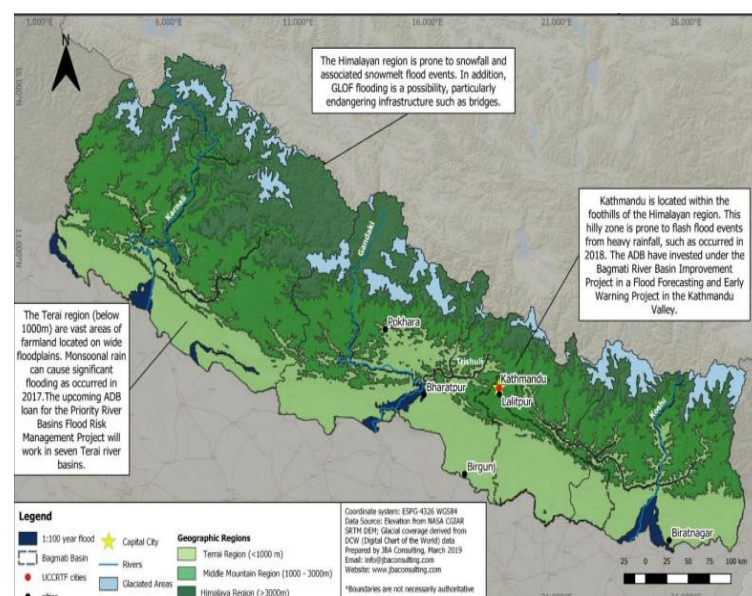


Fig.3.1 Topographic map of Nepal

Frequency and intensity of extreme flood events is said to be increasing with the impact of climate change. Without comprehensive flood protection or appropriate risk-sensitive urban planning, increased ongoing economic and development activity in the flood plains will likely increase the economic damage associated with floods in the coming years.

3.2 Locations of the floods

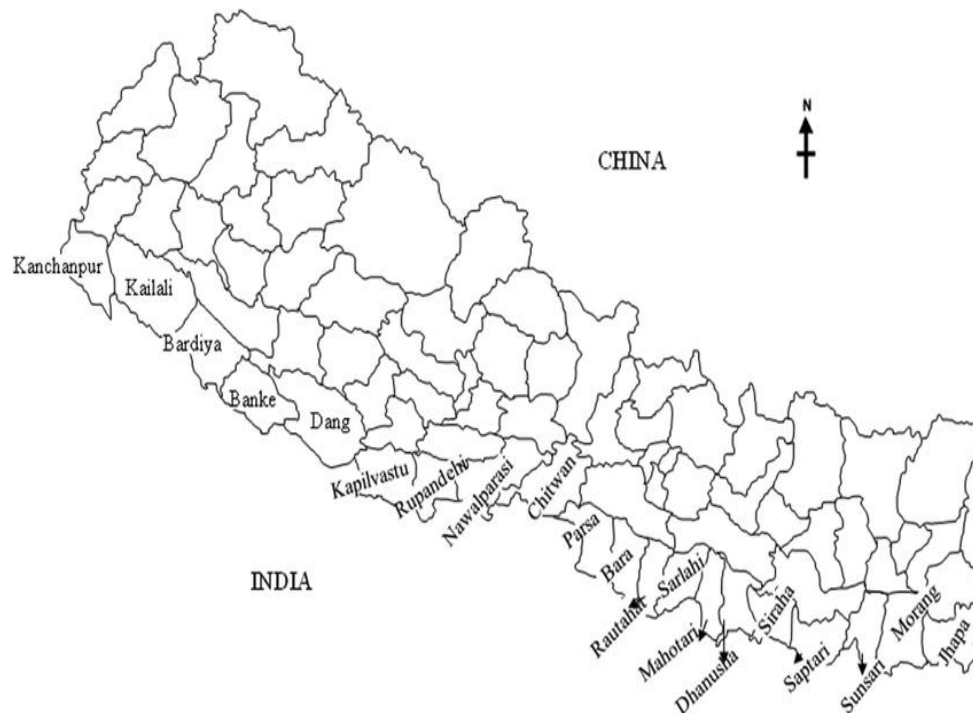


Fig.3.2 Terai Region map of Nepal

Nepal, with its varied topography from the Terai plains to the Himalayas, experiences significant flooding during the monsoon season (June-September). This heavy rainfall, combined with complex river systems and factors like deforestation and urbanization, makes Nepal highly susceptible to floods.

Major districts which have suffered due to flood are Jhapa, Siraha, Saptari, Morang, Sunsari, Dhanusa, Mahottari, Sarlahi, Bara, Parsa, Rautahat, and Chitwan.

3.3 Causes of the floods

1. Monsoon Rains:

Nepal experiences heavy monsoon rains typically from June to September. From 2017 to 2021, the monsoon season was particularly intense, leading to excessive rainfall that caused widespread flooding.

The country received around 20-50% more rainfall than the average during the monsoon season, which overwhelmed rivers and drainage systems.

2. Geographical Factors:

Nepal's topography, with its steep mountainous terrain, contributes to rapid runoff and increased river discharge during heavy rainfall.

The steep slopes of the Himalayas cause rapid runoff during heavy rains. Water flows quickly down the mountains, accumulating in river basins and causing rivers to swell rapidly.

The Himalayan region also experiences glacial melt, which can contribute to higher water levels in rivers.

The Terai is a flat, low-lying region bordering India, making it particularly vulnerable to flooding from both heavy rainfall and river overflows. Floodwaters from the Himalayan rivers can inundate large areas of the Terai.

3. Complex River Networks:

Nepal has numerous rivers and tributaries that originate in the Himalayas. During the monsoon season, these rivers receive large volumes of water, leading to increased river discharge and potential flooding downstream.

4. Deforestation and Land Use:

Deforestation and poor land management practices, such as overgrazing and unplanned urbanization, reduce the land's ability to absorb water, leading to increased surface runoff.

Encroachment on riverbanks and floodplains for agriculture and settlement reduces the natural flood-absorbing capacity of these areas.

5. River Siltation:

Rivers in Nepal carry large amounts of sediment from the Himalayas. This sedimentation reduces river channel capacity, making them more prone to overflowing during heavy rains.

6. Unplanned Urbanization:

Inadequate Drainage Systems: Rapid and unplanned urban growth often results in poor drainage systems, which cannot effectively manage heavy rainfall, leading to urban flooding.



Fig.3.3 Affected regions in 2017 Nepal Flooding

3.4 Types of Impacts

3.4.1 Environmental Impacts

1. Erosion and Sedimentation

Soil Erosion: Flooding causes severe soil erosion, particularly in the hilly and mountainous regions, leading to loss of fertile land.

River Sedimentation: Erosion upstream leads to increased sediment in rivers, reducing their capacity and increasing the likelihood of future floods.

2. Habitat Destruction

Flora and Fauna: Floods destroy natural habitats, affecting biodiversity. Wetlands and forests, which serve as critical habitats for various species, are often severely impacted.

Landslides: In the hilly regions, heavy rains trigger landslides that not only cause immediate destruction but also have long-term effects on the landscape and ecosystems.

3.4.2 Economic Impacts

1. Damage to Infrastructure:

Homes and Buildings: Floodwaters have destroyed or severely damaged thousands of homes, schools, and public buildings.

Transport Networks: Roads, bridges, and railways often get washed away or damaged, disrupting transportation and communication. The 2017 floods caused extensive damage to major highways and bridges, complicating relief efforts.

2. Agricultural Losses:

Crops and Livestock: Flooding inundates farmlands, destroying crops and killing livestock. This not only affects the immediate food supply but also the long-term livelihoods of farmers. For example, the 2018 floods led to significant losses in rice crops, a staple food in Nepal.

Economic Strain: The cumulative economic losses due to flooding over the past eight years run into hundreds of millions of dollars, impacting national and local economies.

3. Loss of Life and Displacement

Fatalities: Over the past eight years, floods have resulted in hundreds of deaths. For instance, the 2017 floods alone caused more than 150 fatalities.

Displacement: Thousands of people have been displaced annually due to floods, with significant numbers needing to evacuate to temporary shelters. The 2019 floods affected over 100,000 people.

4. Community Disruption

Displacement: Long-term displacement disrupts community structures and social networks. Families often live in temporary shelters for extended periods, affecting their quality of life and social cohesion.

Education: Schools are often used as shelters during floods, disrupting education for children. Floods also damage educational infrastructure, leading to long-term disruptions.

5. Poverty and Vulnerability

Economic Hardship: Repeated flooding pushes vulnerable populations further into poverty. The loss of homes, livelihoods, and access to resources makes it difficult for affected families to recover.

Increased Vulnerability: Communities affected by frequent flooding are often caught in a cycle of poverty and vulnerability, with limited resources to invest in flood resilience measures.

3.4.3 Infrastructural Impacts

1. Structural Damage:

Many buildings, particularly those constructed with traditional materials like mud, wood, and bricks, suffered severe structural damage.

Walls collapsed, foundations were undermined, and roofs were destroyed, leading to partial or complete destruction of houses.

Inundation and Water Damage:

Buildings in low-lying areas, especially in the Terai region, were inundated with water, causing significant water damage.

Prolonged water exposure led to weakened structural integrity, mold growth, and deterioration of construction materials.

2. Landslides:

In hilly and mountainous areas, landslides triggered by the heavy rains caused buildings to be buried or swept away.

