Structural Crack Classification and Grading after Disaster: A Supervised Learning Approach

by

Zahin Zaima
20201147
Abid Hossain Ashik
20201162
Md Yasin
20201157

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Zahin Zaima
20201147

Abid Hossain Ashik
20201162

Md Yasin
20201157

Approval

The thesis/project titled "Structural crack classification and grading after disaster: A supervised learning approach" submitted by

1. Zahin Zaima (20201147)
2. Abid Hossain Ashik (20201162)
3. Md Yasin (20201157)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 20, 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Dept. of Computer Science and Engineering
Brac University

Co - Supervisor:
(Member)

Rafeed Rahman

Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson & Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

A structure can be cracked by various disastrous events (for example: flood, cyclone, volcanic eruption, earthquake, fire outbreak). When a disastrous event occurs in an area, a lot of structures get damaged within a very short time. To reduce the death toll, it's essential to analyse and classify the structural damage or cracks quickly after the impact. Especially, in dense cities like Dhaka, where millions of structures have been built without following any organized plan, the damage can be unimaginable. Keeping this in mind, the first step after the event should be locating the severely damaged areas quickly and starting rescue operations by prioritizing the level of damage. Researchers conducted much research to identify the damage scale and classify structural cracks using machine learning algorithms. In most of their research, they considered a visual representation of the structure, structural parameters (for example: age, materials quality, strength, etc.), soil quality, magnitude, and so on. Considering these parameters is very important for accurate and precise prediction. However, collecting these types of data is a lengthy process. For that, their methods fail to provide a quick assessment. Therefore, this paper aims to classify the structural cracks and provide a quick assessment by considering only the visual representation of the damaged structure. Additionally, it implements various machine learning (for example: SVM, Decision Tree, KNN, RF etc) and deep learning algorithms (for example: VGG16, VGG19, ViT, ADA-ViT, D-ViT, etc). It also analyses and compares the performance of those models. Finally, this study proposes an architecture that can bring the highest accuracy (98.1%) among all the models that were implemented. Furthermore, in this architecture we have introduced a new approach which is we have considered both initial and damaged visual representation of a structure while analysing the damage grade. For annotating the dataset, this study follows EMS-98 (European Macro-Seismic Scale -98) standard.

Keywords: Quick Damage Assessment, Visual Representation, Machine Learning, Deep Learning, VGG16, VGG19, ViT, ADA-ViT, D-ViT, SVM, Decision Tree, KNN, RF, Accuracy.

Dedication

This thesis is dedicated to the citizens of Bangladesh as they will have to suffer a lot if a seismic event occurs and our department's respectable faculties have always encouraged, motivated, supported, and pushed us beyond the end line.

Acknowledgement

Firstly, all praise to almighty Allah for whom we have been able to complete our thesis works despite going through a very difficult time.

Secondly, we want to thank our supervisor Dr. Md Golam Rabiul Alam sir and co-supervisor Rafeed Rahman sir of the Department of Computer Science and Engineering at BRAC University for their continuous support and advice in our work. Their valuable guidance, support and effort have motivated us to finish our thesis successfully.

Thirdly, we are grateful to our parents who ensured that we got the best educational facilities from one of the best universities in Bangladesh. Their prayers, support, and encouragement helped us to reach this point.

Finally, last but not least, we would like to thank our friends who were always there when we needed mental support and refreshments.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$ADA - ViT$ Adaptive Vision Transformer

ANN Artificial Neural Network

BPE Byte Pair Encoding

CNN Convolutional Neural Network

$D - ViT$ Dynamic Vision Transformer

DNN Deep Neural Network

DT Decision Tree

$dVAE$ Discrete Variational Autoencoder

$EMS - 98$ European Macroseismic Scale

ESC European Seismological Commission

FC Fully Connected

IMS Macro Seismic Scales

KNN K-Nearest Neighbor Algorithm

ML Machine Learning

NPL Natural Language Processing

RF Random Forest

VGG Visual Geometry Group

ViT Vision Transformers

Chapter 1

Introduction

1.1 Overview

In recent years, disastrous events have become one of the biggest problems for human beings. There are various types of disastrous events such as earthquakes, tsunamis, cyclones, etc. Among these events, the most dangerous and unpredictable is the volcanic eruption and earthquake. According to a report by Al Jazeera, in 2023 more than 50000 people were killed and around 0.87 million people were injured in Turkey and Syria by a devastating earthquake [1]. Moreover, according to a report of The Daily Star, Dhaka city is extremely vulnerable in the face of powerful earthquakes and more than 2 million people will die if an earthquake of 6.9 magnitude hits [2]. A lot of study is required since seismic events are still a danger for regions affected by geological activity and they are densely populated urban areas. This research supports the formulation of plans that protect people and property in land areas by addressing the immediate issues connected to earthquake prediction, preparation, and reaction. According to a joint 2009 survey done by Jica and the Integrated Comprehensive Disaster Management Programme (CDMP), 72,000 buildings in Bangladesh's major cities, including Dhaka, Chittagong, and Sylhet, would collapse and 135,000 would receive damage in the event of an earthquake of magnitude 7 or greater [3]. Here, this research investigates the use of machine learning (ML) methods for the grading of damage or cracked buildings and what necessary steps should be taken based on the damage level. We also believe that by comparing the same building's initial image and cracked image, it will provide the strong analysis of structural damage grade prediction.

Using these initial and damaged-datasets opens the door to analyse the damage grade from a new angle. Using all this information for grading damage, the urban development organization of an area can identify the structural cracks and damages quickly just after the event. In summary, if we could introduce a quick crack detection method, it will be beneficial for damage assessment.

1.2 Motivation

The motivation behind this thesis is to develop an effective architecture by which we will be able to detect structural cracks and find out damage grade quickly using only the visual representation of a structure. Other researchers often rely on post-event surveys or pre-event surveys, which can be time-consuming and may not be able to provide real-time necessary steps. Using both initial and cracked structural visual representation that is affected by a disastrous event, creates a strong observation to analyse the damage grade. Although traditional techniques frequently depend on surveys performed either before or after an event, we believe that using structural visual data of both initial and damaged state improves our ability to more accurately evaluate the impact of seismic events. By analysing the both conditions of a building, we can also predict how much damage can occur to a building during the various types of disastrous events. This act will help to quickly predict the future losses in residential and commercial structures. Furthermore, responsible organizations can take proper steps to arrange for repair of defective structures, which create a significant role to minimize the damage for an event. Also, risky buildings can be marked and people will become aware and damage will be decreased. Combining both pre-event and post-event data improves the traditional approaches and gives the better output of the prediction. This study expects to not only advance the traditional approaches of damage classification but also provide a new perspective for further research in this field.

1.3 Problem Statement

Nowadays, seismic events have become a regular incident in the whole world. According to The Daily Star, more than 15 earthquakes were recorded in Bangladesh last year [2]. Moreover, people from all over the world witness at least one devastating disastrous event each year. This sudden impact of an event causes severe damage to the structures of that particular area (for example: collapsed buildings, cracked structures, damaged roads etc). Therefore, to minimize the damage and death toll, we have to quickly analyse and evaluate, which area is most affected, perform rescue operation accordingly.

One solution can be human based damage analysis. It will require highly trained expats with years of experience in this field. Though it is an effective way to detect the level of structural damage, it will require a lot of time and the result might vary from one to another. However, this process will be slower, and it will require many human resources. But to reduce the damage, we need a quick damage assessment. To overcome this problem, a lot of research has been done. They tried to classify the structural damage using different types of machine learning algorithms. This [4] paper proposes image based crack detection by using convolutional neural networks. In another paper [5], researchers have developed a machine learning framework to evaluate post earthquake structural safety. They have collected a robust data set of damaged buildings. Their data set contains multiple structural parameters (for example: building age, materials quality, strength etc.), soil quality, magnitude, visual representation and so on. Furthermore, They successfully achieved high precision and accuracy. However, we have found a drawback which is if we focus on high

accuracy, our data set should have multiple parameters. But collecting this type of data will take a lot of time. For that, if we follow a similar approach, we will not be able to provide a quick assessment.

Moreover, we have analysed the previous workings in this field and found out all of them worked on the visual representation and data of post disastrous events. We have found another problem in this method. If we consider only the damaged visual data of a structure as the data and evaluate, it might produce a one-sided result, as we did not consider the initial state of the structure when the structure was fine. Therefore, they left behind an important parameter which might be able to open up another angle of this research.

Therefore, the question this paper is trying to answer is, what if we only consider the visual representation of the structure? Will this approach produce reliable accuracy and precision? What if we analyse both the pre and post-structural state and then predict the damage level? Will this combined approach outperform the previous methods of damage analysis and crack detection?

1.4 Research Objectives

This research aims to develop a combined approach to analyse the structural damage. The objectives of this research are,

- Proposing a technique that will provide quick damage assessment instantly by providing the damage grade of a structure with good accuracy and precision.
- As mentioned above, none of the research considered the post event structural state, we would like to consider the initial state and see how the result varies.
- Creating a dataset that contains both initial and post-visual representations of the damaged structure.
- Providing a comparative analysis by applying different machine learning and deep learning algorithms and finding out which model performs the best on our small dataset.
- Optimizing the performance of the best-performed model so that it can be more reliable and accurate.
- Creating an interface to classify the damage and generate a damage map.

Chapter 2

Related Work

This study [6] describes the evaluation of earthquake resistance of urban buildings using image processing and machine learning techniques. They developed an automated decision support system that takes the image of a building along with basic information as input. The system then outputs whether the building is at risk and requires structural evaluation. They used CNN-based deep learning models. They have used the FEMA P-154 report as the baseline of the model. The main goal was to automate the process of rapid visual screening using machine learning. The maximum accuracy was 71%. They mentioned that because of the small data set the accuracy score was low. They have manually collected the data from the various areas of Dhaka city. The data set is not given.

This study [7] describes the damage to buildings caused by earthquakes using machine learning techniques. This paper presents the level of damage prediction to buildings caused by the Gorkha Earthquake in Nepal. They have implemented Neural Network and Random Forest machine learning approaches. The predictions have been made based on mathematically calculated eight tectonic indicators and past vibration records. They have collected the data set from the Driven Data competition platform [can't find]. Each row of the data set contains a specific building and there are 38 columns which were used as the features. Among them, important features are (i) Land surface condition (ii) Foundation type (iii) Roof type (iv) Ground floor type (v) Position (vi) Plan configuration (vii) Legal ownership status. They have trained both models and compared them. The NN model gave 62% accuracy score and the RF model gave 74% accuracy score.

This study [8] describes Earthquake building damage detection based on synthetic-aperture-radar imagery and machine learning. The paper describes the damage prediction of four earthquakes such as 2015 Gorkha earthquake, 2017 Puebla earthquake, 2020 Puerto Rico earthquake, 2020 Zagreb earthquake. We use a random forest ML classification model. Both multi-class and binary damage classification are attempted and we compare the predictions with actual results. As input we use OpenStreetMap as building footprints and other dataset for instance Ground shaking intensity maps (ShakeMaps), SAR-derived damage proxy maps for detecting damage of buildings in the affected area by an earthquake by comparing pre and post-event InSAR images. The system then outputs whether the building is damaged or not and compares the balanced accuracy of the multi-class damage classification

with binary damage classification. The main goal of this paper is to show multi-class damage grade classification using InSAR data for detection of earthquake building damage which was rarely used previously. When multi-class damage classification was attempted, the balanced accuracy score was 0.23 for the 2017 Puebla earthquake, 0.36 for the 2015 Gorkha earthquake and 0.40 for both the 2020 Zagreb and Puerto Rico earthquakes. However, the balanced accuracy scores improved significantly when binary damage classification was used. The balanced accuracy score is 0.65 for the 2017 Puebla and 2020 Zagreb earthquakes, 0.72 for the 2020 Puerto Rico earthquake, and 0.82 for the 2015 Gorkha earthquake. Limited training data, imbalanced class distributions impact the balanced accuracy scores in multi-class damage classification. Data augmentation, and model fine-tuning reduce these challenges and improve the balanced accuracy scores in multi-class damage classification.

Another study [9] describes Object-based classification of earthquake damage from high-resolution optical imagery using machine learning.The paper describes the damage prediction of post-event aerial imagery for the 2011 earthquake in Christchurch, New Zealand. As input they take the images of buildings to predict the earthquake damage. The system then gives outputs by identifying the different classes from a picture. The paper considered five classes that are able to cover all the land in an image. Those classes are building, pavement, vehicle, vegetation, damage. To level those classes from a picture they use pixel-based classifier and object-based classifier. The main goal of this paper is to analyze the efficiency of class segmentation and classification, and compare various multistep image segmentation levels to make a valid comparison between pixel-based classification and objective-based classification by observing results from the same imagery, same training and validation data. In Pixel-based classification they use the SVM classifier. Pixel-based classification system's overall accuracy of 62% and in object-based approach it improved to 77% overall accuracy. object-based approach gave better accuracy because in object-based approach a systematic approach is used in evaluating object-based image segmentation. Using scale parameters of 20, 50, 100, and 200, the systematic approach compared four levels. They used a naive Bayes classifier for level four and SVM for level two for achieving optimal results. Object-based approaches do not always give better results than pixel-based methods, as a systematic approach is required to ensure optimal classifier parameter selection.

