BRAC
UNIVERSITY

Inspiring Excellence

Final Thesis Report

Image Processing and Deep Learning Approach to Evaluate Earthquake Resistance of Urban Buildings

Group Members

Yasin Ahmed

13301118

Tawsiful Alam

13301114

Md. Hasibur Rahman Shakil

13301016

Md. Tanjidul Hossain

13101153

Supervisor

Dr. Md. Ashraful Alam

Assistant Professor

Dept. of Computer Science and Engineering

BRAC University

Declaration

We hereby declare that this is an original report written by us with our own findings and has not been published or presented in parts or as a whole for any other previous degree. Resources and materials by other researchers used as guidelines for our research are fully mentioned in reference citations.

Signature of Authors :	Signature of the Supervisor :

____

Yasin Ahmed

____

Md. Hasibur Rahman Shakil

____

Tawsiful Alam

____

Md. Tanjidul Hossain

____

Dr. Md. Ashraful Alam

Assistant Professor

Dept. of Computer Science and

Engineering, BRAC University

Acknowledgement

We would like to express our heartiest gratitude towards the Almighty Allah. Secondly, we would like to share our sincere gratitude to our advisor Dr. Md. Ashraful Alam for his constant motivation, support and immense knowledge to our research. It would not have been possible without his constant guidance in all phases of our project. Also, special thanks to Dr. Mehedi Hasan Ansari, professor of the faculty of Geotechnical Engineering, Department of Civil Engineering, BUET holding the position of Secretary General, Bangladesh Earthquake Society (BES) & Linkage Director, for authenticating civil engineering granted model for evaluating building. We would also like to thank our friends for helping us collecting image data with their cameras and capturing photos of this busy city's buildings. Special thanks to Mr. Monirul Islam Pavel and Safat Islam for perfectly guiding us in Image Processing and Machine Learning. Finally, we would like to thank our gratefulness to parents, brothers, sisters and beloved friends for encouraging and supporting us from the very beginning.

Index

List of Figures

List of Tables

List of Equations

List of Abbreviations

- **API** : Application Program Interface
- **BRI** : Building Research Institute
- **BUET** : Bangladesh University of Engineering and Technology
- **CNN** : Convolutional Neural Network
- **CSS** : Cascading Style Sheets
- **CUET** : Chittagong University of Engineering and Technology
- **FEMA** : Foreign Exchange Management Act
- **HTML** : Hypertext Markup Language
- **MATLAB** : Matrix Laboratory
- **PHP** : Personal Home Page
- **PWRI** : Public Work Research Institute
- **RAJUK** : Rajdhani Unnayan Kartripakkha
- **RC** : Reinforce Concrete
- **SQL** : Structured Query Language
- **SVM** : Support Vector Machine
- **UI** : User Interface
- **URM INF** : Unreinforced Masonry Infill
- **USGS** : United States Geological Survey
- **VGG** : Visual Geometry Group

Abstract

The capital of Bangladesh, Dhaka, is one of the most densely populated cities in the world, sits atop the world's largest river delta at close to sea level, which can trigger a massive earthquake resulting in death of millions of people. To minimizing such casualties, marking risky buildings can be an efficient approach as these buildings have more chance to collapse. In this paper, a new approach has been introduced on spotting these buildings by taking visual view through images and calculating the risk factors using Image Processing and Deep Learning. By following FEMA154 method of calculating risk factors of C3 type URM INF buildings, Image processing has been applied on the image data to get results and SVM has been run on the manual data of the risk factors. We used neural network model VGG16 and altered it to a newer version for beam detecting via images. The buildings have been shown in the map marking red or green to let people know about the vulnerability of these buildings.

Keywords: Earthquake, Deep Learning, Image Processing, CNN, SVM, Machine Learning.