Many homes and buildings situated on slopes were either destroyed or rendered uninhabitable due to landslide impacts.

3. Infrastructure Damage:

Floodwaters and landslides damaged roads, bridges, and public infrastructure, making access to affected buildings difficult and delaying rescue and relief operations.

Utilities such as electricity, water supply, and sewage systems were disrupted, compounding the impact on buildings and their habitability.

4. Specific Examples:

a) Residential Buildings:

Thousands of homes were either completely destroyed or severely damaged. In some districts, entire villages were submerged or washed away.

Displacement of families led to temporary shelters being set up, often in schools or community centers, further straining public buildings.

b) Commercial Buildings:

Shops, markets, and small businesses faced significant losses due to flood damage to their premises and goods.

Many commercial buildings in urban areas experienced basement flooding, damaging stored goods and infrastructure like electrical systems.

c) Public Buildings:

Schools, hospitals, and government buildings were not spared. Many schools were damaged or converted into temporary shelters, disrupting education.

Health facilities were overwhelmed with damage to infrastructure, impacting their ability to provide medical care.

d) Cultural and Historical Buildings:

Some cultural heritage sites and historical buildings also suffered damage, which posed a significant loss to the community and tourism.

3.5 Response and Mitigation:

In 2021, the government approved an adaptation-based action plan that addressed climate vulnerabilities and risks in the short, medium, and long term. The National Adaptation Plan (2021-2050) set out priority programs in nine sectors to reduce climate vulnerability and risk. As part of the adaptation plan, the disaster risk reduction and management sector aims to build climate resilience through policy reforms. It will also maintain, upgrade, and strengthen early warning systems and multi-hazard monitoring to allow for climate adaptive actions. Parallely, the water resource and energy sector aim to bridge climate information gaps and promote climate-informed decision-making. Part of the goal for this sector is to establish a Glacial Lake Outburst Flood (GLOF) risk reduction and early warning systems and construct climate resilient check dams.

In conjunction with government efforts, ICIMOD is working on customizing NASA GSFC's Landslide Hazard Assessment for Situational Awareness model to develop a landslide mapping and forecasting system for Nepal. This landslide inventory can be used to make data-backed decisions that should improve hazard management and action plans. The model can also generate landslide predictions which can support early warning systems and evacuation. Similarly, the institution has an existing Flash Flood Prediction Tool (FFPT) for Nepal that predicts flash floods for over 12,000 river segments in Nepal. FFPT can be used for flood forecasting, early warning system, and response decisions.

It is imperative for the government to effectively integrate such tools and models into its disaster risk management strategy to reduce the damage and destruction that comes with the monsoon rains. As the annual impacts of climate change intensify, the government must prioritize investment in data collection and analysis at all levels. By doing so, a proactive disaster management system can be effectively implemented before it reaches a point beyond our control.

Chapter 4

Methods and Materials

4.1 Technologies Used

The major aspect of this project is based on a high-level, general-purpose programming language known as ‘Python’. It was then powered by ‘Jupyter Notebooks’ platform hosted by Google over the cloud, using their Google Colaboratory service, along with locally hosted environment using Visual Studio Code and Anaconda IDE.

Technologies Used:

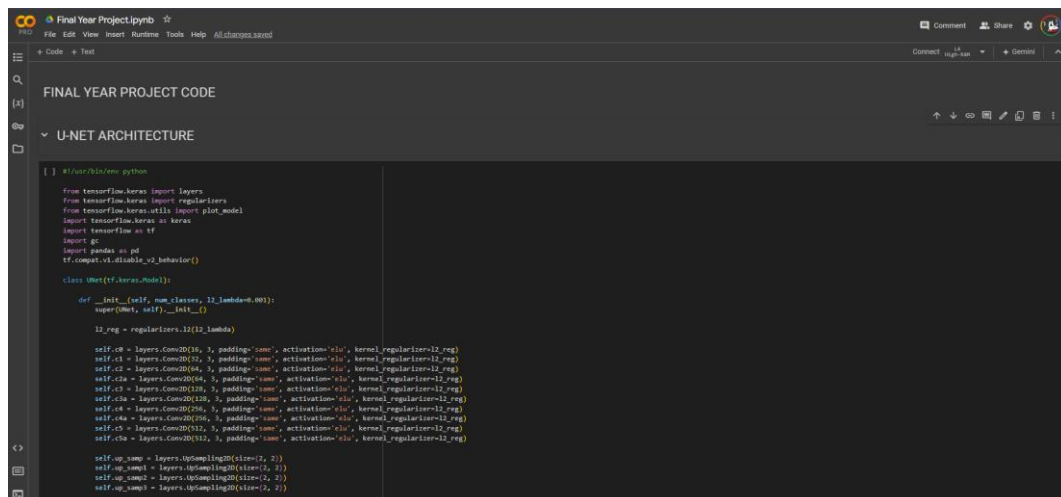
- Google Colaboratory / Google Cloud Service
- Anaconda IDE
- Visual Studio Code
- QGIS (For Visualization)
- Google Earth Engine (For Dataset Testing)

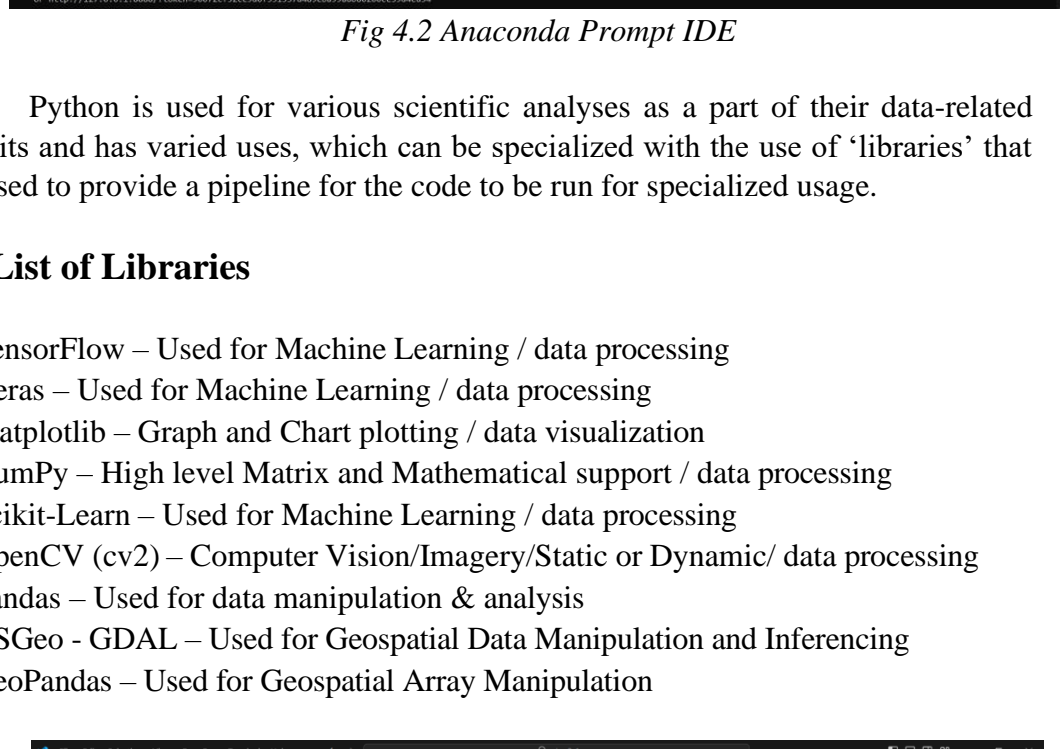
Technical Specifications for Local Machine

Processor	I7-11800H – Octa-Core Processor
RAM	16GB DDR4
Graphics Card	Nvidia RTX 3050 Ti
Graphics RAM	4GB GDDR5
Storage	256 GB SSD / 1TB HDD

Technical Specification for Google Colaboratory / Google Cloud Service

Processor	Intel(R) Xeon(R) CPU @ 2.20GHz – 10 Core Processor
RAM	60 GB DDR4
Graphics Card (1)	Nvidia L4
Graphics RAM (1)	24 GB GDDR5
Graphics Card (2)	Nvidia A100
Graphics RAM (2)	40 GB GDDR5





```
1 import os
2 os.environ["MKL_NUM_THREADS"] = "1"
3 os.environ["NUMEXPR_NUM_THREADS"] = "1"
4 os.environ["OMP_NUM_THREADS"] = "1"
5
6 import numpy as np
7 np.random.seed(1)
8 random = np.random
9 random.seed(1)
10 import pandas as pd
11 import cv2
12 import timeit
```