Furthermore the study [10] describes Multi-Resolution Feature Fusion for Image Classification of Building Damages with Convolutional Neural Networks. As an input they use satellite and airborne(manned, unmanned) imagery to perform the image classification of building damage detection. The system then outputs which portion of the image is damaged. In the paper there were three multi-resolution CNN feature fusion approaches (MR a, MR b, MR c) that were proposed and tested against two baseline (mono-resolution) methods. The main goal was to understand which type of image data improved the output damage accuracy. The better output is seen in the baseline ft and MR c networks. When they use multi-resolution feature fusion methods, the satellite and aerial (unmanned) resolutions improve the image classification. In aerial (manned) resolution tests, multi-resolution feature fusion systems gave lower accuracy than the baseline method. In this scenario, the optimum approach was to fine-tune a network that trained with generic aerial

(manned) image samples.

The study [11] describes a process to classifying earthquake damage grades of RC buildings rapidly. They categorized the damage of the RC building from four different earthquakes such as Ecuador, Haiti, Nepal, and Pohang earthquake. This paper goal was to investigate some ML techniques on and compare each technique's efficiency. They used several machine learning algorithms, such as KNN, RF(Random forest), SVM and ET(Extra tree) and categorized the RC buildings based on their vulnerability. Total eight features were used in this study. Features are total floor area, no of floor, column area, area, concrete wall area, concrete wall, masonry wall area, captive columns, etc. Among the machine learning algorithms the ET(Extra tree), RF(Random forest) perform well.

Another study [12] describes sampling and machine learning methods for a rapid earthquake loss assessment system. The study proposed ML models to pre-earthquake damage state detection. The main goal of this paper is to create an accurate machine learning model for quickly assessing earthquake losses in residential houses at the municipality level. Using the 2010 Kraljevo M5.4 earthquake dataset Two representative sampling strategies(random or clustered) and three ML algorithms such as Decision Trees (DT), Neural Networks (NN), and Random Forest (RF), are compared. After monitoring damage states on a pre-earthquake representative set of houses, the total repair cost may be estimated with less than 20% error, with a probability of 0.71 for the 2010 M5.4 Kraljevo, Serbia earthquake. The features of this study are six such as building type(BT), construction year, footprint area, number of floors, and x and y geographic coordinates (spatial coordinates). Clustered sampling was better than random sampling for all ML methods (DT,NN,RF). Because the K-Means clustering algorithm is used to improve the proposed sampling approach for selecting representative houses (training set). Future Geological data, data about the earthquake can improve the accuracy in predicting the total expected repair cost of the damaged building.

The paper [13] is about a machine learning damage prediction model for the 2017 Puebla-Morelos, Mexico, earthquake.This paper focused on the development of a machine learning based damaged prediction model. Total of 4 algorithms were used in this paper to check the classification capabilities. Logistic regression, SVM, decision trees, and random forest algorithms are suitable to perform supervised classification tasks.Firstly, in this paper the author preprocesses the features using one-hot encoding. After preprocessing the data, the author selects four models and training data where splitting ratio is 75% is training and 25% is testing purpose. The accuracy is 65% in LR, 61% in SVM, 67% in Decision tree, and 67% in random forest. Here the Decision tree and Random Forest model gives the highest prediction accuracy. Since we used post hoc methods, random forest allowed us relatively good outcomes.

This paper [14] is about earthquake damage prediction using machine learning. This paper talks about how dangerous calamities like earthquakes can be detected during the earthquake so that the damage can be measured using Machine Learning techniques. KNN, Decision Tree, Random Forest, SVM,XGBoost, Neural Network

Lightgbm are the most used classification algorithms which had been used for this paper. These classification techniques are used to define building grade level. These grade levels are the measure parameters of the risk factor that had been caused by the earthquake.The positive side of the paper is that it can measure the building vulnerability by grade level of the buildings that may help to reduce the lives cost and destruction of houses. But the negative side of the paper is that the algorithm is not generalized for different seismic conditions. Moreover, although the building damage level can be predicted, no information about accuracy is given in the paper. This paper [15] is about Modeling Earthquake Damage Grade Level Prediction using

Machine Learning and Deep Learning Techniques.This paper dealing with modeling the earthquake damage grade for predict the damage of the earthquake.The damage measured using machine learning and deep learning techniques such as Random Forest Classifier, Logistic Regression,KNN,Decision Tree, Artificial Neural Network techniques are most useful for modeling the earthquake damage grade level. Here Random Forest classification is the best fit algorithm for this research paper. The F1 score is 84.46% in the Random forest classification model.

This paper [16] is about application of edge detection techniques for concrete surface crack detection. This paper mainly focuses on finding the crack in a building for ensuring the structural health of a building.Since there are lots of methods available but here author used image processing technique for finding the cracks. To detect the crack, we need to collect the data such as images. The images should be high resolution pictures that will ensure the quality of the works. The proposed idea of the paper is to collect the data using a high resolution camera and process the images through segmentation and crack dimension calculating. The positive side of the paper is that it can detect the crack automatically because the authors used CNN and edge detection methods. On the other hand, if the picture's angles, resolution are not perfect, then the algorithm shows low accuracy.

This study [17] describes a study on the determination of damage levels in reinforced concrete structures for different earthquakes. In this study, five distinct earthquakes in Turkey were taken into consideration while classifying the damage to reinforced concrete structures using the European MacroSeismic Scale (EMS). Here five different levels of damage were found for the sample buildings based on the EMS. They follow the earthquakes are listed: 1999 D'uzce, 2003 Bing'ol, 2011 Van, 2020 Sivrice (Elazie), and 2020 Izmir. The measured peak ground acceleration values for each earthquake taken for use in this study were compared to the peak ground acceleration levels that are currently advised. Based on the earthquake level they focus on different ground motion levels. Within the parameters of this study, five distinct damage ratings for reinforced concrete structures on the European Macroseismic Scale -98 (EMS-98) were used to calculate earthquake damages for various earthquakes.

E. Isik et al. [18] conducted research to determine and manage all the information about the building's damages after the earthquake. They mainly focused on quick damage assessment immediately after the earthquake. They used visual representation of damaged structures to classify the damage grade. To label the visual data,

they have used EMS-98 (European Macro-Seismic Scale -98) standard. Similarly, the study of S. Okada and N. Takai [19], presented some methods of visual structural labeling precisely within short time. They proposed AIJ (Architectural Institute of Japan) standards for RC building and EMS-98 (European Macro-Seismic Scale -98) for Masonry buildings.

Chapter 3

Model Architecture

3.1 CNN Architecture

In this research paper, we analyzed disaster damage grading with different CNN models such as: VGG16, VGG19, also used transfer learning models such as: ViT, D-ViT, AdaViT and some machine learning classification algorithm such as: SVM, DT, KNN. Convolutional neural network models also transfer learning models are the part of deep learning models. Artificial Neural Networks perform wonderfully in Machine Learning. Neural networks are used in several datasets, including image, text, audio datasets. There are several uses for different kinds of neural networks. For instance, Recurrent Neural Networks—more specifically, an LSTM—are used to predict word sequences. For image classification we can use CNN. In our research paper, we want a classifier crack or damage grade based on the initial and damage image. For this reason, we used different CNN models such as: VGG16 and VGG19. One kind of Deep Learning neural network design that is frequently utilized in computer vision is the convolutional neural network (CNN). For dealing with images and other visual data, "Computer vision" is the area of artificial intelligence that allows it [20].

CNN is a collection of pooling and convolutional layers that let you build feature maps by accumulating and removing the most crucial elements of a picture. Like edges, textures, and patterns properties and characteristics are retrieved by using convolution layers and filters. Via max-pooling layers ,the feature map is reduced in size. These layers work together to give CNNs the ability to learn hierarchical data representations, which is essential for their performance in computer vision tasks [21]. In order to extract features from the input image, the convolutional layer applies filters also To save time in processing, the Pooling layer downsampled the picture and finally, a fully connected layer gives a final prediction [20].

3.2 Convolutional Layers

Convolutional neural networks, often known as CNNs, are a part of deep learning models that are often used to identify patterns in images. Convolution layers serve as CNNs' fundamental building blocks and convolution is a key mathematical procedure carried out by these layers [22]. In addition, they have roles in many sectors such as: signal processing, computer vision, natural language processing, spatial

data analysis. FEATURE EXTRACTION is Convolutional Layers' primary goal. The key network processes in this particular kind of artificial neural network use one kind of term named "convolution". To identify the features that are present across an image, such as edges, CNN uses filters which are often called kernels. Four primary functions are performed in CNN which are Classification (Fully Connected Layer), Pooling or Sub Sampling, Non Linearity (ReLU), and Convolution. A convolutional layer is always the first layer of a CNN model [23]. When we give an image as an input of CNN model, it first applies the convolution and this image has length, breadth (the image's dimensions), and height (the channel, since most images have red, green, and blue channels). Consider applying a small neural network which is also known as a filter or kernel where K outputs on a small patch of this image data and represents them vertically.
If we want to do convolution on a 34×34×3 sized image. Filter sizes can be a×a×3, with "a" being any value like 3 ,5, 7, but smaller than the image's dimensions. The filters/kernels are smaller matrices, typically measuring 2 x 2, 3 x 3, or 5 x 5. In the convolution layer, it calculates the dot product between the kernel weight and the matching input image patch after sliding across the input image data. For instance, for a $5 \times 5 \times 3$ image the single filter size is 3×3. We calculate the total of the highlighted pixel values for each input channel, weighted by the corresponding convolution kernel values [24]. A CNN can learn more complicated features from the input picture or video by using additional convolution layers. Simple features like corners and edges are learned by the first convolution layer. More complex elements, including shapes and objects, are learned by the deeper convolution layers[25].

3.2.1 Pooling layer

Several convolution and pooling layers placed one after the other is a common CNN model architecture. As part of the pooling operation, a two-dimensional filter is slid over each feature map channel which then summarizes the features that fall under the filter's coverage area. As a result, it reduces the quantity of calculation carried out in the network and the number of parameters to learn. Polling layer's main objective is dimensionality reduction. Consider that we wish to reduce the size of a large image while preserving all of its essential features, such as colors and edges. Here, the pooling layer works independently on each input depth slice. Using a window that is slid over the input data and the Max pooling or average pooling of the numbers, it resizes it across regions [26]. A pooling layer of two 2x2 value-added channels with two down samples is the optimal setup. Here all depths, widths, and statutes are cut in half, removing 75% of the decrees.

3.2.2 Fully connected layer

Fully Connected (FC) layer, also known as a dense layer. Because of its full linkage, it is called "fully connected".
The final output predictions are produced by fully connected layers, which are usually located near the end of the neural network architecture. While fully connected layers are more flexible and may be used for any type of data, convolutional layers are particularly useful for spatial data, like images, where locality and translation invariance are crucial. CNN deep learning architectures in use combine the two

kinds of layers: convolution layer and fully connected layer. To extract and learn features, convolution layers are typically used in the earlier stages where to make predictions based on these features fully connected layers are often used at the end of the architecture [27].

3.2.3 Activation Functions

Activation functions are an integral building block of neural networks. After determining the neuron's activation, activation functions are used to convert the node's input signal into the output signal. Three types of activation functions can be defined: binary, linear, and non-linear activation functions [28]. In our study we are using only linear activation function which is ReLU activation and Soft max activation.

3.2.4 Rectified Linear Unit (ReLU)

ReLU is a piecewise linear function. It is also called rectified linear function. For positive input, it will send it to the output directly, otherwise, it will output zero. In the hidden layers, the model generally uses the ReLU as the activation function. Because with this activation function, the training process becomes faster [29]. The activation function for the Rectified Linear Unit (ReLU) is as follows:

$$f(x) = \max(0, x) \tag{3.1}$$

It solves the problem of vanishing gradient, in contrast to TanH (another activation function), it has no backpropagation error and a low activation rate of 50%.

3.3 Transformer Model

Another kind of deep learning model that was released in 2017 is the transformer model. These models have been used for a variety of machine learning and artificial intelligence tasks, and they have quickly established themselves as foundational concepts in natural language processing (NLP) [30]. As a transfer learning model we used ViT, D-ViT, AdaViT. There are two primary innovations that transformer models bring to the table. Consider these two innovations within the context of predicting text which is Positional encoding and Self-attention. The encoder-decoder concept is the primary core feature of the Transformers architecture. Also, Multi-Head Attention, Converting in patches, Flatten patches, Linear Projection, Tokenization, Embedding, Positional encoding, Transformer Encoder, etc are the core steps to understand the architecture of Transformer Models working procedure. The encoder-decoder structure is used in the transformer model architecture. Before transformers were invented, neural networks had to be trained using massive, costly, labeled datasets. Transformers reduce that necessity by mathematically identifying patterns between pieces, making available the trillions of photos and petabytes of text data stored in corporate systems and on the web.