Chapter 1
Introduction

Earthquake is one of the most occurring natural calamities that causes outrageous disaster in seconds. It is usually possible for human beings to predict an upcoming natural calamity but it is almost impossible to predict a forthcoming earthquake. As we cannot impede an earthquake from occurring, all we can do is prepare ourselves to survive against it. The best way is to assure that our buildings should not get wrecked at an event of an earthquake. Proper planning and maintaining the codes while constructing a building is the primary postulate here. In contrast to the existing situation in our country we see that these rules are hardly being followed. So gradually, the risk rate of the mega city Dhaka crumbling in ruins is too high to imagine even against an earthquake of moderately high seismicity. In this case, determining the risky buildings across the city is an important approach to minimize the casualties, however it is not a simple task. Generally, it takes weeks for a group of civil engineers to calculate the risk factors of a building. So, it is very difficult to calculate the risk factors of all the buildings in Dhaka and thus we need to find out another faster and effective way to do the job with the help of technology. Technological advancement is a continuous process and we should always try to apply these technologies to different fields. In order to make things better or more efficient than before these new technologies can be applicable to these different fields. Here, we tried to do this difficult job with the help of one of the subsets of Artificial Intelligent - Deep Learning, and also Image Processing. We followed the procedures of FEMA154 Tier 1 for Walk Down Survey process to detect the risk factors of a building over visual inspection.

1.1 Motivation

We have been hearing this for a long period of time that a massive earthquake can occur at any time in our country, causing huge destruction and taking millions of human lives. Geographically Bangladesh is located close to the two subduction zones created by two active tectonic plates: the Indian plate and the Eurasian plate. Five geological fault lines run through the country, exposing it to highly vulnerable of a major quake by the experts. If a massive earthquake with 7 or greater magnitude occurred in this country will led a major human tragedy due to the faulty structures of many buildings and proper awareness [29]. Moreover, in Madhupur, there is another fault line, which might not be that strong but can make substantial destruction as it is nearest to the capital [29]. Furthermore, the recent earthquakes near our country that happened in Nepal and Assam indicates how vulnerable Dhaka city can be. Even though the epicenter was not in and around Dhaka city, aftershocks of that earthquake were also being felt. So, we cannot be assured whether or not a massive earthquake will occur but we can definitely be sure of that when it happens our existence will be in danger. To minimize the casualties, everyone should come forward and work on a solution. It might be through research or by following the rules or even simply by making the general people aware of the scenario. This is the primary motivation behind our work. Again, since we are responsible citizens of our country and we are getting the privilege to study technological subjects, we realized that we should try to do something for the betterment of our country through our field and agreed that working towards a solution for such imminent earthquakes in order to decrease the casualties and save the people will surely be a noble deed.

1.2 Objectives

The main objective of this thesis is to introduce a technological approach to determine the risky buildings of Dhaka city, which are at a threat in an event of an earthquake. Basically, what the civil engineers do to determine whether a building is risky or not at the very first phase. Our goal is to do this job with the help of machine learning and image processing by capturing photos of a number of buildings from different areas of the city and applying these methods over these photos and collected data. We aim to generate a result about the risk factors of these buildings and based on their scores, marking these on a map so that people get to know if a particular building is at risk or not. Thus, we hope to grow vast awareness among the people to follow the building codes while constructing a building as well as the authorities to take necessary steps or decisions about risky buildings of the city, which will be responsible for the loss of life during an earthquake. Above all, we believe that our research will encourage others to contribute in this sector with their profound knowledge and that will obviously result in a secured Dhaka in future.

1.3 Thesis Outline

- Chapter 2 Discusses the Literature Review of related works in this field
- Chapter 3: Discussing details of Algorithms that are used with structure and equations
- Chapter 4: Proposed model or methodology of our research.
- Chapter 5: Experiments and results of different algorithms and codes
- Chapter 6: Conclusion, challenges and future work proposal.