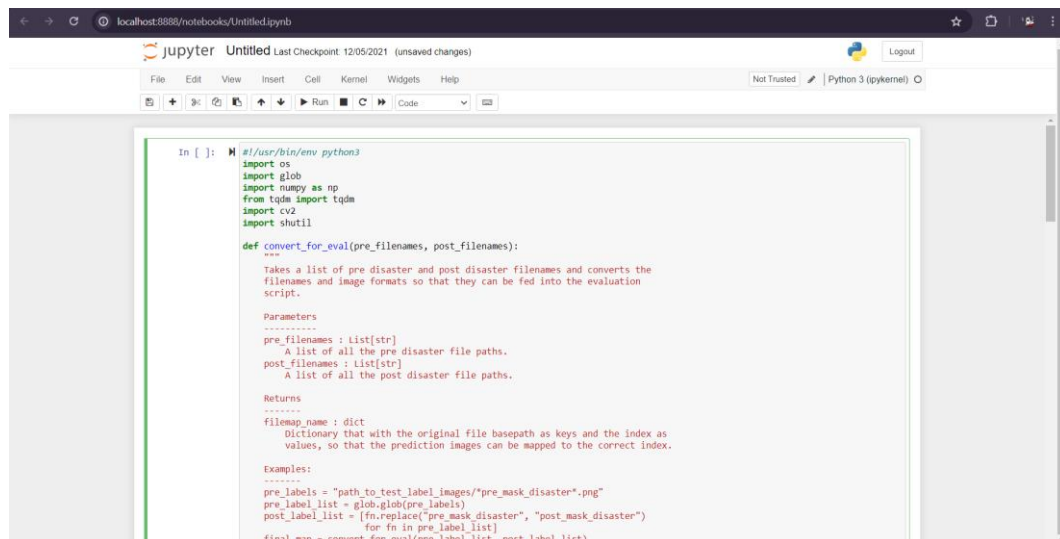
- > targets
- architecture.fdb_latexmk
- architecture.fdb

- ```

16 from multiprocessing import pool
17 from skimage.morphology import square, dilation
18 from skimage import io
19
20 from shapely.wkt import loads
21 from shapely.geometry import mapping, Polygon
22
23 # import matplotlib.pyplot as plt
24 # import seaborn as sns
25
26 import json
27
28 masks_dir = '/content/drive/MyDrive/FVP/Custom Train/masks'
29
30 train_dir = '/content/drive/MyDrive/FVP/Custom Train/train'
31 labels_dir = ''
32
33 def mask_for_polygon(poly, im_size=(1024, 1024)):
34 # ... (code continues)

```





*Fig 4.4 Jupyter Notebook Interface*

Along with this, use of Convolutional Neural Networks, which in turn is a subset of Deep Learning will be used. In deep learning, multiple factors will be included in the mathematical, theoretical as well as physical working of a model, which will be included in the project in itself.

### 4.3 Exploring Neural Networks

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. There are different types of Neural Networks

1. **Standard Neural Networks** The normal type of Neural Networks with a group of multiple neurons at each layer, it's only processed in the forward direction. It consists of 3 layers, Input, Hidden and Output. It is used for Simpler problems.
2. **Convolutional Neural Networks** Convolutional Neural Networks work on the principle of Convolution Kernels and is most commonly used for Image processing. The process of convolution is applied in these types of networks, which states that for a mathematical operation on two functions ( $f$  and  $g$ ) that produces a third function ( $f*g$ ) that expresses how the shape of one is modified by the other.
3. **Recurring Neural Networks** RNNs are Neural Networks which are a derivative of Standard Neural Networks in which, a looping constraint on the hidden layer of the SNN turns it into an RNN. This type of NN is used for one-dimensional temporal sequence or multi-dimensional sequence data, like Audio, Video.
4. **Hybrid Neural Networks** These involve complex problems like Autonomous Car Systems which use different types of NN in sync with each other.

Artificial Neurons sometimes input the size, compute this linear function, takes a max of zero, and then output the estimated price. This is called famously as RELU, which stands for Rectified Linear Unit Function.

In Neural Networks, we will be working with various parameters that will define how a Neural Network will run and also determines the accuracy of the model as well.

### **Loss Functions:**

Loss functions measure the inconsistency between predicted values and actual values in machine learning models. They quantify how well or poorly a model is performing, guiding the optimization process to minimize errors during training.

### **Activation Functions:**

Activation functions introduce non-linearities to the neural network, allowing it to learn complex patterns and relationships in data. They determine the output of a node or neuron, helping the network to capture and understand intricate dependencies within the input data.

### **Optimizers:**

Optimizers adjust the parameters of a model during training to minimize the loss function, facilitating efficient convergence to the optimal solution. They control the learning rate and update model parameters, affecting the speed and accuracy of model training.

## **4.4 Methodology of Assessment**

The major steps involved in the methodology of this assessment involve:

1. Data Collection
2. Data Pre-processing
3. Data Augmentation
4. Prediction of damage via Machine Learning Model
5. Results

### **4.4.1 Data Collection**

Multiple datasets fit the use-case of the project, which typically required Very-High-Resolution Satellite Imagery, of the same region, before and after a specific kind of natural disaster. The biggest challenge we faced while finding a dataset was data annotation. Data annotation is a technique, in which data is labelled according to whatever has ‘actually happened’ by either human, or autonomous processes. In the case of our Satellite Data, the annotation was done to map out the building polygons as well as other various factors, like the type and scale of the disaster, the satellite alignment, etc.

Finally, the ‘xBD’ dataset was found suitable to meet our requirements of proper VHR imagery, added with proper data annotation, hence completing the Data Collection process.

The xBD Dataset is a robust, and an open-source dataset released for the xView-2 competition by the Defence Intelligence Unit of the Military of the United States, along with SAAB AB-owned CrowdAI.

The images in the dataset are a compilation of multiple pre and post-disaster satellite imagery of an estimated 45,000 km<sup>2</sup> of land, spanning across a few disasters which includes: -

1. Wildfires
2. Landslides
3. Dam Collapses
4. Volcanic Eruptions
5. Earthquakes
6. Tsunamis
7. Storms
8. Floods

The raw satellite data was provided to the DIU and CrowdAI by Maxar Technologies Inc. under the Maxar Open Data Program, along with US Geological Survey. The data was collected by Maxar's WorldView 1 & 2 satellites, with a pixel resolution of 50cm.

The xBD dataset was a breakthrough in its field, as the sources for Open-Source Satellite Data of this type of resolution, scale and annotation was not available, or only existed for single disaster types. With the creation of this dataset, rapid assessment models for disasters could be made on the Open-Source software network, leading to rapid developments in the fields of Satellite Imagery Segmentation and Predictions.

The Disasters that were included in the utilized dataset included

*Table 4.1 Disaster Information used in the training dataset*

| <b>Disaster Event Name</b>       | <b>Event Date</b>      | <b>Environmental Factors</b> |
|----------------------------------|------------------------|------------------------------|
| Guatemala Fuego Volcano Eruption | June 2018              | Yes                          |
| Hurricane Michael                | October 2018           | Yes                          |
| Santa Rosa Wildfires             | October 2017           | No                           |
| Hurricane Florence               | September 2018         | Yes                          |
| Midwest US Floods                | First Quarter of 2019  | Yes                          |
| Indonesia Tsunami                | September 2018         | Yes                          |
| Carr Wildfire                    | August 2018            | No                           |
| Mexico City Earthquake           | September 2017         | Yes                          |
| Hurricane Harvey                 | August 2017            | Yes                          |
| Hurricane Matthew                | October 2016           | Yes                          |
| Sunda Strait Tsunami             | December 2018          | Yes                          |
| Moore, OK Tornado                | May 2013               | Yes                          |
| Tuscaloosa, AL Tornado           | April 2011             | Yes                          |
| Lower Puna Volcanic Eruption     | Second Quarter of 2018 | Yes                          |
| Joplin, MO Tornado               | May 2011               | Yes                          |
| Woolsey Fire                     | November 2018          | No                           |
| Pinery Fire                      | November 2018          | No                           |
| Portugal Wildfires               | June 2017              | No                           |

The data for the case-study was also obtained from the same project by DIU, USGS, and CrowdAI Inc.

The Case Study Data Included

*Table 4.2 Disaster Information used in the case study test dataset*

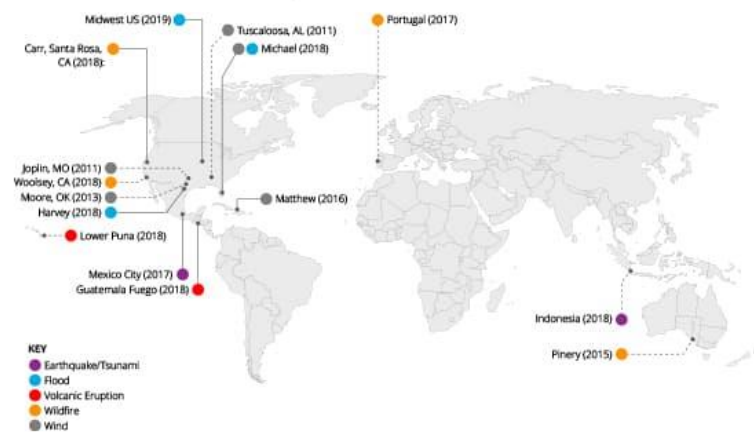
| Disaster Event Name        | Event Date | Environmental Factors |
|----------------------------|------------|-----------------------|
| Nepal-India Monsoon Floods | June 2017  | Yes                   |

The whole dataset has a total storage size of 120GB which was reduced to 25GB for the purpose of easier utilization on the Machine Learning model.

Other data sources for purposes of minor experimentation included Sentinel-2 derived satellite imagery, which was dated during the days of the disaster. The resolution of this imagery was 10m. This was also used for verifying the location and visual inspection of the region before utilizing the main dataset.