3.3.1 Tokenization

The process of transforming input data into corresponding tokens is known as tokenization. If it is text as an input then it uses byte pair encoding(BPE) for transforming into a token. If it is imaged as a picture first divided into non-overlapping patches and then linearly projected to obtain a token. When processing input speech the spoken signal is transformed into a spectrogram which is handled like an image input and tokenized similarly to how images are tokenized [31]. 1 patch size is =(16 * 16). And (16 * 16) = 256 small parts which are called tokens. In this case, 1 patch= 256 tokens = 1D patch. By breaking down the image into smaller patches and by converting them into a sequence of tokens the transformer model can process and understand the different parts of the image separately. Discrete variational autoencoder (dVAE) trained tokenizer and it is done by the tokenizer itself. The tokenizer and decoder are the two sections of the dVAE, which is taught to reconstruct an image. Generating tokens from the image pixels is the responsibility of the tokenizer [32].

3.3.2 Input Embedding

In the Transformer architecture, the input embedding is essential since it transforms input tokens into vectors with a specific dimension. We apply learned embeddings to convert each token to a high-dimensional space (vector). In essence, every vector contains information like syntactic and semantic information about the token. The numeric representation of an image token into a vector representation. If our input data is an image, the embedding compresses the visual data complexity into a compact form. Higher-level semantic information like objects are captured by providing semantic information of image embeddings. Image embeddings encode meaningful features of an image and understand the content of an image. For detecting an object, image classification, semantic information is essential [33].

3.3.3 Positional Encoding

The following stage is to combine all of the vectors into a single vector. For combining several vectors into a single vector the most popular method is to add one vector component with another vector component. For instance, the technique of summing the vectors (of length 3), [0,1,2], and [3,4,5], is [0+3, 1+4, 2+5], or [3, 5, 7]. The outcome remains the same even if we add the same vectors' integer values in a different order. To make it easy to understand, I will explain it in text input. If the input sentence is "I'm Zaima, He's Ashik" and "I'm Ashik, He's Zaima" this line has the same words but the arrangement is different, it will produce the same summation vector. As a result, we must determine a method that would provide us with a different vector for each of the two different statements. To do this the transformer model introduces positional encoding. Positional encoding consists of adding a position to the input embeddings for the reason of providing information about each token in the sequence [34]. Also in image input, for each input embedding there is also mention of the position of the token [35]. Here we mention positional encoding equations for a token.

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{3.2}$$

$$\mathrm{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{3.3}$$

3.3.4 Transformer Block

A very large neural network that has been specifically trained to predict the next word in a sentence. Such a big neural network can be trained by including an essential step which is the attention component. Here we could significantly improve it. There are two main components in the Transformer block. One is the attention component and the other is the feedforward component. Every feedforward network block gets the attention component. For each input embedding, it goes to several transformer encoder blocks and an attention component is added to each of these blocks [35].

3.3.5 Multi-Head Self-Attention Mechanism

This Self-Attention is the key component of the Transformer. The first layer of the transformer encoder is the Multi-Head Self-Attention. Thus, Multiple self-attention operations are performed in parallel. So we call it multi-head self-attention where each self attention operation focuses on learning different aspects of the relationship between tokens. Self-attention allows each token (word, patch) to gather information from other patches. It can capture the dependency between patches (for image input) or word (for text input). Mainly it helps the model to figure out which words are important to each other and how they relate to one another. For example, we have two sentences: Money in the bank, The bank of the river. Here the word "bank" has two different meanings of two sentences. For image data this confusion also rises which patch means what and how patches are related to each other. So the attention mechanism is very important in transformer models [36].

For each token the model makes sure that self-attention mechanism will perform. By performing self-attention and knowing the relationship between tokens the model calculates the similarity score. To calculate similarity scores(attention score) there are three parameters for each token. One token has to do three different tasks: query, key, value.

- Query: It is a vector that stands in for a particular token(patch or word) from the input sequence. Query is looking for other words to pay attention to.
- Key: is also a vector in the self-attention mechanism. For every word or token in the input sequence, a key serves as a vector in the attention mechanism. Key like a word being looked at by another word.
- Value: Every value has a corresponding key, and the self-attention layer's output is generated using these values. Value is like the information or meaning of a patch or word.

A score matrix is produced by calculating a dot product matrix with multiplication between the queries and keys. Self attention layer calculates the similarity score for each word/patch query with all words/patches key by calculating attention calculation and attention weight. When a query and a key match well, then it means they have a high attention score or whose words are more related to each other. These attention scores are used to determine how much attention each word should give to other words in the sentence [37]. To determine the attention weights, the softmax function is used to adjust scores. An output vector is generated by multiplying the value vector with the softmax function. Overall in self-attention mechanism, we calculate the query, key, value for each token. With the multiplication of query and keys, the model calculates attention score calculation and attention weight. Then with multiplication with attention weight and values(vector) of tokens we get the output vector.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.4}$$

$$\text{MultiHead}(Q, K, V) = [\text{head}_1, \ldots, \text{head}_h]\, W_0 \tag{3.5}$$

$$\text{where head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{3.6}$$

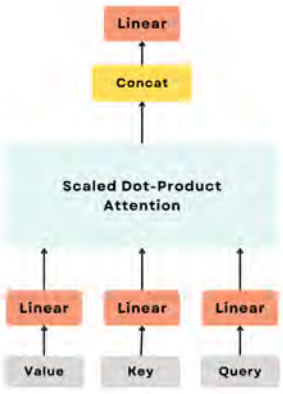

Figure 3.1: Self-Attention Layer Architecture

3.3.6 Feed Forward

In simple terms, a feed-forward network is a multi-layered structure where information moves from input to output in a single direction. Data is processed in a straight line by feed-forward networks. Using a feed-forward network to convert the output of self-attention mechanism into the model's final output. Although it can also be applied to various types of tasks and models. For instance, in CNN a feed-forward network is used to convert the convolutional layers output into the final output of the model [38].

3.3.7 The Softmax Function

A softmax function is calculated in self-attention mechanism to compute attention weights. in this layer converts all the attention scores into probabilities(that add to 1). We get the greatest scores corresponding to the highest probabilities. We can then select a sample for the next word from these possibilities. [35]. Softmax function's equation is:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.7}$$

3.3.8 Encoder-Decoder

The transformer model uses encoder-decoder architecture. If it is an input text and our intention is language translation, where the transformer model takes the input text and output in another language text. The encoder, typically an LSTM, transforms a sequence of tokens into a vector and a further LSTM called the decoder converts the vector into a series of tokens. Examples of the transformer encoder-decoder architecture in operation are Facebook's M2M-100, a huge multilingual machine translation model that can translate between 100 different languages, and Google Translate, which employs the T5 model to translate text between languages [39].

3.4 Existing Deep Learning and Machine Learning Models

In our thesis, we have implemented several existing deep-learning models and machine learning classifier algorithms such as VGG16, VGG19, ViT, ADA-ViT, D-ViT, SVM, DT, KNN, RF. Details architecture informations are discussed below.

3.4.1 VGG16

It is a model of convolutional neural network. It has an input layer, several hidden layers and an output layer [22]. VGG16 architecture is used for image classification with deep learning which was trained on 1.2 million images out of which can label different things into one of the 1000 categories that are present in ILSVRC-2012 ImageNet dataset [40]. It has in total 16 layers. Among them, 13 layers are convolutional layers and 3 layers are danse layer. Convolutional layers that analyze the image and dense layers that make the final decision on what the image is. The layers of the VGG16 model are organized into blocks. These blocks contain multiple convolutional layers and a max-pooling layer [41]. This model is already pre-trained using a large dataset(ImageNet). With 92.7% accuracy, VGG16 can classify 1000 images with 100 different categories [42]. We have images of 224x224 size. For that, we set VGG16 model's input size to 224x224x3. This input image is passed through multiple block of convolution and pooling layers, and features are extracted each time. Finally, it goes through danse and soft max layer and predicts the damage grade.

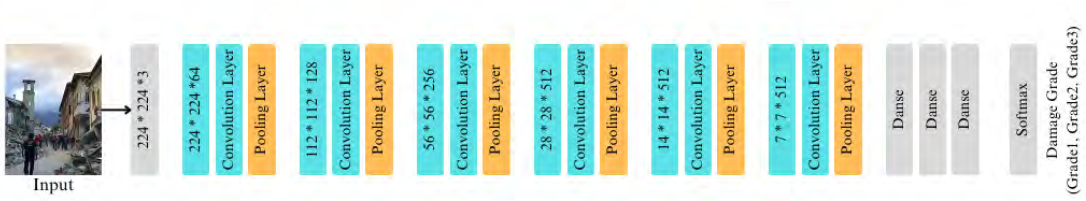

Figure 3.2: VGG16 Model Architecture

3.4.2 VGG19

VGG19 is also a convolutional neural network (CNN) model. The VGG19 model shares the same fundamental concept as the VGG16 model, with the exception that It supports 19 layers [43]. It has in total 16 convolution layers which are organized into 5 separate blocks. VGG19's higher layer count can offer a slightly better accuracy than VGG16, particularly for datasets that call for the learning of deeper features [44].

3.4.3 ViT

The Vision Transformer (ViT) is a type of deep learning model that applies the Transformer architecture, originally designed for natural language processing (NLP), to computer vision tasks. As we use image input, to do tokenization we have to create patches of the image. Vision Transformer (ViT) uses self-attention. Self attention is a way for computers to understand the relationship between the different parts of the image. For example, a cat image in a self-attention computer can focus different parts of cats like its ears, eyes,and tails. Understand how they relate to each other and the form of the whole picture. The first step in ViT model is Split an image into patches where patches have fixed size (16*16).

Figure 3.3: Splitting Image to Patches

To simplicity we consider a smaller image size as an example and understand the patch calculation. A 48*48 sized image split into 16x16 sized patches and result is total of 4 patches. Number of patches= (48/16) * (48/16) = 3 * 3 = 9. The stride used is also 16. Stride means how many pixels the sliding window moves each time. As the stride size and the patch size (16*16 size) is the same, there will be no overlap between the patches.

Next the model does flatten the image patches. Instead of treating this patch as a picture, we want: computers will process a patch as a sequence of smaller elements called tokens. One patch size is =(16 * 16). And (16 * 16)=256 small parts which are called tokens are generated. After tokenization next the model does linear

projection of flattened patches. Transforming each 1D vector into a lower dimensional vector. Linear projection in a vision transformer will work on these flattened patches(tokens). We do Linear Projection by reducing the dimensionality of 1D vectors so that we can use less memory and less computational resources and do faster, extracting important features, reducing noise from the picture [45].

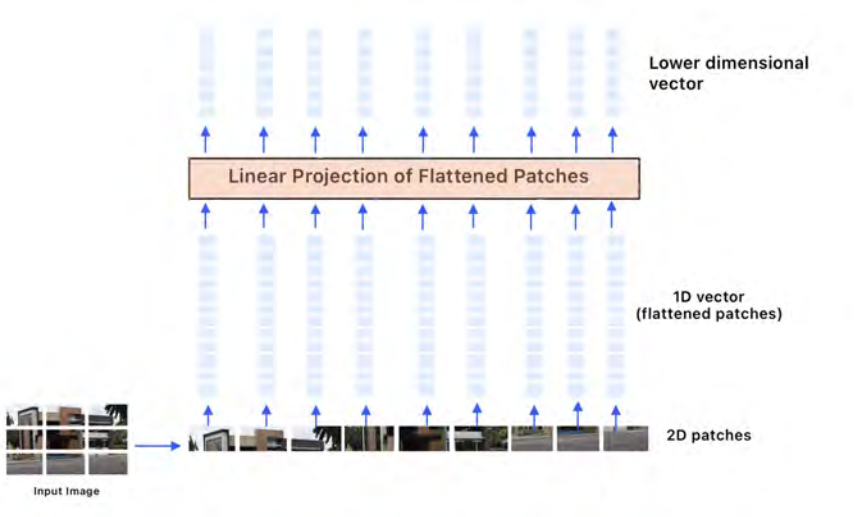

Figure 3.4: Linear Projection of Flattened Patches

After that, the ViT model goes to the transformer encoder where it includes positional embeddings, multi-head self-attention for each token. This work is the same as transformer model architecture for any dataset. After the self-attention layer we have a feed forward network and the output of each patch is passed through a feed forward neural network. This helps capture the linear relationships within the patches. After that we have a classification layer. The final layer of the transformer encoder is the classification head which maps the output of the transformer into the desired output format. For example it can be image Classification, object detection. The whole architecture of the ViT model is given below.
ViT is an encoder only transformer. Instead of decoder there is just an extra linear layer for final classification which is called MLP head. The absence of a decoder is one of the key differences between the ViT and the traditional transformer architecture. Natural processing task where we perform translations or text generation. Over there we need a decoder. Because the decoder component is used to generate output sequences based on the learned representation. Encoder-Only Architectures (like ViT) are suitable for tasks where you only need to understand the input to produce a simple output (like a class label) [46].