Chapter 2

Literature review

2.1: Brief History of Earthquake

Earthquake is a worldwide warning phenomenon that is different from any other natural disasters due to its unpredictability and possible rate of disasters. Japan is the most frequently damaged country due to earthquake [1]. There are many major earthquakes around the world in recent times including Haiti, Mexico, Nepal. Bangladesh has also faced many earthquakes and its consequences over the last hundred years. Among those, many major earthquakes shook the nation in an alarming way. From 1548's first recorded major earthquake that highly affected Sylhet and Chittagong there are some historically recorded earthquakes near this South Asian country. 1897's "great Indian earthquake" of magnitude 8.7 is the most dangerous one for Bangladesh till date [3]. The history of the most possibly dangerous natural disaster, earthquake, is versatile around Bangladesh. According to USGS, from 1900 there are 67 earthquakes near or on Bangladesh, which are above 6.0 in magnitude [2]. In April 2015, the earthquake of Nepal created a severe impact on Bangladeshi people's consciousness of earthquake, as it was the most severely felt quake of 21st century in our country. Especially the capital, Dhaka's people are concerned more for the density of population and RC type buildings. In the year, 2015 and 2016 followed by the Nepal's disaster there occurred many significant shakes that frightened the nation more. From the last 20 years, the capital Dhaka city has managed to grow many buildings mostly with no proper plans. These buildings are risky for any major earthquake and these can lead to catastrophically destructive event. Since Dhaka is home to almost 50 million people, the amount of damage can be undoubtedly huge. With this history and current affairs our work is specially focused on the earthquake aspects of Dhaka city, the capital of Bangladesh.

2.2: Technology and Science Against Earthquake

Computer science, physics, and many sensing technological fields are working to minimize the damage of earthquake in many ways. Earthquake detecting with seismic wave is not a new approach. In 1868, a doctor named J.D. Cooper laid out his vision of such a system [1]. After over a century later, in 1985 Caltech geophysicist Thomas Heaton sketched out a modern-day version, which he dubbed SCAN for Seismic Computerized Alert Network. In 1960's, Japanese Railway invented a seismic wave to alert and break the train if there is a major earthquake about to generate nearby. In 1970's that train break became automated for any earthquake that is about to come with more than 4 rector-scale [1]. In last 50-60 years, there went many studies and researches along with the usage of technology for stopping earthquake or reducing the damage factors. Predicting an upcoming earthquake, its rate and damage is the most needed yet not predictable. A news report of 2017 says that earthquake prediction can be soon given by the technology of satellite, as satellites are able to observe dips and bulges over earth surface [4]. Despite many computer science and sensing technologies earthquake damage control is the topic that needs fine tuning over time. On the other hand, Building Research Institute (BRI), Public Work Research Institute (PWRI), International Institute of Seismology and Earthquake Engineering worked for building risk management in the matter of earthquake [5]. There are many building performance studies and methods to verify a building's strength. Considering Bangladesh, particularly Dhaka, is different in many aspects like soil type, building type and structural attributes from the area of these studies and methods for building performance. For detecting damage rate for Bangladeshi building, we need to consider the aspects of Bangladeshi buildings and soils.

2.3 Modified Model from FEMA

FEMA is a method for building's strength calculation considering different attributes like high/mid rage, soil type, plan irregularity, vertical irregularity of buildings etc. According to the FEMA's model of 2002 there are many types of buildings among those experts considers Dhaka city's buildings are highly moderate buildings. This type of buildings has the basic score of 3.2 and considers risk free if the final score considering all the aspects results more than equal 2. Our thesis is followed by this model of FEMA considering Dhaka city's soil type which are basically D (S2) and E (S3 and S4) type soils [5]. As buildings often don't follow government rules in Dhaka and hardly maintain proper gap from one building to another. This gap, also known as pounding is a very important issue for earthquake's affect. Thus, to consider risk of buildings of Dhaka we did consider pounding as a factor along with other factors of FEMA 154 tier1 model [8]. We find "pounding" to be considered as a building risk factor by a study of CUET (Chittagong University of Engineering and Technology) [7].