#### The xBD Dataset

**1 dataset** **6** different types of disasters **15** countries **850,736** annotated buildings  
**45,362 km<sup>2</sup>** of "before" and "after" images



*Fig 4.5 Disasters and their locations in the xBD dataset.*

Sample images obtained from the xBD dataset include:



*Fig 4.6 Pre and Post disaster images in the xBD Dataset*



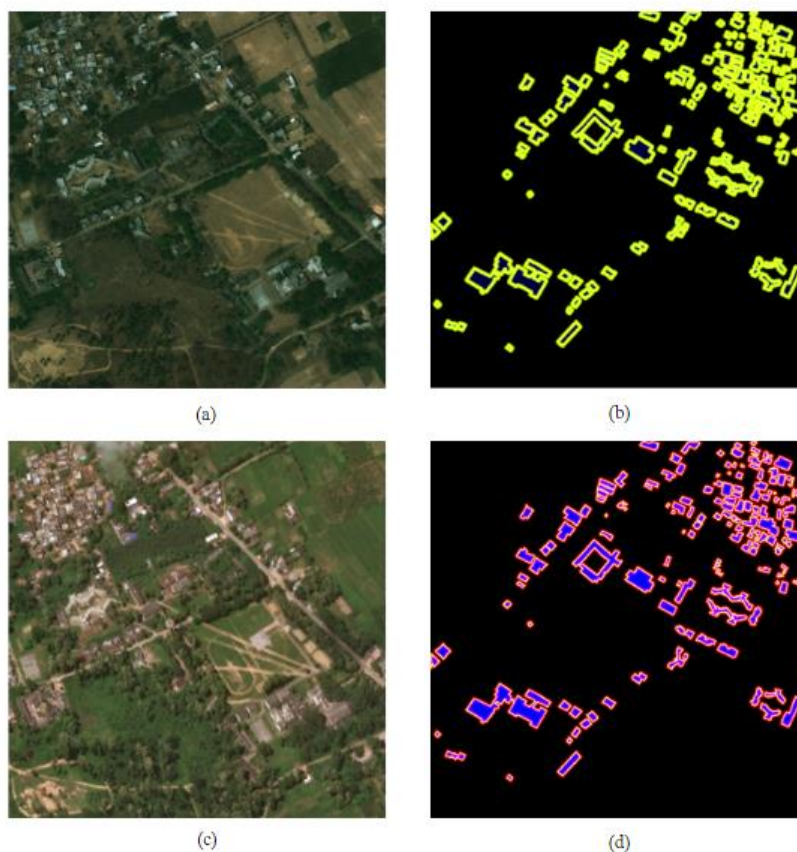
#### 4.4.2 Data Pre-Processing

Data in the datasets, now is to be pre-processed, cleaned of any outlier objects and then has to be given proper tags.

Major pre-processing is done by the method of ‘Masking’ with the help of a python library known as OpenCV (Open Computer Vision)

Masking is basically a method used in pre-processing of data, which helps us identify and finding portions of an image that are of interest to us, and helps us create a bounding region for the same.

The data for the polygons (the bounding boxes) for the training set, are provided in the xBD dataset, as well as multi-Class labels, which showcases the type of disaster that is present in the following image. The bounding boxes for the environmental regions, like rivers, forests, etc. are rough, and hard to annotate, and may cause overlaps with the building features.



*Fig 4.7 (a), (b) Pre-Disaster Image, Pre-Disaster Labels  
(c), (d) Post-Disaster Image, Post-Disaster Labels*



### 4.4.2 Data Augmentation

For a machine learning model, the performance and the generalization capability of the models depend heavily on the diversity and quantity of the training data. Data Augmentation is a technique which increases the volume of data, by utilizing various transformations and modifications of the existing images. This increases the model's ability to predict over unseen data and reduces the chances of overfitting.

Data Augmentation utilizes a range of modifications to the existing image, creating new examples that simulate and showcase examples in a real-world example. These transformations simulate different conditions under which the model might need to perform, thereby making it more robust

By utilizing a library in python called as '**albumentations**', we can apply the following 10 types of data augmentation techniques, boosting an already 17000-image dataset to about 1,70,000 images, divided into 85,000 pre-post image pairs.

The data augments utilized in our experimentation include:

#### 1. **ISONoise**

It adds ISO noise (Light Sensitivity Noise) to the image. It simulates the noise that typically appears in digitalized images. It increases robustness of the Machine Learning model against grainy & noisy images.

#### 2. **RandomFog**

It adds fog-like noise to the image, simulating foggy conditions, enhancing model performance with images with fog or clouds in it.

#### 3. **VerticalFlip**

It flips the image vertically, increasing dataset variability, also useful for images where image orientation isn't fixed normally.

#### 4. **RandomRotate**

It randomly rotates the image into a multiple of the given angle. We have used an angle of 90 degrees as a factor, causing the image to randomly rotate between 90 to 360 degrees. It adds in a layer of rotational variability to the training dataset.

#### 5. **RandomBrightnessContrast**

This augment randomly changes the brightness and contrast of the image, simulating various lighting conditions for the image taken, hence increasing its reliability for any type of brightness conditions.

#### 6. **RandomGamma**

Gamma is a characteristic of digital imaging systems that describes the relationship between a pixel's numerical value and its actual luminance, and typically represents the shape of the curve from black to white or white to black.

This function randomizes the value of the gamma function in images, leading to the model to distinguish pixels correctly in case of gamma gradient shifts caused by random events.

## 7. **Blur**

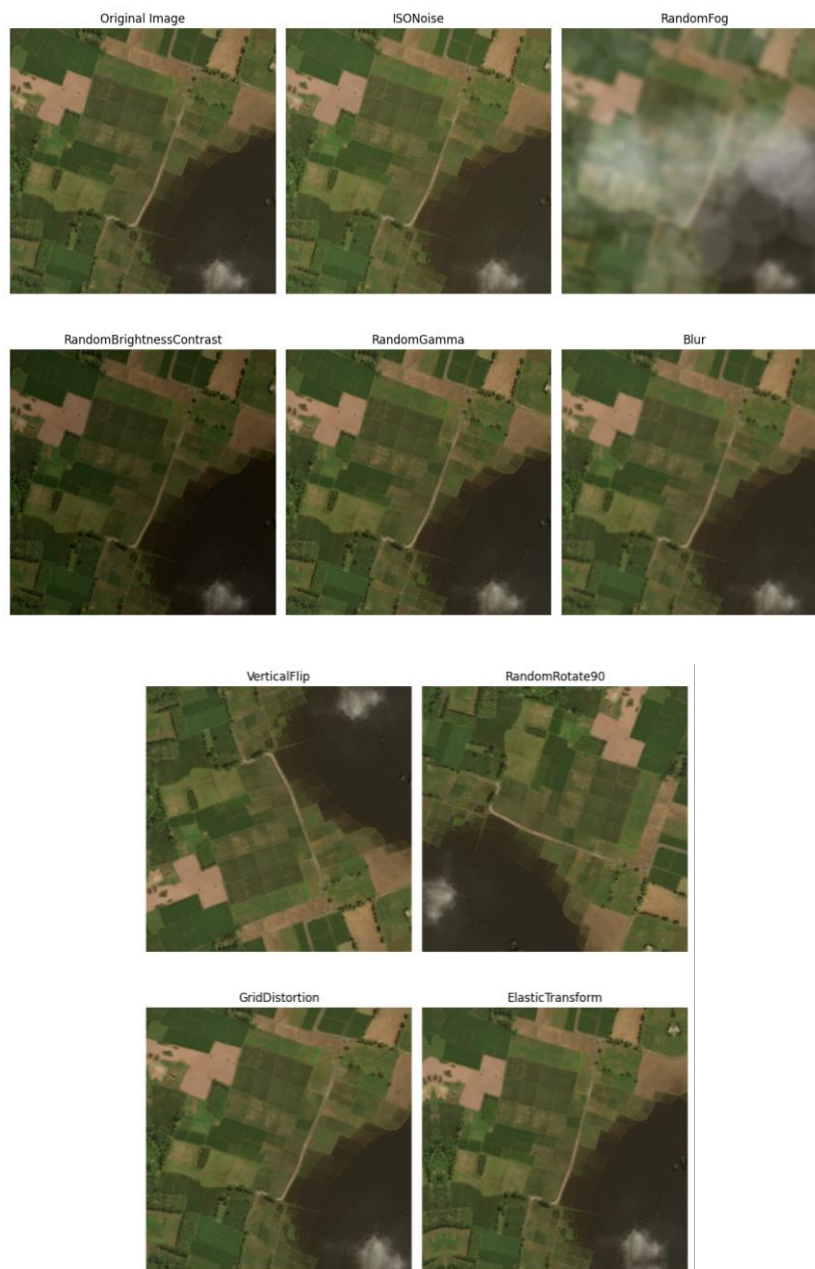
This applies a blurring effect to the picture. It simulates out-of-focus effects, helping the model handle images that are not perfectly focused.

## 8. **GridDistortion**

This technique changes around some points in the grid of the image, adding a layer of geometric distortion to the image. This increases performance of the model against non-linear imagery.

## 9. **ElasticTransform**

It applies elastic transformation to the images, stretching it randomly. It mimics natural distortion in satellite imagery that occur very consistently in raw imaging data.



*Fig 4.8 Examples of Data Augmentation used on the xBD Dataset*

### 4.4.3 Prediction of damage via Machine Learning Model

The accurate prediction of building damage following natural disasters was the aim of the project, which in turn provides additional support towards building robust Natural Disaster response. In this project, we implement a two-encoder U-Net Model.