3.4.4 D-ViT

D-ViT is the extension of the Vision Transformer model. ViT model divides images into fixed-size patches and applies self-attention to process the patches. On the other hand, D-ViT introduces dynamic mechanisms to adapt the processing, according to input. Every patch in the input image is not treated the same by D-ViT like the ViT model. D-ViT dynamically determines how much attention each patch requires rather than processing them all. Here the level of depth and attention will be the

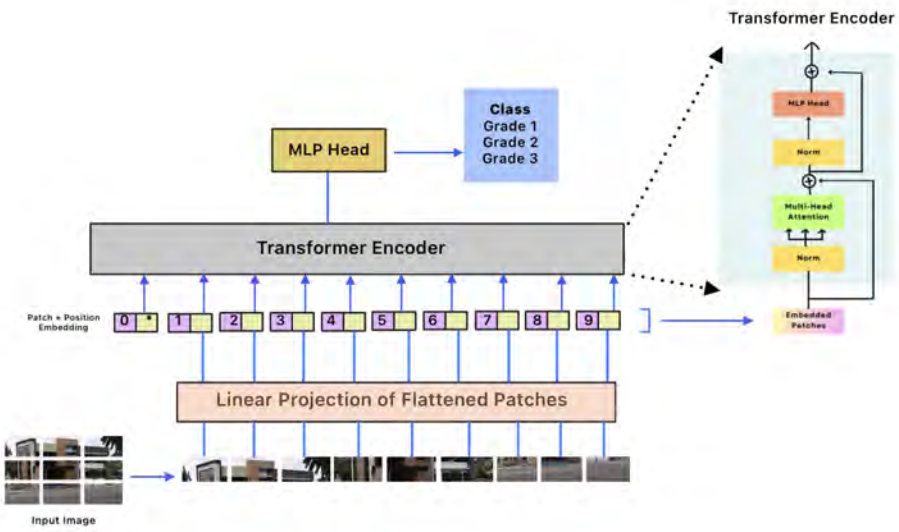

Figure 3.5: Vision Transformer (ViT) Architecture

same. Token pruning is one of D-ViT's main modifications where next layers of the transformer can skip or reduce less informative patches (tokens). The D-ViT model cascades multiple transformers to achieve the goal of setting correct token numbers based on the inputs. During testing, they are turned on one after the other until an impressive prediction is made or the final model is deduced [47].

3.4.5 ADA-ViT

ViT models give us very good results in images and video classification tasks. Vision Transformer models take an input image and split it into non-overlapping patches and transform them into input tokens. These tokens are passed through an encoder layer. In the classification head, ViT performs the ultimate classification. These tokens are very important because final prediction depends on these tokens. The ViT model's computational complexity is very high and that complexity gradually increases in respect to the number of input tokens. Adaptive Vision Transformer, adapt any changes that need for image classification for better performance [48]. In the Adaptive ViT model, it will dynamically adjust the attention mechanism. There might be adaptive layering for controlling the complexity [49]

3.4.6 SVM

It is a supervised machine learning algorithm which can be used for both classification and regression. Though it is best suited for classification. First, all the data points are plotted into a N-dimensional plane. Next, the algorithm identifies the optimal hyperplane that effectively separates data points into different classes. It tries to draw the hyperplane in a way that the margin between the closest points of different classes is maximized [50] Support vectors are the closest points of different classes from the hyperplane. Margin refers to the distance between the hyperplane and the support vectors. There is also a term called Kernel which is a mathematical function used to map input data into a higher dimensional plane.

3.4.7 DT

It is a flow chart-like structure and consists of multiple nodes. In this algorithm, nodes are of three types such as root node, internal node and leaf node. Root node is the first node that represents the initial decision to be made on the entire dataset. Each internal node represents a different decision and leaf nodes represent the final decision or prediction. There are also branches that connect the nodes and represent the outcome of the decision or test. Moreover, in this algorithm, different types of matrices (for example: gini impurity, entropy, etc.) are used to split the dataset into subsets [51]

3.4.8 KNN

It is also a simple supervised learning algorithms. It can handle both numeric and categorical data. First, all the data points are plotted into a 2D plane. These points form clusters of different classes. When a new data point comes, this algorithm calculates the distance between the point and the k-nearest neighbours. Then it performs voting for classification. The new data point is assigned to a class depending on the majority number of similar neighbours [52]. For calculating the distance we can use matrices such as Euclidean distance, Manhattan distance, etc. Depending on the value of nearest neighbour (K), prediction may vary.

3.4.9 RF

It is a tree-learning algorithm. It creates many decision trees in the training phase and each tree is created using a random subset of the data. This randomness characteristic introduces variability, reduces the risk of overfitting and increases the overall accuracy. In prediction, it considers the result of all trees and gives the prediction based on voting [53].

Chapter 4

Methodology

Every scientific researcher must follow the right procedure to complete the work accurately and on schedule. In our research paper we also follow a research methodology where it in short represent our working procedure.

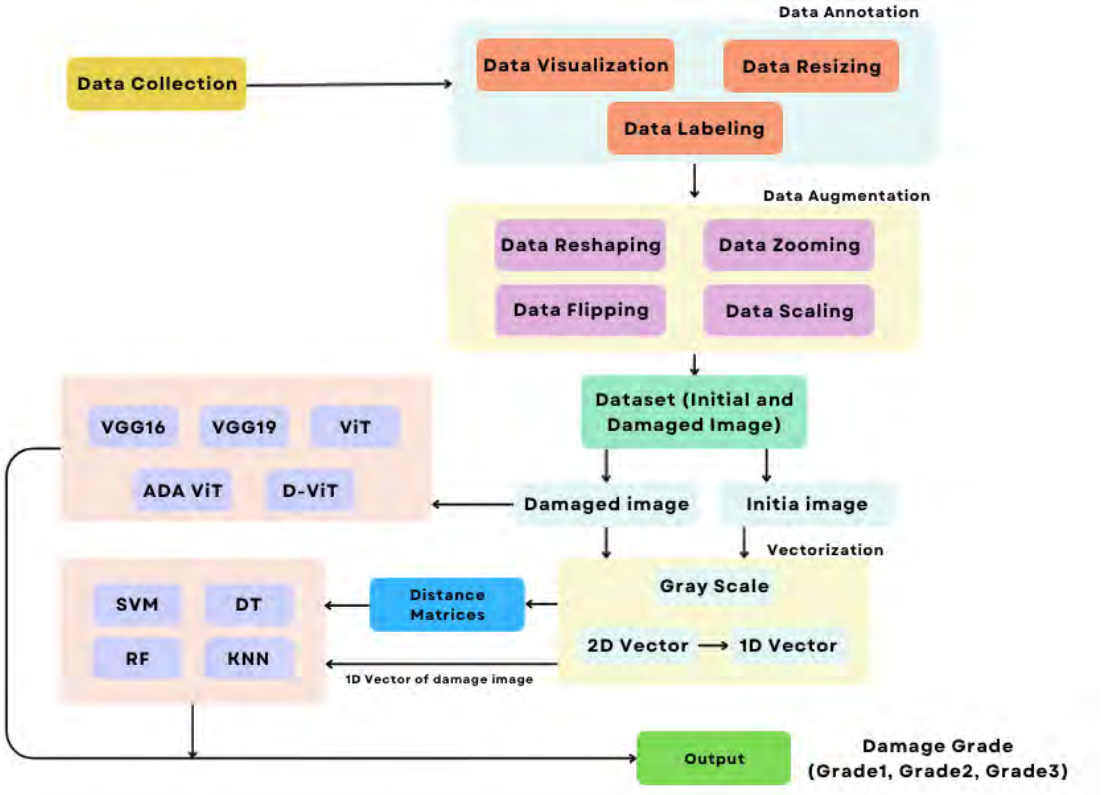

Figure 4.1: Top-level overview of proposed system

4.1 Data Collection

For our research purpose, we have decided to collect the previous and post images of cracked or damaged buildings for making our dataset. We need initial stage images and after disaster's images of the buildings because we considered both previous and post stages of the buildings.There is no body research on this type of work so that the dataset of our work is not available till now. We need to make our own dataset. Basically, the dataset is made of previous and post images of damaged buildings. We collect the images from the internet (Facebook, Instagram, X, Google Search). Since we need to compare the two state images, collecting the data is too challenging. For that reason, we can hardly collect 105 images for our dataset that extends further.

Figure 4.2: Collected Image

4.2 Grading Approach and Survey on Data

4.2.1 Introduction to Grading approach

Based on the eyewitness observations, there are many different international macro seismic scales(IMS) which can be used to quickly classify structural damage after an earthquake [54]. Compared to these earlier scales, one of the more recent ones created by the European Seismological Commission (ESC) has a wider range of damage levels and covers a greater variety of damage levels [18]. This modern scale is EMS-98. As we want to classify structural damage or cracks of various disastrous events such as: flood, cyclone, volcanic eruption, earthquake, fire outbreak, we can use this EMS-98 to categorize our grades.

EMS-98 is visual inspections and human reports and one of the main objectives is the detailed assessment by detecting the grade of damage to different types of buildings(especially RC, masonry structures). In building damage classification we can use EMS-98 and it defines five grades of damage for buildings which is Grade-1 damage to Grade-5 damage. The RC buildings' damage grading and the description of grade in EMS-98 is described below.

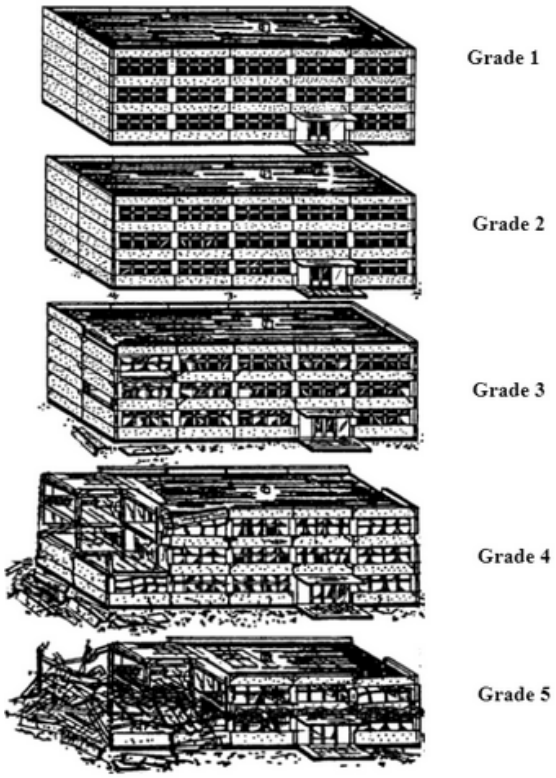

Figure 4.3: Classification of damage to buildings of RC [18]

Grade 1: No damage to slight damage (Hairline fine cracks in partitions and walls)
Grade 2: Moderate damage (some structural damage, more considerable non-structural damage) Small cracks in columns and beams; cracks on columns, additional long foundation walls. Cracked partitions and walls; deterioration of delicate cladding, trimming. Mortar will begin to fall out of the joints between wall panels.
Grade 3: Substantial to severe damage (Strong to heavy structural and non-structural damage) cracks in the elements of frames at level beams, columns, beam-column joints near base or at coupled wall junctions Concrete Cover Failure, Buckling of Steel rods. Cracks in full height partition and infill walls, failure of separate infill panels.
Grade 4: Very heavy damage (heavy structural damage and non-structural). Extensive failure or weakening of structures, very large cracking with compression concrete failures and rebar fracturing may occur. Beams reinforced bar bonds are also likely to fail. Partial walls, some, may collapse; A few columns will fall. One or two beams fail in the upper floors.
Grade 5: Destruction (very heavy structural damage)Collapse of the ground floor or parts of buildings [18].
For our own simplicity and running our models, we converted these 5 grades in EMS-98 into 3 grades. For example, we consider Grade 2 in our model, and it is the combination of Grade 2 and Grade 3 of EMS-98. That means the definition of Grade 2 and Grade 3 of EMS-98 is considered as a Grade 2 in our research.

- Our research Grade 1 = Used Grade 1 of EMS-98
- Our research Grade 2 = Used Grade 2 and Grade 3 EMS-98
- Our research Grade 3 = Used Grade 4 and Grade 5 EMS-98

Grade 1: No damage to slight damage (Hairline fine cracks in partitions and walls)
Grade 2: Moderate damage to severe damage (Strong to heavy structural and non-structural damage)
Grade 3: Very heavy damage to destruction (very large cracking with compression concrete failures and rebar fracturing may occur or collapse of the ground floor or parts of buildings) Damage pattern chart based on our research is as follows.