Chapter 3

Algorithms

3.1 Image Processing

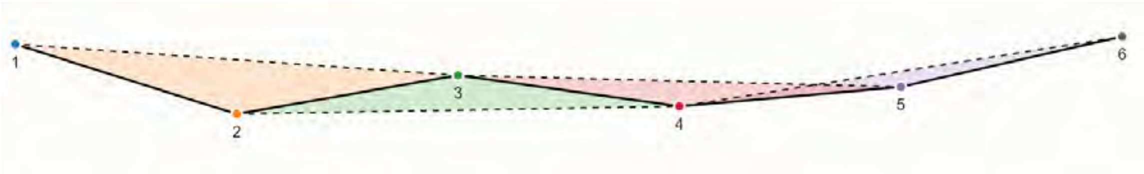
3.1.1 Ramer-Douglas-Peucker Algorithm

The Ramer-Douglas–Peucker algorithm is a famous algorithm for reducing the number of points in a curve that is approximated by a series of points or polylines. To do this, it “thinks” of a line between the first and last point in a set of points that form the curve. It checks which point in between is farthest away from this line. If the point (and as follows, all other in-between points) is closer than a given distance 'epsilon', it removes all these in-between points. If on the other hand this 'outlier point' is farther away from our imaginary line than epsilon, the curve is split in two parts:

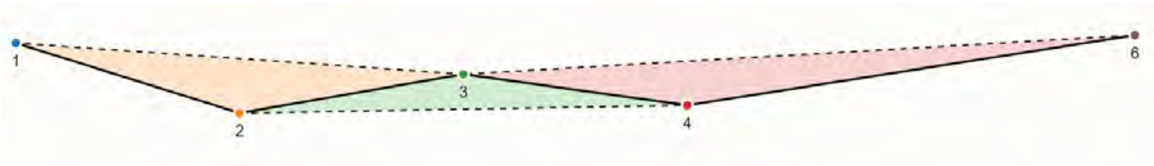
1. From the first point to the end including the outlier
2. The outlier and the remaining points.

The function is recursively called on both resulting curves, and the two reduced forms of the curve are put back together. In figure [3.1.1], we are showing an example of 5 polylines becoming 1 by using this algorithm.

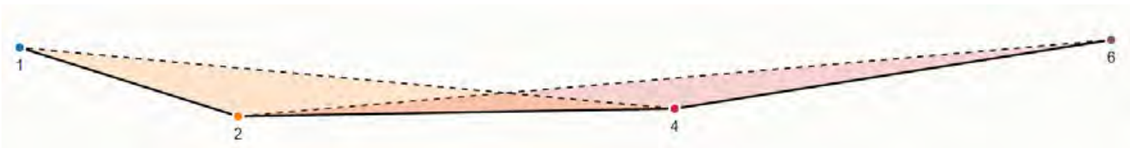
5 (Five) polylines consisting 6 series of points (1,2,3,4,5,6)



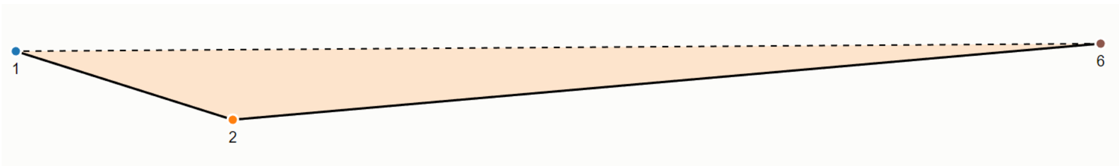
Selecting by skipping one points



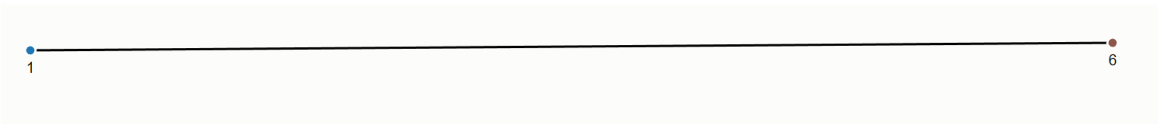
Eliminating the 5th point and drawing straight line from 4 to 6.