This architecture combines two separate encoder networks to process pre- and post-disaster imagery, effectively capturing changes and anomalies that indicate damage. By integrating these encoders into a U-Net framework, the model can generate detailed segmentation maps highlighting the extent of damage. Our methodology focuses on training this model using a diverse dataset of satellite images, employing advanced preprocessing techniques and fine-tuning hyperparameters to optimize performance. The resultant model aims to deliver accurate and reliable damage assessments, facilitating swift and informed decision-making in disaster response scenarios.

#### 4.4.3.A Multi-Class Classification

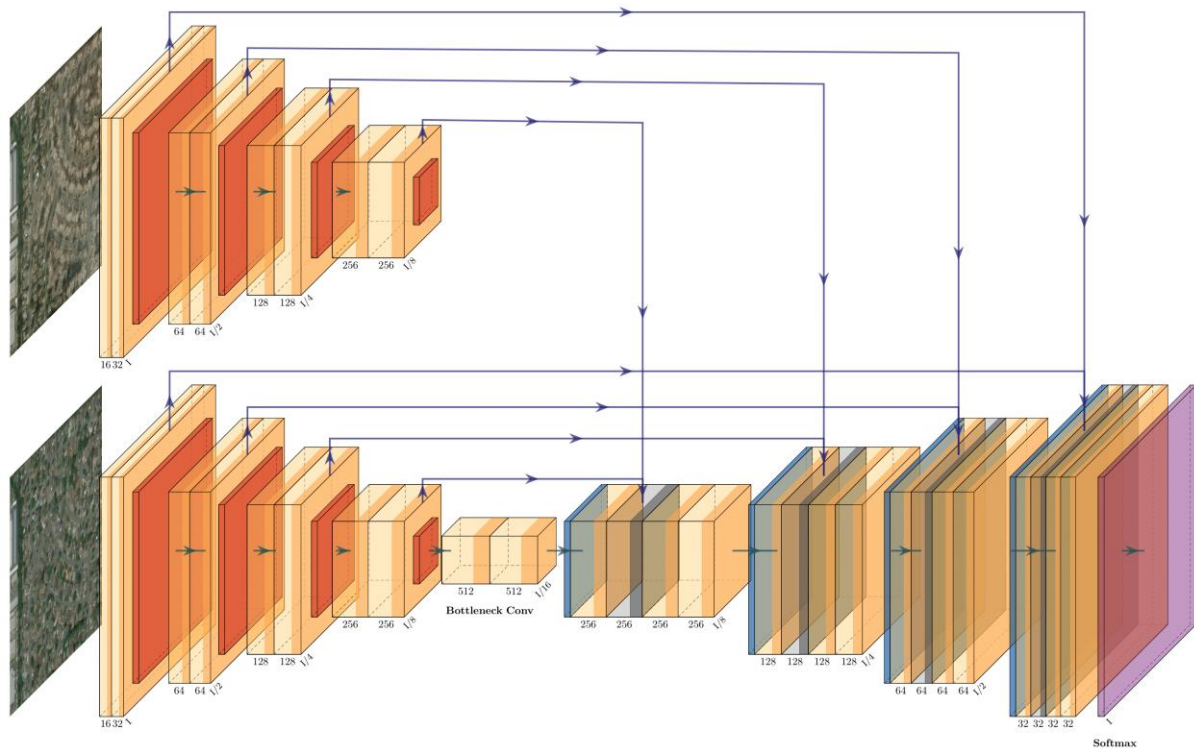
When we go for a classification problem having only two class labels, then it becomes easy for us to filter the data, apply any classification algorithm, train the model with filtered data, and predict the outcomes.

But when we have more than two class instances in input train data, then it might get complex to analyse the data, train the model, and predict relatively accurate results. To handle these multiple class instances, we use multi-class classification. Multi-class- classification is the classification technique that allows us to categorize the test data into multiple class labels present in trained data as a model prediction.

*Table 4.3 Multi-Class Classification*

| Subtype      | Category No | Representation                  |
|--------------|-------------|---------------------------------|
| Unclassified | 1           | Unclassified Buildings          |
| No Damage    | 2           | Found & Undamaged Buildings     |
| Minor Damage | 3           | Found & Minor Damaged Buildings |
| Major Damage | 4           | Found & Major Damaged Buildings |
| Destroyed    | 5           | Found & Destroyed Buildings     |

#### 4.4.3.B Two-Encoder U-Net Model



*Fig 4.9 Two-Encoder U-Net Model architecture*

To predict the changes in the two pre and post-disaster images, and to give us an output on what buildings are damaged, the following CNN-based model was utilized, known as a ‘Two-Encoder U-Net’ model. This model is an innovation to the general U-Net architecture, which is particularly useful in Image Segmentation models.

The utilization of the two-encoder outlook at a U-Net architecture reduced the training and inference times significantly, as well as it made the entire training process efficient. Through the convolution process on the image data that flows through the network processing both the pre- and post-Disaster images applying multiple layers with the kernels set to the same weights, the model noted the difference that exists between the two images and make a better assessment of the actual level of damage.

The input for the network consisted of pre- and post-disaster image tensors of shape (1024, 1024, 3). Throughout the encoder stage, the images underwent simultaneous processing, with each image traversing through dedicated encoder layers. At each stage of the decoder, the outputs of the encoder layers were concatenated with the difference between the corresponding pre- and post-disaster image encoder layers, facilitating enhanced feature representation and abstraction.

The output of the model was a tensor of shape (1024, 1024, 5), with each channel in the final dimension representing one of the five distinct classes: No Damage, Minor Damage, Major Damage, Destroyed, and Background. To obtain the predicted class for each pixel of the image, the indices of the maximum values across the channel dimension were determined, resulting in a tensor of shape (1024, 1024) containing the predicted classes for each pixel.

Therefore, employing the integration of building localization and damage classification into a single model, with a backbone based on EfficientNet technique, we have built an efficient segmentation model based on a Convolutional Neural Network.

## Understanding the U-Net model, and its terminologies:

### **Encoder Section:**

- **Convolutional Layers:** Let  $f(x)$  represent the output feature map of a convolutional layer, where  $xx$  denotes the input feature map and  $W$  denotes the convolutional kernel. The output  $f(x)$  is computed as  $f(x)=\sigma(x*W+b)$  where  $*$  denotes the convolution operation,  $\sigma$  represents the activation function (e.g., ELU), and  $b$  is the bias term. The dimensions of the output feature map are determined by the input size, kernel size, padding, and stride.
- **Batch Normalization:** Given the output feature map  $f(x)$  from a convolutional layer, batch normalization normalizes the activations across the batch dimension to stabilize training. It introduces two learnable parameters, typically denoted as  $\gamma$  and  $\beta$ , to scale and shift the normalized activations, respectively.
- **Dropout:** Dropout layers probabilistically deactivate a fraction of neurons during training to prevent overfitting. Mathematically, dropout can be represented as  $f_{\sim}(x)=f(x) \cdot \text{Bernoulli}(p)$ , where  $p$  is the dropout probability and  $f_{\sim}(x)$  denotes the modified feature map.

### **Decoder Section:**

- **Upsampling Layers:** Upsampling layers increase the spatial resolution of the feature maps. They employ interpolation techniques such as nearest-neighbor or transposed convolution to magnify the feature maps to their original dimensions.
- **Skip Connections:** Skip connections concatenate feature maps from the encoder with those in the decoder to facilitate information flow. Mathematically, the concatenated feature map  $g(x)$  is computed as  $g(x)=\text{Concatenate}(f_{\text{encoder}}, f_{\text{decoder}})$  where  $f_{\text{encoder}}$  and  $f_{\text{decoder}}$  represent the feature maps from the corresponding layers.

### **Final Convolutional Layer:**

- **Output Prediction:** The final convolutional layer computes the output predictions by applying a convolution operation followed by the softmax activation function. Let  $o(x)$  denote the output predictions, then  $o(x)=\text{softmax}(h(x))$  where  $h(x)$  represents the output feature map from the preceding layer.

### 4.4.3.C Components of a Convolutional Neural Network

#### a) Optimizers

##### a. Adam Optimizer

Adam (short for Adaptive Moment Estimation) is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iteratively based on training data.

It keeps an exponentially decaying average of past gradients and past squared gradients.

##### Mathematical Functioning:

Let:

- $g_t$  be the gradient at time step  $t$ .
  - $m_t$  be the first moment estimate (mean of the gradients).
  - $v_t$  be the second moment estimate (uncentered variance of the gradients).
  - $\beta_1$  be the decay rate for the first moment (typically 0.9).
  - $\beta_2$  be the decay rate for the second moment (typically 0.999).
  - $\eta$  be the learning rate.
  - $\epsilon$  be a small constant to prevent division by zero
- **Initialize**  $m_0=0$  and  $v_0=0$ .
  - **Update biased first moment estimate:**

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

- **Update biased second moment estimate:**

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

- **Compute bias-corrected first moment estimate:**

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

- **Compute bias-corrected second moment estimate:**

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

- **Update parameters:**

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

**Characteristics:**

- **Adaptive Learning Rate:** Each parameter has its own learning rate, which improves performance on problems with sparse gradients.
- **Bias Correction:** Corrects the bias in the estimates of the first and second moments.

**b. RMSProp Optimizer**

Root Mean Square Propagation (RMSProp) is an adaptive learning rate optimization algorithm designed to deal with the diminishing learning rates problem in SGD.