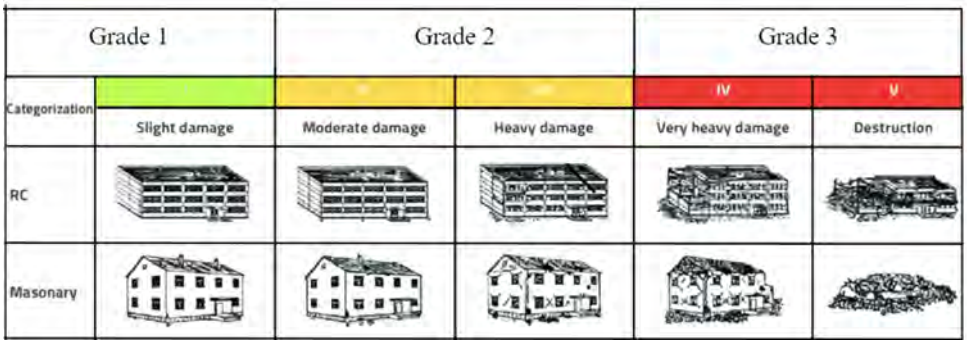

Figure 4.4: Damage Pattern Chart [19]

4.2.2 Survey on Data

After collecting the initial and damaged images of a structure, it's time to find ground truth. Therefore, we initiated our work by conducting a survey on a Google form. In this form we mentioned pre and post building images and people had to choose whether it was grade 1 or grade 2 or grade 3. We also describe in previously what is the meaning of grade 1, grade 2 and grade 3. Here we follow the EMS-98 grading definition to choose the grade. In this way we do labeling for all of our raw dataset. Participants labeled the image by understanding the grading system and compare the initial and damage image of a building by visual assumptions. The in-person survey was conducted with BRAC university students. Moreover, we selected those who are willing to a part of the survey. As we want to apply supervised learning models on our dataset, we need the labelled dataset. For this reason, surveys on data are an important part of our research. These labels serve our models as the ground truth for the training of our data. Our model will learn from this labelled dataset to make predictions about new unlabeled data. After the survey of labeling our dataset, we followed the most popular and efficient techniques to proceed our research work.

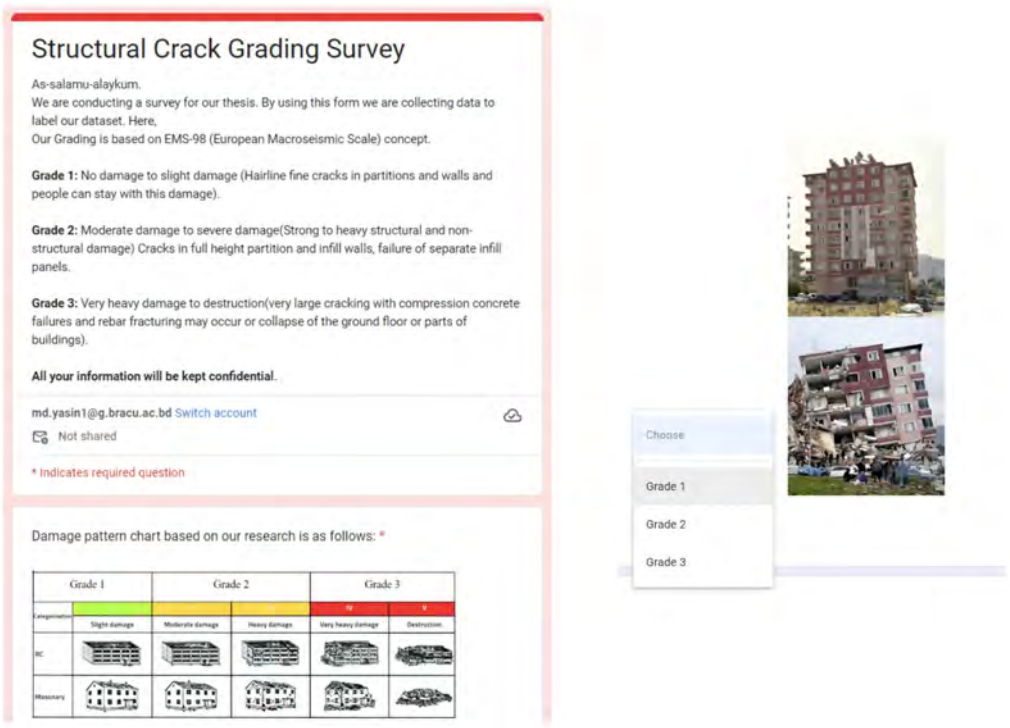

Figure 4.5: Survey on Structural Crack Grading

4.3 Data Annotation

We primarily annotate our dataset by human labeling to identify the building damage grade prediction based on visual structural changes. Using Google form, we conducted a survey. This survey allows the participants to give their opinions on the images and annotate the damage level by their observations. Here we again mention that they follow the grading based on the EMS-98 approach.

4.4 Data Reshaping

In this stage, we do data reshaping which means we reshape the collected images that were collected from various resources. We make a standard resolution dataset to run our model because different pixels of data can not run into one model. We adjust the pixel density, reduce lighting where needed, and make color variations where needed. These reshape datasets are more compatible with our desired model to detect previous and post-disaster building damage analysis.

4.5 Data Pre-processing

In this stage, we format the input images to do our research in a way that we can do our research work smoothly. Since we collected a set of images as our dataset from various resources, the image sizes are not the same. Some of the images are vertical, and some of the images are horizontal. First, we separated each image and made two sub-datasets named original and damaged. Then, we resize the images (224*224). However, we keep the image order same so that we can compare initial and damage image of a structure in later.

Figure 4.6: Sample Dataset

4.6 Data Generation

As we have mentioned above, we were able to collect only 105 images. Due to the lack of these types of combined data, we can't add more to our dataset. It is nearly impossible to apply machine learning algorithms on this dataset let alone deep learning algorithms. The reason is models which work on visual representation require a massive number of data to be trained well. Therefore, if we use our primary dataset, surely models will not get enough data that they require and

they will perform poorly. Keeping these in mind, we implemented traditional data augmentation techniques (for example: rotating, zooming, flipping, cropping, etc.) with the help of Keras library. We have generated only 8 images from each original image because if we generate many images from each it might increase the biasness. At the end of this step, our dataset has in total of 840 images.

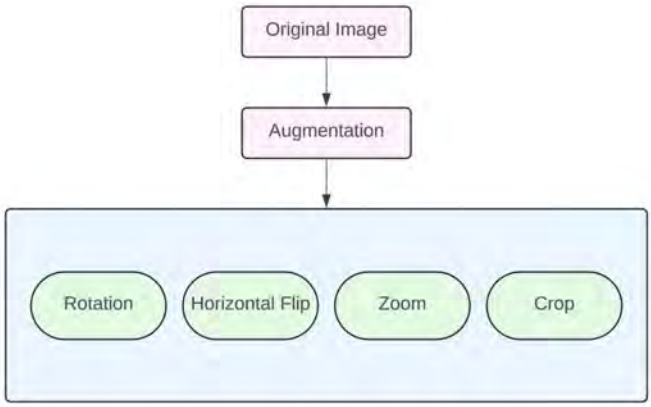

Figure 4.7: Augmentaion Process

Figure 4.8: Generated Images

4.7 Data Mounting

In this stage, we mount our dataset where we are comfortable to work. It could be Google collaboratory or Visual Studio Code to mount our dataset. In Google colaboratory, for mounting dataset we need to mount Google Drive by using some coding command or do this work manually. we use from google.colab import drive command to import the drive module for our research from Google collaboratory module. And then we use drive.mount("/content/drive") command to mount our google drive. After running this code, the Google collaboratory gets access to the datasets stored in Google Drive, giving us full access to model training [55]. In VS Code, we manually set up the dataset in a folder so that when we run the model the model finds the paths easily [56].

Chapter 5

Model Implementation

Model implementation is the process of carrying out our model's implementation strategy. When implementing a model, we have to start small. Since most machine learning effort is done on the data side, setting up a complete pipeline for an advanced model is more challenging than working on the model itself. After setting up our data pipeline and implementing a simple model with a small number of features, we can iterate on our model to make it more accurate [57].

5.1 Workflow Overview

Selecting the appropriate work pattern is necessary to getting the best results. Since our primary objective is to observe the output of models when we used only the post damage dataset, we used several models to run on the post damage dataset and determine which model or models could work best for this particular experiment. Finding the best model for our work, we will apply initial and damage image dataset and fine-tuning on that model. An outline of our workflow is provided below. Here are the details:

- Dataset is collected from various online platforms such as Facebook, Instagram, X, Google Search. Here we strongly made sure that we had to find out those building images as a dataset which has two images. One image is before the structural damage or crack and another is after the structural damage or crack image.

- As we want to apply supervised learning models on our dataset, we labeled the dataset by conducting a survey based on the concept of grading system in EMS-98 through a Google form.

- The dataset has been processed and cleaned. This includes labeling and augmenting.

- As finding initial and damage/crack images of a specific building during the disaster is hard to find, augmentation on our dataset is very important to run our models.

- Some deep learning models such as: VGG16, VGG19, ViT, D-ViT, ADA-ViT and some machine learning classifier algorithms such as: SVM, Decision Tree,

KNN, and Random Forest were used on our dataset. The Keras Application has been used to implement deep learning models. Pre-trained weights are provided along deep learning models known as Keras Applications. For fine-tuning, feature extraction, and prediction, this models also applicable [58]

- Comparing the models' output to see which one performs better. This part is just our own verification where we can understand our model working process is fine on our own dataset.

- Analyzing the best model and fine-tuning on that model by using initial and damage both image dataset.

5.2 Training Set

We have splitted our dataset into training and testing sections to help with our understanding of the characteristics of the model. We use our training dataset to train computer vision models and fit the machine learning models using supervised machine learning algorithms. We trained the model using 70% of the dataset from grades 1, 2, and 3.

5.3 Test Set

A section of the data that is made available but is "held back" and not used for model training is known as the test set. After the model has been trained, the test set's objective is to assess how well it performs on hypothetical data. The test set is used to assess the performance of the final model.It should be ensured during the splitting procedure that the test set's data distribution is reflective of the entire dataset [59].We used for test data using 30% of the dataset from grades 1, 2, and 3.

5.4 Train-Test Split

Before training the models, we have splitted the dataset into tainingset and testset. For training set we have picked 70% data and for testset we have picked 30% data from our original dataset.

Chapter 6

Challenges Faced

In our study, we have implemented several models to achieve our desired performance by considering initial and post-structural damage. During implementation, we had to face many challenges that we addressed perfectly and overcome the challenges. We are the pioneers here because there is no one who worked with combined structural representation (previous and post-structural) after a disaster. But there are many researchers who worked with post-structural damage to buildings [60]. There are many challenges we face such as data dependency, overfitting, problems during data collection, gradient vanishing, and the techniques we used to overcome these challenges. Every challenge we faced during the research period, we discussed everything below.

6.1 Data Dependency

CNN and Deep Learning are highly data dependent models. Because during the model training period, these models need high volume data to get desired results [61]. For analysis of a high volume of initial and damaged images are required. The accuracy and the reliability of the models depends on the quality and the quantity of data. Since the dataset is not available for our research, we built the dataset from scratch. Moreover, we were dependent on social media, internet for collecting the images that added more hassle to do our research.

6.2 Overfitting

The size of our dataset is small and has only 105 images, which occur overfitting. Overfitting happens in such a time that the model learns very well on the training dataset, but the model could not generalize new images that was the cause of poor performance [62]. To find the problem solution, different kinds of methods can be used, for example, augmentation [63].

6.3 Issues During Data Collection

Since the initial and damaged structures dataset was not available for grading, we were bound to be forced to collect the data from online platforms like Facebook, Instagram, X, and Google search. During the data collection procedure, we faced lots of challenges. Since we need both initial and damaged images, these types of images are hardly available on the internet. We hardly found 105 images. Those images fulfil our requirement for datasets. Since the images are found on the internet, we also consider the image quality like whether the resolution, and lighting of the image are right or not. Moreover, there were more images available that have written something on the image, so the structure was not visible perfectly.

6.4 How did we overcome?

We faced many problems during the research periods. To overcome the challenges, we followed some strategies. Let's discuss how we overcome the challenges below.

- To overcome data dependency and overfitting, we increased the number of datasets. We increased the number of images from 105 to 840 images by using data augmentation. We can increase the number of images using augmentation where the images are transformed in different scales, rotations and make adjustments in colour [64].

- We were very careful while dealing with the datasets. We collect relevant data during data collection. As we need to grade the structure, for proper grading, we need a balanced dataset where three types of images (grade 1, grade 2, grade 3) are presented equally. We did a survey to label our dataset to overcome the challenge of imbalanced datasets.