Eliminating the 3rd point



Eliminating the 4th point



Eliminating the 2nd point. Now it is a straight line from 1 to 6.

Figure 3.1.1: Ramer-Douglas–Peucker algorithm [30].

3.1.2 Moment of Image

Image moments are used to identify characteristics about objects including the centroid point and the bounding box surrounding it. This was used in the implementation as it provides simple and easy equations, which can be calculated to identify the information required. There are a variety of image moments, which can be calculated such as raw moments, central moments, scale invariant moments and rotation invariant moments.

The math of moments: In pure math, the nth order moment about the point c is defined as:

$$\mu_n = \int_{-\infty}^{+\infty} (x-c)^n f(x)dx \quad \text{------------------------------------} \quad (3.1)$$

Calculating area: To calculate the area of a binary image, you need to calculate its zeroth moment:

$$\mu_{0,0} = \sum_x^w \sum_y^h x^0\, y^0 f(x,y) \quad \text{------------------------------------} \quad (3.2)$$

The x_0 and y_0 do not have any effect and can be removed.

Centroid: To calculate the centroid of a binary image you need to calculate two coordinates -

$$centroid = \left(\frac{\mu_{1,0}}{\mu_{0,0}}, \frac{\mu_{0,1}}{\mu_{0,0}}\right) \quad \text{--} \quad (3.3)$$

3.1.3 Adaptive Thresholding Based Binary Masking, K-Means Clustering For Color Segmentation

Image thresholding is a simple, yet effective, way of partitioning an image into a foreground and background. This image analysis technique is a type of image segmentation that isolates objects by converting grayscale images into binary images

Threshold is usually the first step in any image processing.

Single value thresholding can be given mathematically as follows

$$G(x,y) = \begin{cases} 1, & f(x,y) > T \\ 0, & f(x,y) <= T \end{cases} \quad \text{--------------------------------} \quad (3.4)$$

K-Means Clustering

Clustering is a classification technique. Given a vector of N measurements describing each pixel or group of pixels in an image, a similarity of the measurement vectors and therefore their clustering in the N-dimensional measurement space implies similarity of the corresponding pixels or pixel groups. Therefore, clustering in measurement space may be an indicator of similarity of image regions, and may be used for segmentation purposes [12].

Most popular clustering algorithms suffer from two major drawbacks: First, the number of clusters is predefined, which makes them inadequate for batch processing of huge image databases. Secondly, the clusters are represented by their centroid and built using an Euclidean distance therefore inducing generally an hyper spherical cluster shape, which makes them unable to capture the real structure of the data. This is especially true in the case of color clustering where clusters are arbitrarily shaped.

Steps of K-Means Clustering Algorithm [13]

1. Initialization – generate the starting condition by defining the number of clusters and randomly select the initial cluster centers.
2. Generate a new partition by assigning each data point to the nearest cluster center.
3. Recalculate the centers for clusters receiving new data points and for clusters losing data points.
4. Repeat the steps 2 and 3 until a distance convergence criterion is met.
5. The K-means clustering is a partitioning method for grouping objects so that the within-group variance is minimized. By minimizing dissimilarity of each subset locally, the algorithm will globally yield an optimal dissimilarity of all subsets.

The algorithm, as applied to image threshold, is given by the following steps:

1. Initialize the (K) class centers. For simplicity, an equal-distance method is used to define the initial class centers:

$$\text{center}(i,0) = GL_{min} + \frac{\left[i - \frac{i}{2}\right](GL_{max} - GL_{min})}{k} \quad \text{--------------------} \quad (3.5)$$

i =1, 2, …., k

Where center(i,0) is the initial class center for the i^{th} class, GL_{max} and GL_{min} are the maximum and minimum of the gray value GL in the sample space, respectively.