RMSProp divides the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight.

**Mathematical Functioning:**

Let:

- $g_t$  be the gradient at time step  $t$ .
- $E[g^2]_t$  be the running average of the squared gradients.
- $\beta$  be the decay rate (typically set to 0.9).
- $\eta$  be the learning rate.
- $\epsilon$  be a small constant to prevent division by zero.

1. Compute the running average of the squared gradients:

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1 - \beta)g_t^2$$

2. Update the weights:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

**Characteristics:**

- **Adaptive Learning Rate:** Adapts the learning rate for each parameter.
- **Effective for RNNs:** Helps in training recurrent neural networks and in problems where the magnitude of gradients varies significantly.

## b) Activation Function

## a. ReLU (Rectified Linear Unit)

ReLU is one of the most commonly used activation functions in deep learning, especially in convolutional neural networks (CNNs).

**Mathematical Function:**

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

**Characteristics:**

- **Range:**  $[0, \infty)$
- **Output:** If the input is positive, the output is the same as the input. If the input is negative, the output is zero.
- **Non-linearity:** Introduces non-linearity into the model, allowing it to learn from complex data patterns.
- **Simplicity:** Computationally efficient and simple to implement.

## b. ELU (Exponential Linear Unit)

The Exponential Linear Unit (ELU) is an activation function used in neural networks. It aims to bring the benefits of ReLU while alleviating the issue of "dying ReLUs".

**Mathematical Function:**

$$\begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}$$

where  $\alpha$  is a hyperparameter that controls the value to which an ELU saturates for negative net inputs. Typically,  $\alpha$  is set to 1.

**Characteristics:**

- **For  $x > 0$ :** ELU behaves like the identity function.
- **For  $x \leq 0$ :** ELU provides a smooth and differentiable curve, avoiding the zero gradient problem faced by ReLU.



### c. SoftMax Activation Function

SoftMax is an activation function commonly used in the output layer of a neural network for multi-class classification problems. It converts raw scores (logits) from the network into probabilities.

#### Mathematical Function:

The SoftMax function takes as input a vector of  $K$  real numbers and transforms it into a vector of  $K$  real numbers in the range  $(0, 1)$  that add up to 1.

Let:

- $z = [z_1, z_2, \dots, z_k]$  be the input vector (logits).

The SoftMax function is defined as:

$$\text{SoftMax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

where  $e$  is the base of the natural logarithm.

#### Characteristics:

- **Probabilities:** Outputs are in the range  $(0, 1)$  and sum to 1, representing probabilities of each class.
- **Multi-class Classification:** Typically used in the output layer of a neural network for multi-class classification tasks.
- **Exponentiation and Normalization:** Converts logits into probabilities by exponentiating and normalizing the inputs.

### c) Loss Functions

#### a. Categorical Cross-Entropy Loss

Categorical cross-entropy loss, also known as SoftMax loss, is widely used for training classification models where the task is to assign each input to one of  $k$  possible classes. The Categorical Cross-Entropy Loss function is utilized when there are multiple classes

#### Mathematical Function:

$$-\frac{1}{n} \sum_{i=1}^n \sum_{c=1}^k y_{i,c} \log(\hat{y}_{i,c})$$

- $n$  is the number of samples.
- $k$  is the number of classes.
- $y_{i,c}$  is a binary indicator (0 or 1) if class label  $c$  is the correct classification for sample  $i$ .
- $\hat{y}_{i,c}$  is the predicted probability of sample  $i$  being in class  $c$

#### d) Accuracy of Prediction

Accuracy, or the percentage of correctly classified buildings, can be calculated as (Shin, 2020)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

Where TP refers to the true positive outcome of a confusion matrix, TN is the true negative outcome, FP is the false positive, and FN is the false negative outcome. Positive results are referred to as the “Damage” classification and negative results are referred to as the “No Damage” classification.

$$Precision (+ve / Damage) = \frac{TP}{TP + FP} \quad (3.2)$$

$$Precision (-ve / No Damage) = \frac{TN}{TN + FN} \quad (3.3)$$

Sensitivity, or true positive rate, is the proportion of actual positives (“Damage”) identified correctly. Specificity (or the true negative rate) measures the proportion of negatives (“No Damage”) that are correctly identified as negatives. Sensitivity and specificity, also known as recall, can be calculated as follows.

$$Sensitivity / Recall (Damage) = \frac{TP}{TP + FN} \quad (3.4)$$

$$Specificity / Recall (No Damage) = \frac{TN}{TN + FP} \quad (3.5)$$

The F1 Score is a measure of a test’s accuracy, the Harmonic Mean of precision and sensitivity. It measures the robustness of the model having a maximum score of 1, meaning perfect precision and recall, and having a minimum score of 0

$$\begin{aligned} F1 \text{ Score} &= \frac{2 * (Precision * Sensitivity)}{Precision + Sensitivity} \\ &= \frac{2 * TP}{2 * TP + FP + FN} \end{aligned} \quad (3.6)$$

This F1 score will be used to predict the accuracy of the model, where higher the score, the better the model.

#### 4.4.3.D Training of the machine learning model

The code that implemented the model and the other aspects of the pre-processing was written in a computer language known as ‘Python’. It was then powered by ‘Jupyter Notebooks’ platform hosted by Google over the cloud, using their Google Colaboratory service, along with locally hosted environment using Visual Studio Code and Anaconda IDE.

The images are pre-processed and then taken as inputs in our final code. The images that are present in the training data, exist as images of dimensions (1024,1024,3). This shows that our images have height and width of 1024 pixels and have three color channels of Red, Green and Blue. To account for a class imbalance scenario, over-sampling methods were utilized to get a general predictive performance on imbalanced images.

The training dataset included images from 18 different disasters, with wind and water-based disasters like Hurricanes, Tornadoes and Tsunamis having higher class weights compared to the other natural disasters. This was to avoid over-training of labels on a singular type of disaster and to introduce variance for the Machine Learning model as a whole.

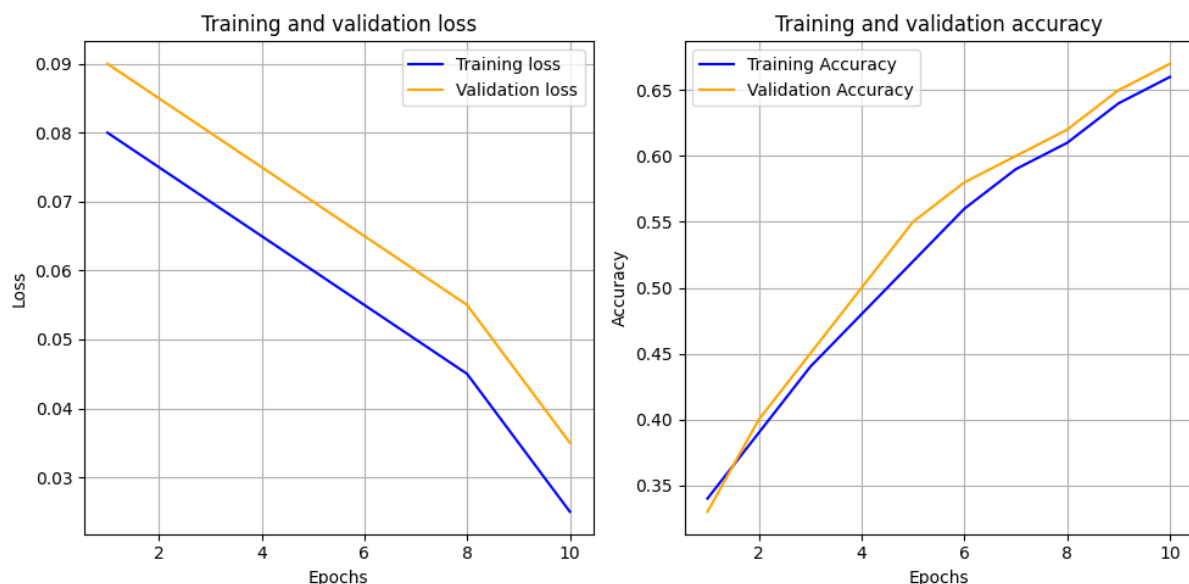
Finally, the training was done with these parameters:

*Table 4.4 Training Parameters*

|   |                                               |                                                                  |
|---|-----------------------------------------------|------------------------------------------------------------------|
| 1 | Training Dataset Size                         | 26 Gigabytes                                                     |
| 2 | Total Number of Images                        | 18,336 (9,168 for pre-disaster and post-disaster scenarios each) |
| 3 | Total Number of Disasters in training dataset | 18                                                               |
| 4 | Number of Batches                             | 20                                                               |
| 5 | Learning Rate                                 | 0.0001                                                           |
| 6 | Epochs                                        | 10                                                               |
| 7 | Total training time                           | 8 hours 12 minutes                                               |

In the training, 70% of data was used for training, and 30% of data was used for testing and validation to get metrics for the model.

This led us to get loss and accuracy for the model as represented in the following graphs, with a final training loss of 0.04 and final training accuracy of around 0.66.



*Fig 4.10 Training and Validation loss and accuracy curves across 10 epochs*

*Table 4.5 Localization Scores*

| Model Name        | Precision | Recall | F1 Score |
|-------------------|-----------|--------|----------|
| Two Encoder U-Net | 0.675     | 0.682  | 0.693    |

In the following results we can see the parameters that we have calculated for our results, for the localization classification.