Overall, we were able to overcome the challenges that we faced during the research period, and we were able to reach our desired destination.

Chapter 7

Comparative Results and Proposed Customization

7.1 Results of Supervised Models

At the start of our research, we start work with very basic Deep Learning models to work with our dataset. Since we are the pioneer of our dataset. At first, we tested Deep Learning models on our dataset. We implemented models like VGG16, VGG19, ViT on the primary dataset (without augmentation) which have an accuracy of 36%, 43.21%, and 48.56% respectively. On the other hand, after the augmentation of our dataset, we found 63.41% in VGG16, 53% in VGG19, 54.47% in the ViT model.

Deep Learning Models		Accuracy
Without Augmentation	VGG16	36%
	VGG19	43.21%
	ViT	48.56%
With Augmentation	VGG16	63.41%
	VGG19	53%
	ViT	54.47%

Figure 7.1: Before Augmentation vs After Augmentataion

With Augmentation Deep Learning Models:
After performing the augmentation, all of the models performed well. As we mention the collection of our dataset before augmentation was 105 and after augmentation we increased it to 840. Before the augmentation, the data was not enough to train models properly. For this reason, the accuracy of the model was low. However, with augmentation, we got the better accuracy of all the models.

Deep Learning Models	Value Loss	Accuracy
ViT	0.9198	54.47%
D-ViT	0.7553	67.25%
ADA-ViT	0.7377	71.08%
VGG16	1.1740	63.41%
VGG19	0.6648	53%

Table 7.1: Results of Deep Learning Models

If we look into the bar chart, we can see that we used some deep learning models where ADA-VIT performed well because of its architecture. The ADA-VIT model's accuracy is 71.08% with a value loss of 0.7377.

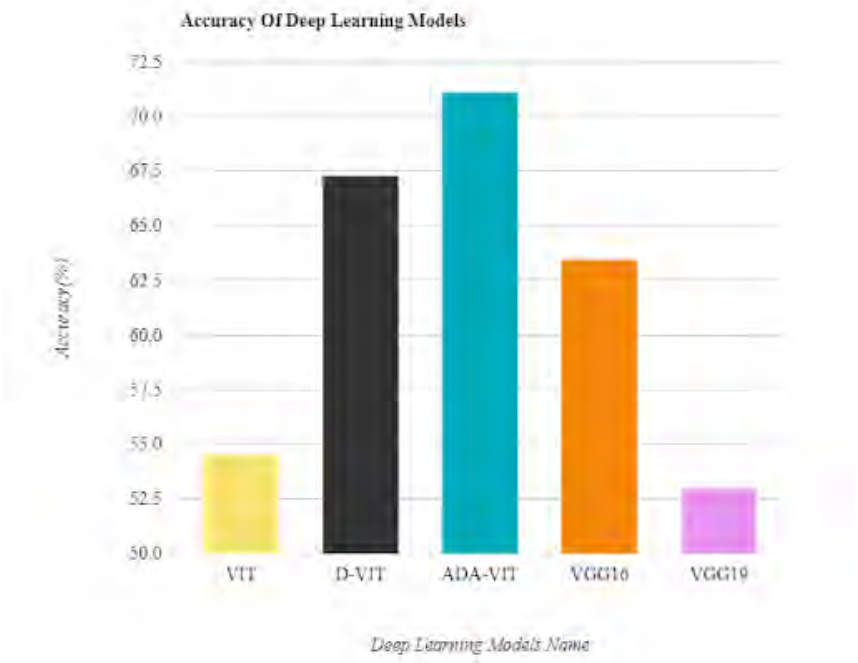

Figure 7.2: Deep Learning Models (image size:224 x 244) Accuracy

After performing Deep Learning models, we could not get our expected output. Because of that, we need to think a bit. We decided to shift to classification models to check whether we could find a better model than the Deep Learning models.

With Augmentation Classification Models: In classification models, we used the same size of input images. In the classification task, we used many models there to reach our destination. Here we used SVM, DT, KNN, RF. In the classification task, SVM performed better than others. With the augmented images,SVM gives 95.67% accuracy.

Classifier Models	Accuracy
SVM	95.67%
Decision Tree	81.01%
KNN	77.91%
Random Forest	93.42%

Table 7.2: Results of Classifier Models

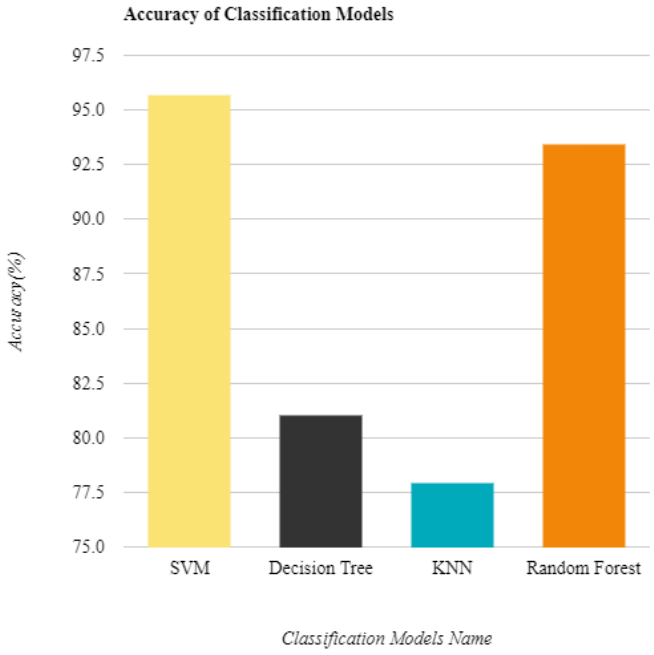

Figure 7.3: Accuracy of Classification Models(image size:224 x 244)

If we observe the Deep Learning models and Classification models, in deep learning models, the ADA-ViT performs better than other Deep Learning models. The accuracy of ADA-ViT is 71.08%. On the other hand, among machine learning classifier models, SVM model provided the highest accuracy of 95.67%. If we compare deep learning and machine learning models, SVM is the highest performer for our dataset. Therefore, we decided to experiment with SVM and customize it.

7.2 Analysis of Comparative Results

In this section, we will compare and analysis the performance of the implemented models. It is a good way to understand how our work is going on. As we mentioned before, previous researchers identified the damage scale and classified structural damages using machine learning algorithms. However, they considered the visual representation, parameters (for example: age, materials quality, strength, etc.), soil quality, magnitude, and so on for each structure. Some of them used aerial view images.

At first, we run our model only on the damaged dataset. Then, we compare our results with the results of the previous research works. We find out that, some of the models are performing well and sometimes better compared to them.
Finally, we implemented our customized model architecture of SVM, which gave the highest result on the damaged-image dataset. Additionally, in our customized model architecture, we used both initial and damaged-images as the data. We converted both images to 2D vectors using the OpenCV library and flattened and converted those images to 1D vectors. After that, we will take the necessary steps to obtain the research paper goal. Moreover, this study will try to provide an optimized solution to consider all the scenarios mentioned in this research paper.

7.2.1 Previous Research

Previously, there were few researchers who used on only the dataset of damaged buildings or aerial images of damaged buildings. Our dataset is different. We not only have the dataset of the damaged pictures but also the initial pictures of the same building. To compare with other research works, initially, we used only damaged pictures to train our models so that we can compare with them.

For example, in a paper [13] focused on the development of a machine learning based damaged prediction model where they used damaged images with parameters and applied Decision trees, Random Forest and SVM algorithms. Their accuracy is 61% in SVM, 67% in decision tree, and 67% in random forest. Whereas in the same concept, using only damage picture dataset our model accuracy is 95.67% in SVM, 81.01% in decision tree, and 93.42% in a random forest.

First, we are going to implement deep learning models and then machine learning classification algorithms with both initial and damaged picture dataset.

7.2.2 Models Comparison

Model Type	Model Name	Precision	recall	f1 -score	Accuracy
Deep Learning Model	VGG16	57%	54%	54%	63.41%
	VGG19	55%	46%	53%	53%
	ViT	53.18%	45.08%	43.80%	54.47%
	D-ViT	65.14%	62.93%	63.67%	67.25%
	ADA-ViT	79.95%	68.63%	75.35%	71.08%
Classification Model	SVM	96%	95%	96%	95.73%
	Decision Tree	78%	78%	78%	81.01%
	KNN	83%	75%	77%	77.91%
	Random Forest	92%	92%	93%	93.42%

Figure 7.4: All Model's Results

Among the Deep Learning models, ADA-ViT performed better and among classification models, SVM performed better. If we looked into the bar chart, SVM performed very well among all models overall.

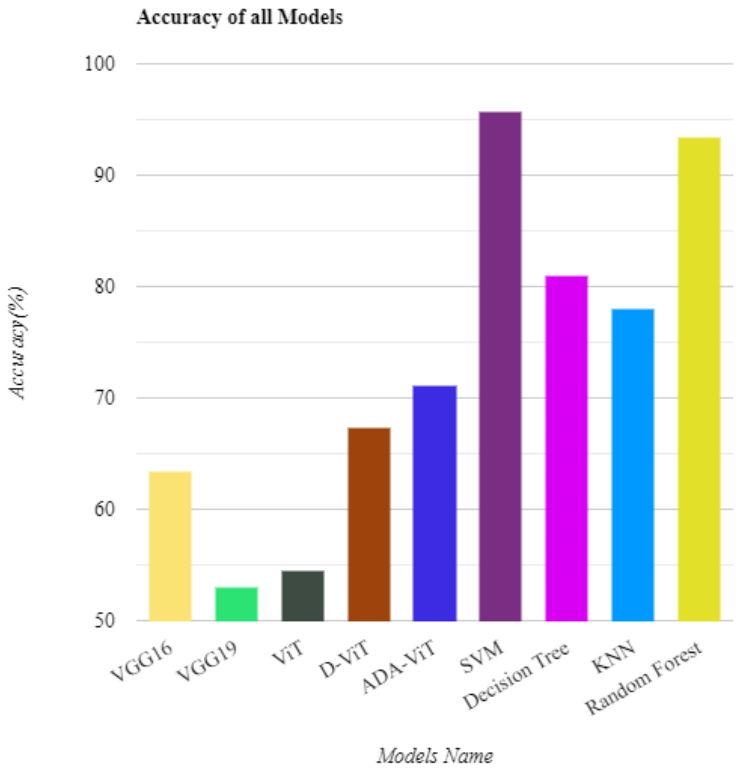

Figure 7.5: Accuracy of All Models (image size:224 x 244)

7.2.3 Accuracy Graphs & Confusion Matrix

Accuracy Graphs

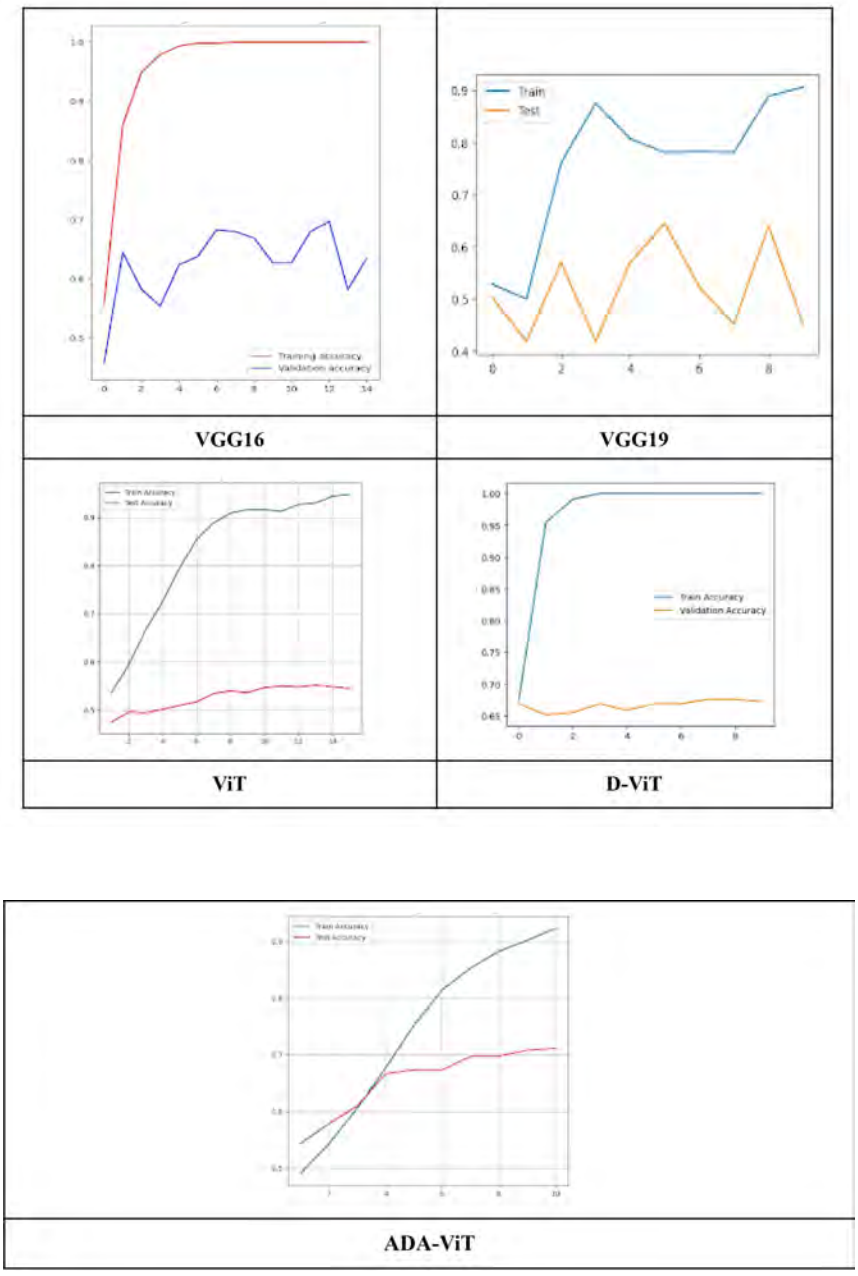

Figure 7.6: Train vs Validation Graphs of Deep Learning Models

After observing the train vs validation graphs of deep learning models, here ADA-ViT models performed well. Because in the ADA-ViT model, the train loss is lower so that the model performance is quite impressive.