2. Assign each point to its closest class center. The criterion to assign a point to a class is based on the Euclidean distance in the feature (GL) space using:

$$distance(i, j) = abs(GL_j - center_i) \quad (3.6)$$

i = 1, 2, …., K;

j = 1, 2,…., N;

Where Distance i, j is the distance from the j^{th} point to the i^{th} class, and N is the total number of points in the sample space.

3. Calculate the (K) new class centers from the mean of the points that are assigned to it. The new class centers are calculated by N_i

$$\text{center}(i, m) = \frac{1}{N_i \sum GL_j} \quad (3.7)$$

i = 1;

j = 1, 2,…., K;

Where N_i is the total number of points that are assigned to the i^{th} class in step 2.

4. Repeat step 2 if any class centers change, otherwise end the circulation.
5. The threshold value is defined as the average of the K^{th} class center and the $(K-1)^{th}$ class center:

$$\text{Threshold} = \frac{1}{2\,(\text{center}_k + center_{k-1})} \quad (3.8)$$

3.2 Deep Learning

3.2.1: CNN – VGG16 Architectural Model

Convolutional Neural Network in short CNN is one of the smoothest algorithms for deep learning. Convolutional Neural Network is precisely useful for image classification. While Rosenblatt researched artificial neural networks as early in 1960s, it was only in late 2000s when deep learning using neural networks took off [9]. This image classifier shocked people with the superb ability of predicting images correctly. Summary of CNN algorithm consists of four basic steps; convolution, subsampling, activation, fully connected. Every CNN must consist of these four basic layers.

Step 1: Convolution

The first layers that receive an input signal are called convolution filters. Convolution is a process where the network tries to label the input signal by referring to what it has learned in the past. Convolution has the nice property of being translational invariant. Intuitively, this means that each convolution filter represents a feature of interest, and the CNN algorithm learns which features comprise the resulting reference.

Step 2: Subsampling

Inputs from the convolution layer can be "smoothened" to reduce the sensitivity of the filters to noise and variations. This smoothing process is called subsampling and can be achieved by taking averages or taking the maximum over a sample of the signal, which is known as max pooling. If the average is pulled then it is average pooling.

Step 3: Activation

The activation layer controls how the signal flows from one layer to the next, emulating how neurons are fired in our brain. Output signals, which are strongly associated with past references, would activate more neurons, enabling signals to be propagated more efficiently for identification.

Step 4: Fully Connected

The last layers in the network are fully connected, meaning that neurons of preceding layers are connected to every neuron in subsequent layers. This mimics high-level reasoning where all possible pathways from the input to output are considered.

3.2.2 VGG16 ARCHITECTURAL MODEL

CNN algorithm has been used and updated into several models past few years and the results are extraordinary. From 2012's Alexnet's huge success rate with very low loss rate there are many architectural models proposed by different designers and coders. VGG16, one of the most promising model invented in 2014, has the maximum loss rate of 7%, which is surprisingly better than its previous famous model Alexnet. VGG16 architect of CNN algorithm uses 16 layers where Alexnet uses 8. In total 5 convolution filters is passed to iterate through each image. Later one fully connected layer is passed with flatten and dense operations [10]. It takes 224 height and weight as input along with 3 channels for RGB images. First convolution layer consists of 64 filters, second one 128 filters then 512 for the last three layers. It uses 2D max pooling and 'relu' activation function. In the fully connected layer, the model has 1000 dense as output classes with 'softmax' activation.