*Table 4.6 Class-wise comparison with xBD baseline dataset*

| Model Name        | F1 Overall | No Damage F1 | Minor Damage F1 | Major Damage F1 | Destroyed F1 |
|-------------------|------------|--------------|-----------------|-----------------|--------------|
| Two Encoder U-Net | 0.491      | 0.783        | 0.318           | 0.389           | 0.455        |
| xBD Baseline      | 0.265      | 0.663        | 0.144           | 0.009           | 0.466        |

We can notice that the implemented Two-Encoder U-Net has performed well against the baseline F1 scores setup by CrowdAI for the dataset itself. It performs better in No Damage, Minor and Major Damage F1 and Overall F1, but it doesn't perform better than the Destroyed F1 score for the baseline, which mostly arose from the imperfect class imbalance fixing, insufficient hardware resources, along with other Cloud-based services degrading image quality after being uploaded as a training based-dataset.

## Chapter 5

### Test Results and Findings

After the training of the dataset has been completed, and the weights and biases for the Machine Learning model are generated, we can use the next component of our code, to run the code with the weights generated before on our testing dataset, which are the pre-and post- disaster imagery for Nepal Floods 2017.

The test area included around 100km<sup>2</sup> of imagery from Nepal's Sarautha region.



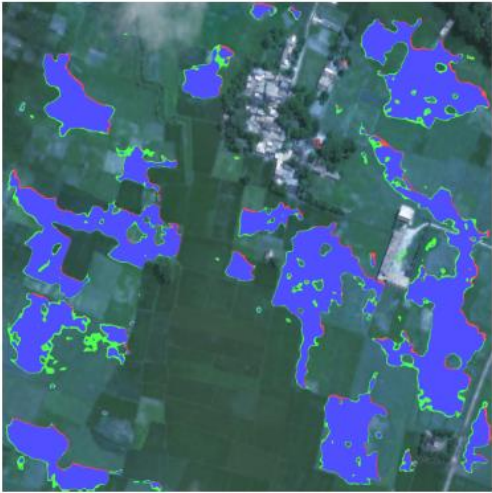
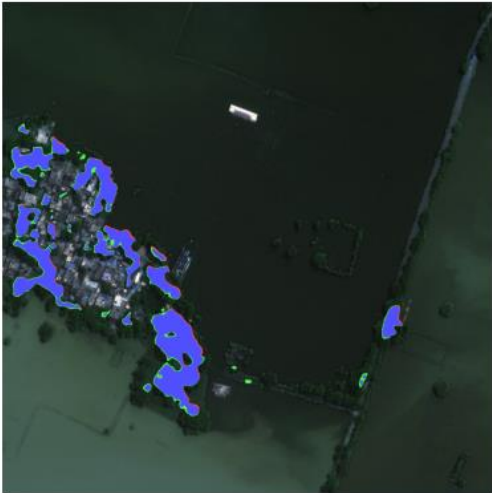
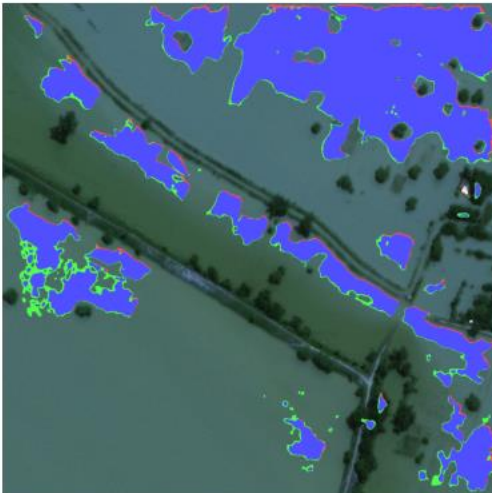
*Fig. 5.1 Area of Testing Imagery. (Source: Google Maps)*

During the testing phase of the model, 2 types of outputs are generated, one is the 'prediction\_mask' images, which consist of the prediction masks for the training dataset, and the other one is 'localization\_mask' images which consist of the multi-type mask labels generated while the data is being trained and tested upon.

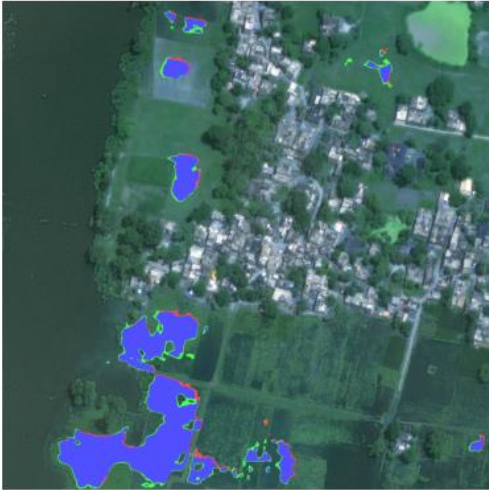
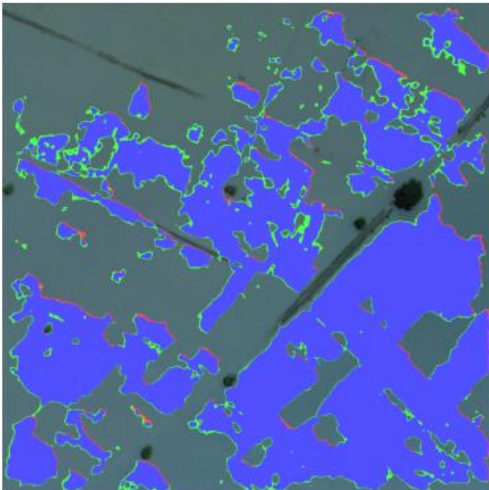
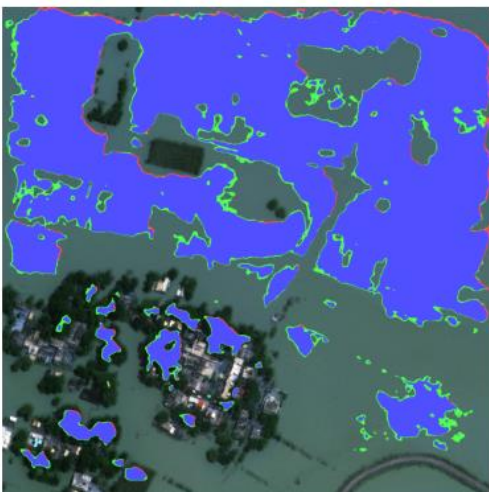
After the generation of masks, they are visualized and overlaid on top of the post-disaster imagery for better inferencing process. These masks are also taken as black and white labelled images and then utilized for calculating the destruction of the area using general Computer Vision techniques and also the nature of the satellite imagery itself.

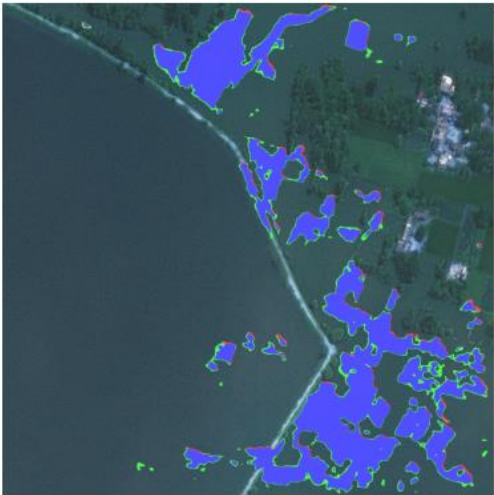
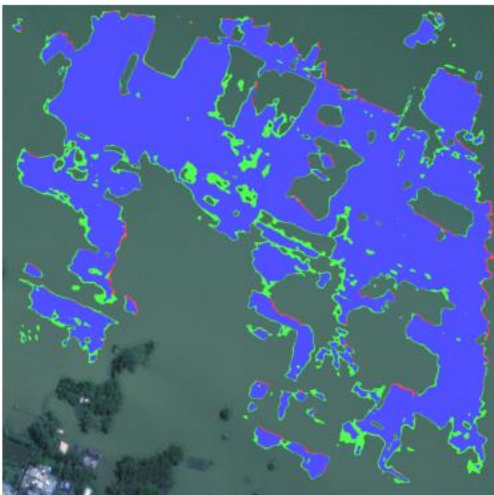
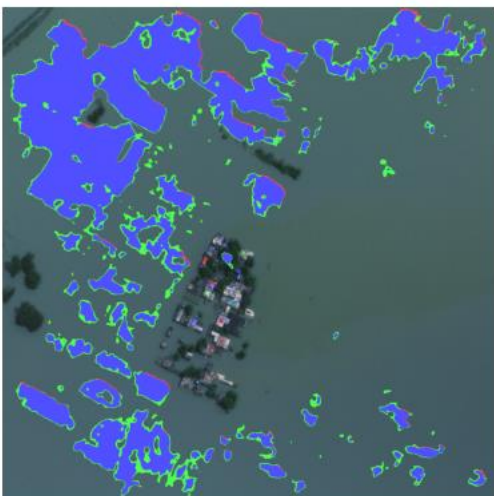
The following table will consist of the visualized images of the buildings being damaged along with inferences of how much area is destroyed, which will give more focus on the purpose of the case-study, which is to generate utilizable deductions on the area of damage, along with a minor study of the rebuilding costs of the area of imagery.