Confusion Matrix

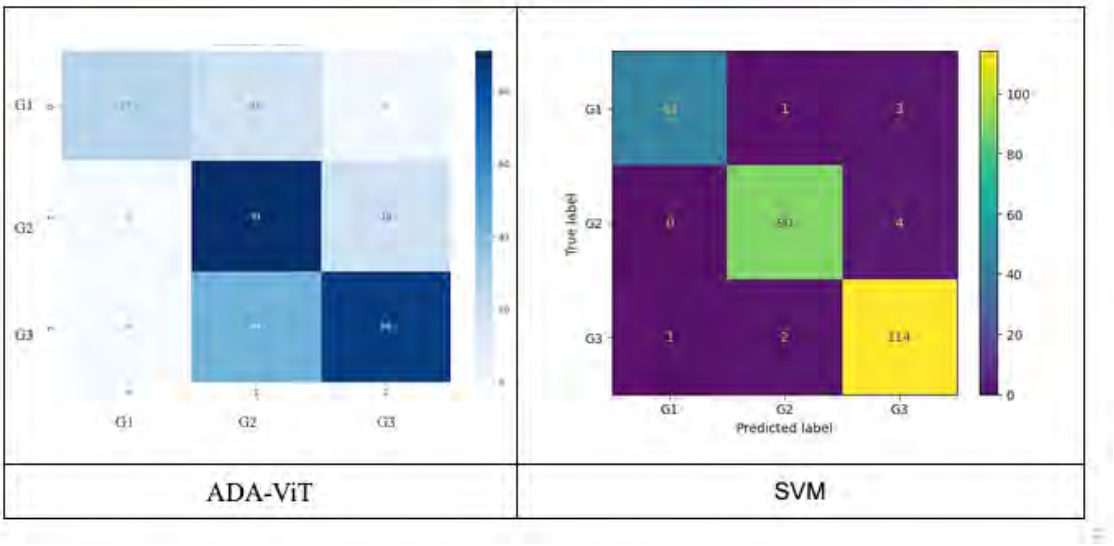

Figure 7.7: Best Performed Model's Confusion Matrix

From the above matrix, we can clearly see that SVM correctly predicted most of the test cases, whereas ADA-ViT predicted many test cases incorrectly.

7.3 Proposed Customization

From the analysis above, it is clear that the vectorization approach outperformed all the deep learning models. Therefore, we experimented with several techniques with a vectorization approach so that we could increase the performance of the best-performed model. As SVM gave the highest result, we have used this model in our customized architecture.

Previously, when we were training and testing models, we considered only the damaged images as the data. Now, we want to use both initial and damaged data on the customized model architecture of SVM. Moreover, our previous training method was converting images to 2D vectors, then flattening them to 1D, and then considering them as features and trains. However, now we can't use this training process as we want to consider two images at the same time. Here comes the necessity of new architecture.

In our customized architecture, firstly, we took both initial and damaged images of a structure. Secondly, we converted those images to gray scale. Thirdly, we converted both images to 2D vectors using the OpenCV library. Fourthly, we flattened and converted those images to 1D vectors. Next, we used some matrices (for example: Cosine similarity, Manhattan distance, Euclidean distance) to measure the similarity or difference between two vectors and made a feature using the values. Finally, we trained the model using the new feature.

In our process, we identified an issue. When we were measuring similarity the

expected range of output out of 100 was for grade 1 (70 to 80), for grade 2 (40 to 60), and for grade 3 (20 to 40), but we did not get the result as expected. The result was for grade 1 (70-80), for grade 2 (55-75), for grade 3 (40-60). We can see that the range of the three categories overlapped. We also found similar characteristics for the distance-calculating matrices. Therefore, if we train only using these values as features, it may produce biased results. As expected, we got an accuracy score below 70%. Sometimes, the model even can't distinguish between grade 2 and grade 3.

At this point, we thought that because of flattening the vector to 1D, it might lose some information and thus the distance matrices can't differentiate them properly. Therefore, we decided not to flatten them, and keep them as 2D. With the 2D vectors, we followed the above procedure and trained our model. However, these changes did not perform well either because of the same issue. Finally, we found a solution to this problem.

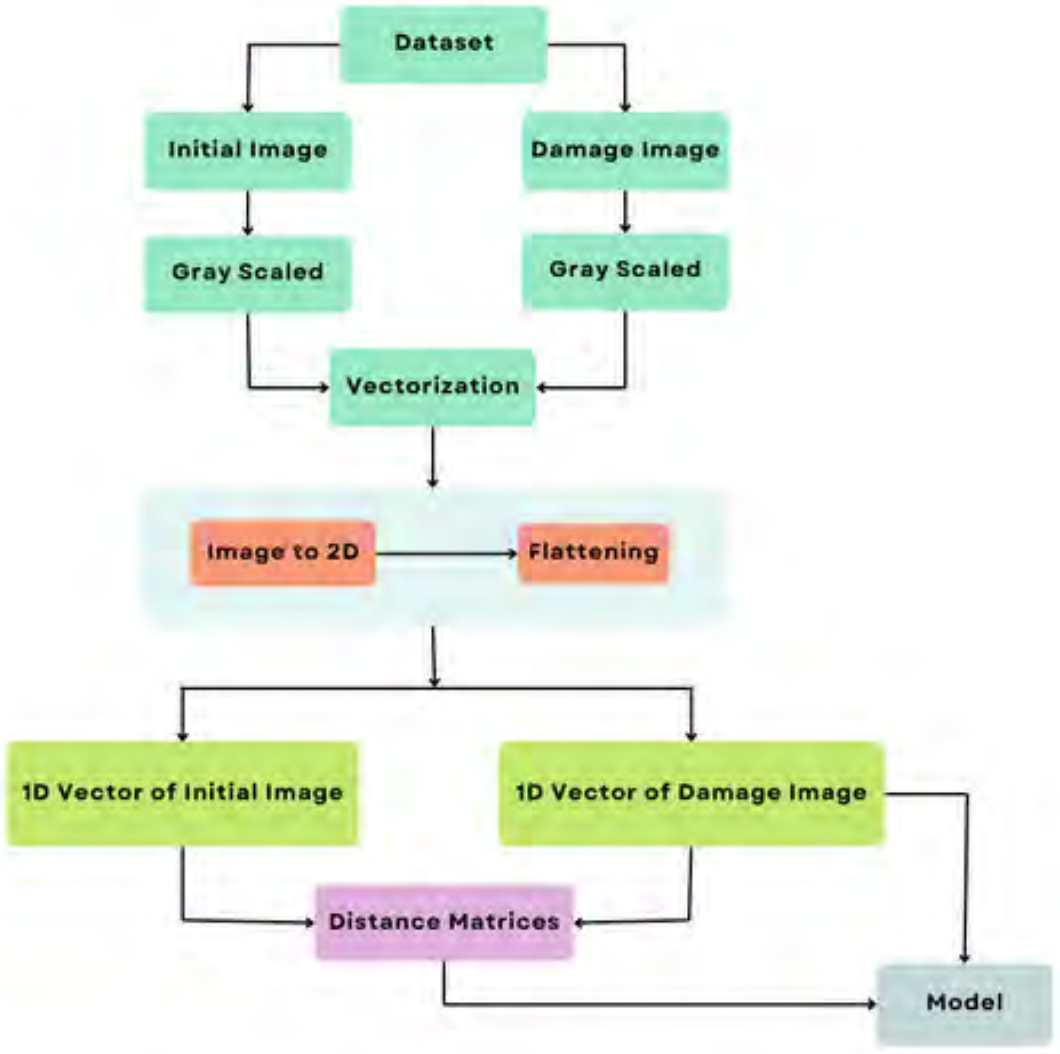

Figure 7.8: Detailed Overview of Customization

In our final customized architecture, first, we took the initial and damaged image separately from our dataset. Second, we converted them to gray scale image separately. Third, we vectorized both using the OpenCV library and flattened them to 1D vector. Fourth, we calculated the similarity and difference between them using distance matrices (for example: Cosine similarity, Manhattan distance, and Euclidean distance). Next, we trained our model using the distance matrix along with the 1D vector of the damaged image. This approach gave us the highest accuracy of (98.1%).

Model Name	Accuracy
SVM	95.67%
SVM (with customized system)	98.1%

Table 7.3: Score Comparison with Customized System

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Seismic and natural disastrous events have devastating effects when it hits an area with a higher magnitude. Therefore, first-world countries are preparing and trying to reduce after-event casualties. Third-world countries like Bangladesh are at high risk because of unplanned urbanization. 65% of structures in Dhaka are at high risk due to the construction of buildings on landfills and without maintaining the safety index [65].

The government should take appropriate steps to minimize the damage. In our thesis, we have tried to provide a quick assessment after the impact. Our goal was to classify the structure and identify the damage grade of the structures using the visual representation of the structure. Additionally, we have implemented various types of models and found the best-performed model in terms of accuracy. Moreover, we have proposed a customized architecture that outperformed all the previous models. As we are the first who used both initial and damaged representations while classifying and grading, it will become helpful for people to think differently who want to work in this field. In conclusion, if we can provide a quick assessment after the impact, it will be very helpful to reduce after-event damage and casualties.

8.2 Future Work

In future, Our idea is to develop a web interface that implements the best-performing model at the backend. This web app will have multiple features such as it will take a structure's initial and damaged images along with the location of the structure from a user. It will generate a damage map, which will help the rescue team to understand the situation and to set priorities based on the data.

Figure 8.1: Damage Map

We hope that this project will be helpful for future research and development.

Bibliography

[1] "Death toll climbs above 50,000 after turkey, syria earthquakes," *Al Jazeera*, 2023, Accessed: 16-10-2024. [Online]. Available: https://www.aljazeera.com/news/2023/2/25/death-toll-climbs-above-50000-after-turkey-syria-earthquakes.

[2] M. Sharmin, "Are we taking the risk of earthquakes seriously?" *The Daily Star*, 2023, Accessed: 16-10-2024. [Online]. Available: https://www.thedailystar.net/opinion/views/news/are-we-taking-the-risk-earthquakes-seriously-3485811.

[3] N. F. Antara, "What happens if a 7.5 quake hits dhaka?" *Dhaka Tribune*, 2023, Accessed: 16-10-2024. [Online]. Available: https://www.dhakatribune.com/bangladesh/dhaka/304530/what-happens-if-a-7.5-quake-hits-dhaka.

[4] X. Z. Shengyuan Li, "Image-based concrete crack detection using convolutional neural network and exhaustive search technique," 2019. DOI: 10.1155/2019/6520620. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1155/2019/6520620.

[5] Y. Zhang, H. V. Burton, H. Sun, and M. Shokrabadi, "A machine learning framework for assessing post-earthquake structural safety," *Structural Safety*, vol. 72, pp. 1–16, 2018. DOI: 10.1016/j.strusafe.2017.12.001. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0167473017300851.

[6] T. Rahman, Y. Ahmed, T. Alam, *et al.*, "Evaluation of earthquake resistance of urban buildings using image processing and machine learning techniques," 2020. DOI: 10.1109/CSDE50874.2020.9411582.

[7] K. Chaurasia, S. Kanse, A. Yewale, V. K. Singh, B. Sharma, and B. R. Dattu, "Predicting damage to buildings caused by earthquakes using machine learning techniques," in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, 2019, pp. 81–86. DOI: 10.1109/IACC48062.2019.8971453.