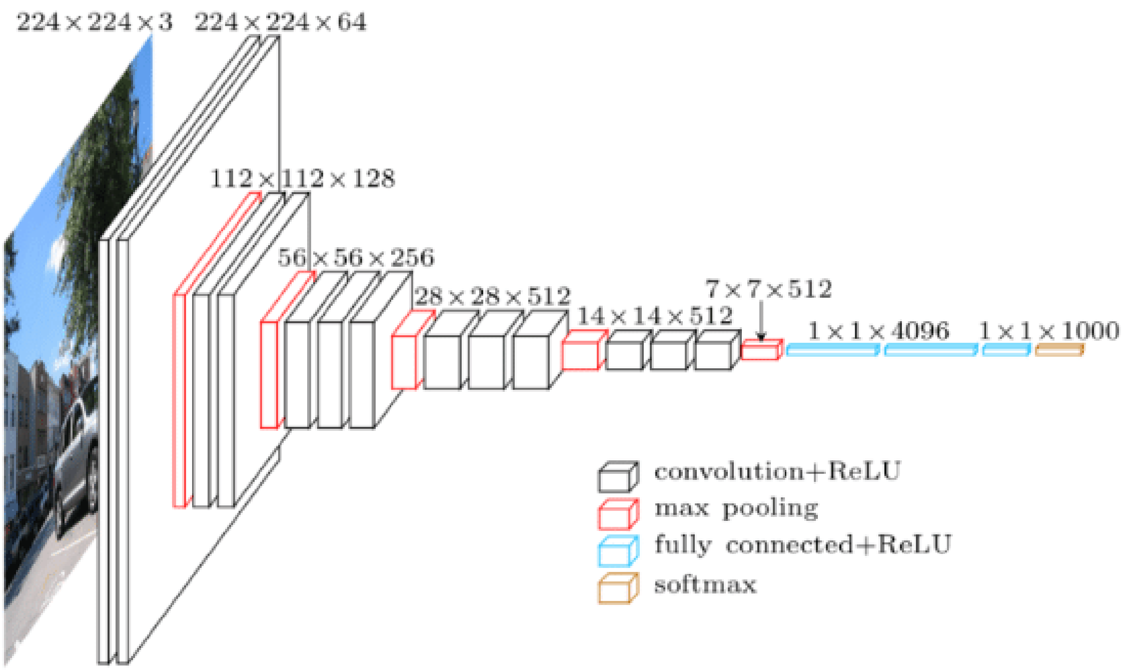


Figure 3.2.1 : VGG16 Architectural Model [28].

3.3 Support Vector Machine (SVM)

In Machine learning support vector machine (SVMs) are supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

SVM concurrently decrease the empirical classification error and increase the geometric classifier, which is a special feature of SVM [12]. So, it is called maximum margin classifier and it is called Structural risk Minimization [12]. The training dataset of n point is,

$$\{(x_1,y_1)(x_2,y_2)(x_3,y_3)\ldots\ldots\ldots(x_n, y_n)\} \quad \text{------------------------------} \quad (3.9)$$

Here, y_n is either 1 or –1 and y_n is indicating a class in which the point x_n belongs to. n=number of sample and x_n is p-dimensional real vector.

Since, we need to find out Maximum Margin Hyperplane which divides the point x_n as group for which y_n=1 to y_n=-1 and this is defined that the distance between the hyperplane and the nearest point is x_n from either group is maximized [12]. Any hyperplane can be written as

$$w.x - b = 0 \quad \text{---} \quad (3.10)$$

Where w is the p-dimensional vector and b is scalar. Hence, we are interested to maximized the margin parallel hyperplane is the best fit for the solution as we see below figure. Parallel hyperplane is expressed

$$w.x - b = 1 \quad \text{---} \quad (3.11)$$

$$w.x - b = -1 \quad \text{---} \quad (3.12)$$

We can use this hyperplane and try to maximize their distance, as our training data is linearly separable.