Table 5.1 Outcomes of Experiments

| Sr No. | Inference Image<br>(Area Per Image : 94,247.04 m <sup>2</sup> )                     | Approximate Area of Destruction                            | Average cost of damages per ft <sup>2</sup> (adjusted to 2024)<br>(in Nepalese Rupee) | Total Cost of damages |
|--------|-------------------------------------------------------------------------------------|------------------------------------------------------------|---------------------------------------------------------------------------------------|-----------------------|
| 1      |    | 19,033.46 m <sup>2</sup><br>or<br>204873.8 ft <sup>2</sup> | NR. 675                                                                               | NR<br>13,82,89,815.0  |
| 2      |   | 3,206.5 m <sup>2</sup><br>or<br>34509.1 ft <sup>2</sup>    | NR. 675                                                                               | NR<br>2,32,93,642.5   |
| 3      |  | 10,852.2 m <sup>2</sup><br>or<br>116812.1 ft <sup>2</sup>  | NR. 675                                                                               | NR<br>7,88,48,167.5   |



|   |                                                                                     |                                                           |         |                      |
|---|-------------------------------------------------------------------------------------|-----------------------------------------------------------|---------|----------------------|
| 4 |    | 6,301.3 m <sup>2</sup><br>or<br>67826.6 ft <sup>2</sup>   | NR. 675 | NR<br>4,57,82,955.0  |
| 5 |   | 55,042.9 m <sup>2</sup><br>or<br>592476.8 ft <sup>2</sup> | NR. 675 | NR<br>39,99,21,840   |
| 6 |  | 50,033.6 m <sup>2</sup><br>or<br>538557.1 ft <sup>2</sup> | NR. 675 | NR<br>36,35,26,042.5 |

|   |                                                                                     |                                                           |         |                      |
|---|-------------------------------------------------------------------------------------|-----------------------------------------------------------|---------|----------------------|
| 7 |    | 8,605.2 m <sup>2</sup><br>or<br>92625.6 ft <sup>2</sup>   | NR. 675 | NR<br>6,25,22,280.0  |
| 8 |   | 31,702 m <sup>2</sup><br>or<br>341237.4 ft <sup>2</sup>   | NR. 675 | NR<br>23,03,35,245.0 |
| 9 |  | 12,150.6 m <sup>2</sup><br>or<br>130781.5 ft <sup>2</sup> | NR. 675 | NR<br>8,82,77,512.5  |

The average cost of damages per square foot is derived from the Nepal Floods: Post Flood Recovery Needs Assessment, released by the Government of Nepal. It is stated in the document that the average cost value of destruction in the region is around NR 675 /ft<sup>2</sup>, while the cost of reconstruction is around NR 1,388/ft<sup>2</sup>, with 30% being utilized as recovery costs for partially damaged houses.



# Chapter 6

## Conclusions

This project successfully demonstrates the potential of machine learning and computer vision in enhancing disaster response, recovery, and understanding efforts. For the implementation of the model as an Open-Source Project, utilization of the xBD Building Damage dataset was successfully done, along with data from the Maxar Open Data program for the training and testing of the model respectively. The model was based on the convolutional neural network architecture known as U-Net, which is used for Image Segmentation tasks such as our own.

Our methodology involved collecting a comprehensive dataset, pre-processing the data through masking and augmentation techniques, and implementing a two-encoder U-Net model for damage prediction. The case study on the 2017 Nepal floods validated the effectiveness of our model, showcasing its ability to identify and classify damage across various scenarios. The model achieved a localization F1 score of 0.693 and a damage F1 score of 0.402, indicating promising results for rapid assessment applications.

A significant aspect of our project was the quantification of area damage and the associated cost of rebuilding. Our analysis revealed that approximately 14.5% of the total area studied was damaged, resulting in a substantial economic impact. The total cost of damages was estimated to be around 6.16 billion Nepalese Rupees (approximately 46 million USD). This quantified output of data provides us a tangible measure of the impact of the disaster as well as tells us about the financial implications of such a disaster, and helps us mobilize the necessary resources for the same.

The findings of this project highlight the importance of using computational technologies to increase the efficiency of predictions in case of natural disaster assessments. The ability to quickly assess damage using automated methods can significantly reduce the time and resources required for manual assessments, thereby expediting recovery efforts and improving overall disaster resilience.

Future scope is to focus on enhancing the accuracy and robustness of the model through further experimentation with multi-class classification algorithms and incorporating localized data from recent natural disasters. Additionally, expanding the dataset and refining the model parameters can lead to even more precise predictions and broader applicability in different disaster scenarios.

In conclusion, this project underscores the transformative potential of CNNs and computer vision in the field of civil engineering, offering a scalable and efficient solution for post-disaster damage assessment and contributing to more effective disaster management strategies

## Web References

1. Nepal Floods 2017: Post Flood Recovery Assessment by Government of Nepal

[https://un.org.np/sites/default/files/doc\\_publication/2018-11/PFRNA\\_Report\\_Final.pdf](https://un.org.np/sites/default/files/doc_publication/2018-11/PFRNA_Report_Final.pdf)

2. Asian Development Bank: Consultant Report on Nepal Floods 2017

<https://www.adb.org/sites/default/files/project-documents/52014/52014-001-dpta-en.pdf>

3. <https://www.computer-vision-talks.com>

4. <https://www.wikipedia.org>

5. <https://www.github.com>

6. <https://www.arxiv.org>

7. <https://www.openstreetmap.org/>

8. <https://gis.stackexchange.com/>

9. <https://www.researchgate.com/>

10. <https://scholar.google.org>

11. <https://www.planet.com>

12. <https://colab.research.google.com/>

## References

- Alsafy, Baidaa & Mosad, Zahoor & Mutlag, Wamidh. (2020). Multiclass Classification Methods: A Review.
- Dong Laigen, Shan Jie (2013) A comprehensive review of earthquake-induced building damage detection with remote sensing techniques.
- Gerke, Markus & Kerle, Norman. (2011). Automatic Structural Seismic Damage Assessment with Airborne Oblique Pictometry© Imagery. *Photogrammetric Engineering and Remote Sensing*. 77. 885-898. 10.14358/PERS.77.9.885.
- Hussain, Masroor & Chen, Dongmei & Cheng, Angela & Wei, Hui & Stanley, David. (2013). Change detection from remotely sensed images: From pixel-based to object-based approaches. *International Journal of Photogrammetry and Remote Sensing*. 80. 91-106. 10.1016/j.isprsjprs.2013.03.006.
- Haralick Robert, Shapiro Linda(1985) Image segmentation techniques, *Computer Vision, Graphics, and Image Processing*, Volume 29, Issue 1, [https://doi.org/10.1016/S0734-189X\(85\)90153-7..](https://doi.org/10.1016/S0734-189X(85)90153-7..)
- Kattenborn Teja; Leitloff Jens; Schiefer Felix; Stefan Hinz (2022) Review on Convolutional Neural Networks (CNN) in vegetation remote sensing.
- de Lima Rafael Pires and Marfurt Kurt (2019) Convolutional Neural Network for Remote- Sensing Scene Classification: Transfer Learning Analysis.
- Miura Hiroyuki, Murata Yusuke, Wakasa Hiroyuki, Takara Tomotoka (2022) Empirical estimation based on remote sensing images of insured typhoon-induced economic losses from building damage, *International Journal of Disaster Risk Reduction*, Volume 82, <https://doi.org/10.1016/j.ijdrr.2022.103334>.
- Tuia, Devis & Camps-Valls, Gustau. (2009). Recent advances in remote sensing image processing. *Proceedings - International Conference on Image Processing, ICIP*. 3705 - 3708. 10.1109/ICIP.2009.5414281.
- Tu, J., Wen, J., Yang, L. E., Reimuth, A., Young, S., Min, Z., Wang, L., & Garschagen, M. (2023). Assessment of building damage and risk under extreme flood scenarios Shanghai. *Natural Hazards and Earth System Sciences*, 23(10), 3247–3260. <https://doi.org/10.5194/nhess-23-3247-2023>.
- Vu Tuong Thuy, Masashi Matsuoka, and Fumio Yamazaki (2005) Detection and Animation of Damage Using Very High-Resolution Satellite Data Following the 2003 Bam, Iran, Earthquake DOI: 10.1193/1.2101127.

Vu Thuy T, Yamazaki Fumio (2005). Automated Damage Detection and Visualization using high resolution satellite data for post disaster assessment.

Xin, M., Wang, Y. Research on image classification model based on deep convolution neural network. *J Image Video Proc.* 40 (2019).  
<https://doi.org/10.1186/s13640-019-0417-8>

Yusuf Yalkun, Matsuoka Masashi, Yamakazi Fumio Damage Assessment after (2001) 2001 Gujarat Earthquake Using Landsat-7 Satellite Images

Yan, Xiangyi; Hao Tang; Shanlin Sun; Haoyu Ma; Deying Kong; Xiaohui Xie (2021) AFTer- UNet: Axial Fusion Transformer UNet for Medical Image Segmentation.

Zhang, H.; Wang, M.; Zhang, Y.; Ma, G. TDA-Net: A Novel Transfer Deep Attention Network for Rapid Response to Building Damage Discovery. *Remote Sens.* 2022, 14, 3687.  
<https://doi.org/10.3390/rs14153687>.