[8] A. Rao, J. Jung, V. Silva, G. Molinario, and S.-H. Yun, "Earthquake building damage detection based on synthetic-aperture-radar imagery and machine learning," *Natural Hazards and Earth System Sciences*, vol. 23, no. 2, pp. 789–807, 2023. DOI: 10.5194/nhess-23-789-2023. [Online]. Available: https://nhess.copernicus.org/articles/23/789/2023/.

[9] J. Bialas, T. Oommen, U. Rebbapragada, and E. Levin, "Object-based classification of earthquake damage from high-resolution optical imagery using machine learning," *Journal of Applied Remote Sensing*, vol. 10, no. 3, Sep. 21, 2016. DOI: 10.1117/1.jrs.10.036025.

[10] D. Duarte, F. Nex, N. Kerle, and G. Vosselman, "Multi-resolution feature fusion for image classification of building damages with convolutional neural networks," *Remote Sensing*, vol. 10, no. 10, 2018, ISSN: 2072-4292. DOI: 10.3390/rs10101636. [Online]. Available: https://www.mdpi.com/2072-4292/10/10/1636.

[11] E. Harirchian, V. Kumari, K. Jadhav, S. Rasulzade, T. Lahmer, and R. Raj Das, *A synthesized study based on machine learning approaches for rapid classifying earthquake damage grades to rc buildings*, 2021. DOI: 10.3390/app11167540. [Online]. Available: https://www.mdpi.com/2076-3417/11/16/7540.

[12] M. Kovačević, Z. Stojadinovic, D. Marinković, and B. Stojadinovic, "Sampling and machine learning methods for a rapid earthquake loss assessment system," Oct. 2019.

[13] S. Roeslin, Q. Ma, H. Juárez-Garcia, A. Gómez-Bernal, J. Wicker, and L. Wotherspoon, "A machine learning damage prediction model for the 2017 puebla-morelos, mexico, earthquake," *Earthquake Spectra*, vol. 36, no. 2, pp. 314–339, 2020. DOI: 10.1177/8755293020936714.

[14] D. T. Nandwani and V. Buradkar, "Earthquake damage prediction using machine learning," vol. 10, 2022, ISSN: 2320-2882. [Online]. Available: https://ijcrt.org/papers/IJCRT2207293.pdf.

[15] S. D. Nukala, V. K. Mishra, and G. K. M. Nookala, "Modeling earthquake damage grade level prediction using machine learning and deep learning techniques," in *Data Management, Analytics and Innovation*, N. Sharma, A. Chakrabarti, V. E. Balas, and J. Martinovic, Eds., Singapore: Springer Singapore, 2021, pp. 421–433, ISBN: 978-981-15-5619-7.

[16] A. Prasetyo, E. Yuniarto, P. Suprobo, and A. Tambusay, "Application of edge detection technique for concrete surface crack detection," in *2022 International Seminar on Intelligent Technology and Its Applications*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 209–213. DOI: 10.1109/ISITIA56226.2022.9855280.

[17] E. Işık, M. Shendkar, F. Avcil, A. Buyuksarac, and S. Deshpande, "A study on the determination of damage levels in reinforced concrete structures during the kahramanmaraş earthquake on february 06, 2023," *E3S Web of Conferences*, vol. 405, Jul. 2023. DOI: 10.1051/e3sconf/202340504029.

[18] E. IŞIK, A. E. ULU, Ş. TUNÇ, J. Shan, and A. KESKINER, "A study on the determination of damage levels in reinforced concrete structures for different earthquakes," *Journal of Science and Technology*, vol. 12, pp. 14–20, 2022, ISSN: 2146-7706. [Online]. Available: https://dergipark.org.tr/en/download/article-file/2223826.

[19] S. Okada and N. Takai, "Classifications of structural types and damage patterns of buildings for earthquake field investigation," *Journal of Structural and Construction Engineering, AIJ*, vol. 524, Oct. 1999. DOI: 10.3130/aijs.64.65_5.

[20] *Introduction to convolution neural network*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.geeksforgeeks.org/introduction-convolution-neural-network/.

[21] N. Malviya, *Convolution and pooling layers*, Accessed: 16-10-2024, 2023. [Online]. Available: https://medium.com/@nikitamalviya/convolution-pooling-f8e797898cf9.

[22] Dshahid, *Convolutional neural network*, Accessed: 16-10-2024, 2019. [Online]. Available: https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529.

[23] *Convolutional layer*, Accessed: 16-10-2024. [Online]. Available: https://www.databricks.com/glossary/convolutional-layer.

[24] *What is a convolutional neural network?* Accessed: 16-10-2024, 2023. [Online]. Available: https://skyengine.ai/se/skyengine-blog/125-what-is-a-convolutional-neural-network.

[25] *Introduction to convolution neural network*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.geeksforgeeks.org/introduction-convolution-neural-network/.

[26] Jorgecardete, *Convolutional neural networks: A comprehensive guide*, Accessed: 16-10-2024, 2024. [Online]. Available: https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175.

[27] *Fully connected layer vs convolutional layer*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.geeksforgeeks.org/fully-connected-layer-vs-convolutional-layer/.

[28] M. Ali, *Introduction to activation functions in neural networks*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.datacamp.com/tutorial/introduction-to-activation-functions-in-neural-networks?utm_source=google&utm_medium=paid_search&utm_campaignid=19589720824&utm_adgroupid=157156376071&utm_device=c&utm_keyword=&utm_matchtype=&utm_network=g&utm_adpostion=&u.

[29] J. Brownlee, *A gentle introduction to the rectified linear unit (relu)*, Accessed: 16-10-2024, 2020. [Online]. Available: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

[30] IBM, *What is a transformer model?* Accessed: 16-10-2024, 2024. [Online]. Available: https://www.ibm.com/topics/transformer-model.

[31] A. P. M, *Tokenization in transformers*, Accessed: 16-10-2024, 2024. [Online]. Available: https://medium.com/@abhijithprasadmkp/tokenization-in-transformers-9e8b0a5fd5f4.

[32] *Bert for image transformers (beit): A definitive guide to computer vision breakthrough*, Accessed: 16-10-2024, 2023. [Online]. Available: https://bolster.ai/blog/bert-image-transformer.

[33] S. Herath, *Input embedding sublayer in the transformer model*, Accessed: 16-10-2024, 2024. [Online]. Available: https://medium.com/image-processing-with-python/input-embedding-sublayer-in-the-transformer-model-7346f160567d.

[34] J. Ferrer, *How transformers work: A detailed exploration of transformer architecture*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.datacamp.com/tutorial/how-transformers-work.

[35] L. Serrano, *What are transformer models and how do they work?* Accessed: 16-10-2024, 2024. [Online]. Available: https://cohere.com/llmu/what-are-transformer-models.

[36] P. BL, *Difference between self-attention and multi-head self-attention*, Accessed: 16-10-2024, 2024. [Online]. Available: https://medium.com/@punya8147_26846/difference-between-self-attention-and-multi-head-self-attention-e33ebf4f3ee0.

[37] D. S. Ali Reza Sajun Imran Zualkernan, *A historical survey of advances in transformer architectures*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/10/4316.

[38] K. Gomez, *The feedforward demystified: A core operation of transformers*, Accessed: 16-10-2024, 2023. [Online]. Available: https://medium.com/@kyeg/the-feedforward-demystified-a-core-operation-of-transformers-afcd3a136c4c.

[39] M. Hoque, *A comprehensive overview of transformer-based models: Encoders, decoders, and more*, Accessed: 16-10-2024, 2023. [Online]. Available: https://medium.com/@minh.hoque/a-comprehensive-overview-of-transformer-based-models-encoders-decoders-and-more-e9bc0644a4e5.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, ISSN: 0001-0782. [Online]. Available: https://doi.org/10.1145/3065386.

[41] GeeksforGeeks, *Vgg-16 — cnn model*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.geeksforgeeks.org/vgg-16-cnn-model/.

[42] *Very deep convolutional networks (vgg) essential guide*, Accessed: 16-10-2024, 2021. [Online]. Available: https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/.

[43] O. Dahmane, M. Khelifi, M. Beladgham, and I. Kadri, "Pneumonia detection based on transfer learning and a combination of vgg19 and a cnn built from scratch," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 3, pp. 1469–1480, 2021, ISSN: 2502-4760. DOI: 10.11591/ijeecs.v24.i3.pp1469-1480. [Online]. Available: https://ijeecs.iaescore.com/index.php/IJEECS/article/view/24399.

[44] *Vgg-net architecture explained. geekforgeeks*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.geeksforgeeks.org/vgg-net-architecture-explained/.

[45] G. Boesch, *Vision transformers (vit) in image recognition – 2024 guide*, Accessed: 16-10-2024, 2023. [Online]. Available: https://viso.ai/deep-learning/vision-transformer-vit/.

[46] D. Shah, *Vision transformer: What it is & how it works [2024 guide]*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.v7labs.com/blog/vision-transformer-guide.

[47] Y. Wang, R. Huang, S. Song, Z. Huang, and G. Huang, "Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21, Red Hook, NY, USA: Curran Associates Inc., 2024, ISBN: 9781713845393.

[48] H. Yin, A. Vahdat, J. Alvarez, A. Mallya, J. Kautz, and P. Molchanov, *Adavit: Adaptive tokens for efficient vision transformer*, 2022. arXiv: 2112.07658 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2112.07658.

[49] S. Zhang, H. Liu, S. Lin, and K. He, *You only need less attention at each stage in vision transformers*, 2024. arXiv: 2406.00427 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2406.00427.

[50] *Support vector machine (svm) algorithm*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.geeksforgeeks.org/support-vector-machine-algorithm/.

[51] *Decision tree*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.geeksforgeeks.org/decision-tree/.

[52] *K-nearest neighbor(knn) algorithm*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.geeksforgeeks.org/k-nearest-neighbours/.

[53] *Random forest algorithm in machine learning*, Accessed: 16-10-2024, 2024. [Online]. Available: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/.

[54] M. A. Zanini, L. Hofer, and F. Faleschini, "Reversible ground motion-to-intensity conversion equations based on the ems-98 scale," *Engineering Structures*, vol. 180, pp. 310–320, 2019, ISSN: 0141-0296. DOI: https://doi.org/10.1016/j.engstruct.2018.11.032. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141029618307570.

[55] U. Day, *How to mount google drive in google colab*, Accessed: 16-10-2024, 2024. [Online]. Available: https://medium.com/@wl8380/how-to-mount-google-drive-in-google-colab-c688ec8eccb7.

[56] *Python and data science tutorial in visual studio code*, Accessed: 16-10-2024, 2024. [Online]. Available: https://code.visualstudio.com/docs/datascience/data-science-tutorial.

[57] S. Herath, *Input embedding sublayer in the transformer model*, Accessed: 16-10-2024, 2024. [Online]. Available: https://medium.com/image-processing-with-python/input-embedding-sublayer-in-the-transformer-model-7346f160567d.

[58] *Keras applications*, Accessed: 16-10-2024. [Online]. Available: https://keras.io/api/applications/.

[59] *Test set —— what is a test set in machine learning?* Accessed: 16-10-2024, 2024. [Online]. Available: https://www.hopsworks.ai/dictionary/test-set.

[60] H. Zhang, Y. Reuland, J. Shan, and E. Chatzi, "Post-earthquake structural damage assessment and damage state evaluation for rc structures with experimental validation," *Engineering Structures*, vol. 304, p. 117 591, 2024, ISSN: 0141-0296. DOI: https://doi.org/10.1016/j.engstruct.2024.117591. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141029624001536.

[61] I. Castillo Camacho and K. Wang, "Data-dependent scaling of cnn's first layer for improved image manipulation detection," in *IWDW 2020 - 19th International Workshop on Digital-forensics and Watermarking*, ser. Proceedings of the 19th International Workshop on Digital-forensics and Watermarking, Melbourne (online), Australia: Springer, 2020, pp. 208–223. DOI: 10.1007/978-3-030-69449-4_16. [Online]. Available: https://hal.science/hal-03000629.

[62] *Overfitting in machine learning: How to detect and avoid overfitting in computer vision?* Accessed: 16-10-2024, 2024. [Online]. Available: https://encord.com/blog/overfitting-in-machine-learning/.

[63] A. Jain, *Data augmentation*, Accessed: 16-10-2024, 2024. [Online]. Available: https://medium.com/@abhishekjainindore24/data-augmentation-00c72f5f4c54.

[64] Jyotsana, *Image augmentation techniques*, Accessed: 16-10-2024, 2023. [Online]. Available: https://medium.com/@jyotsana.cg/image-augmentation-techniques-798243f6afdf.

[65] A. M. Khan, *What if an earthquake of 6.9 magnitude hits dhaka?* Accessed: 16-10-2024, 2023. [Online]. Available: https://www.thedailystar.net/opinion/views/news/what-if-earthquake-69-magnitude-hits-dhaka-3421611.