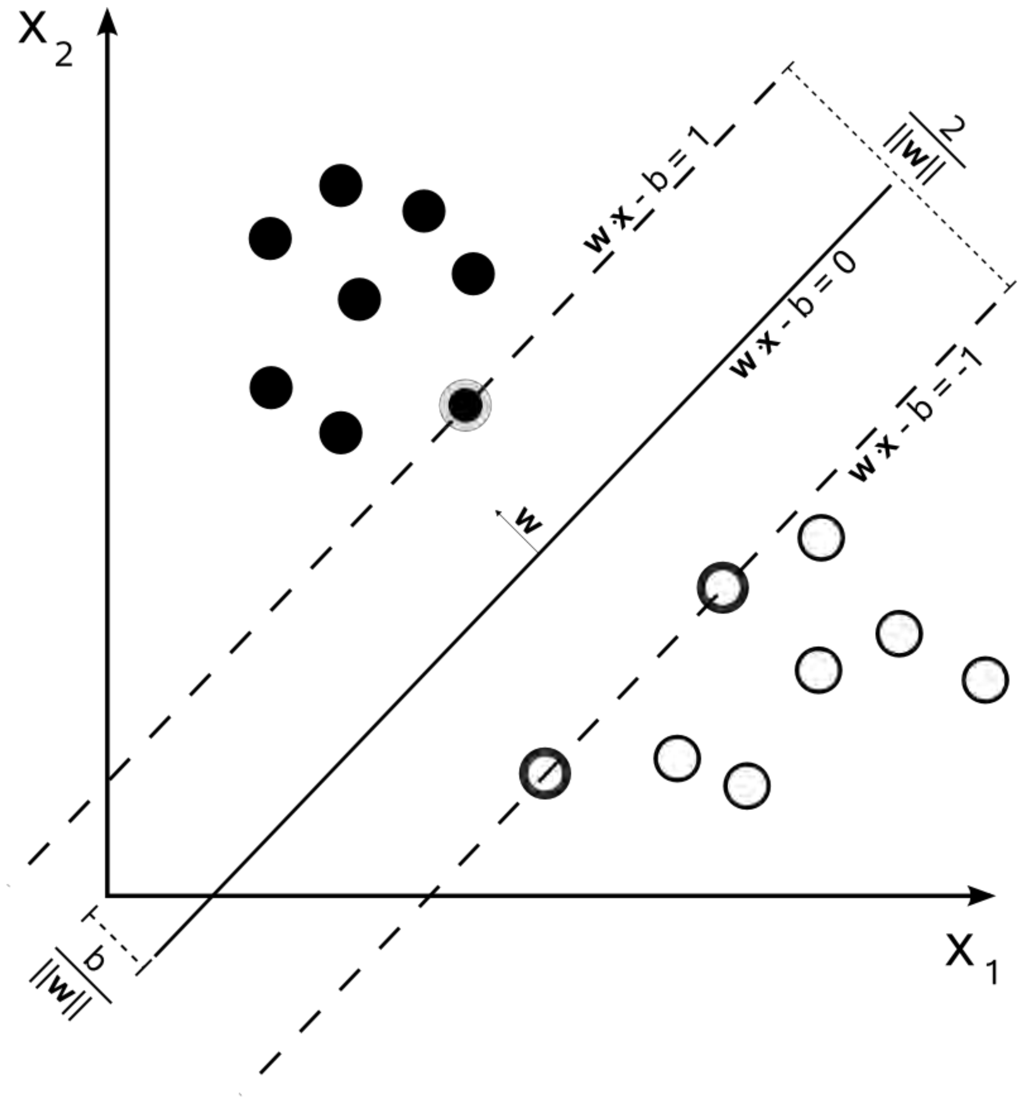


Figure 3.3 : Maximum-margin hyperplane and margins for an SVM

Trained with samples from two classes.

Chapter 4

Proposed model

Block Diagram of our thesis is shown below. First, we collect images of buildings through fieldwork. We extract data from collected images using FEMA 154 data collection form, which is used as manual extracted data, which will be compared to machine driven data. After that, we use CNN (Convolutional Neural Network) and Image Processing for extracting all the linear attribute through machine. After finishing that data sheets, we use SVM (Support Vector Machines). A subset of that data has been fed to SVM for training and a large set of the data are used to test the SVM classifier. Manually extracted data and data classified by SVM has been compared to determine the accuracy of the SVM.

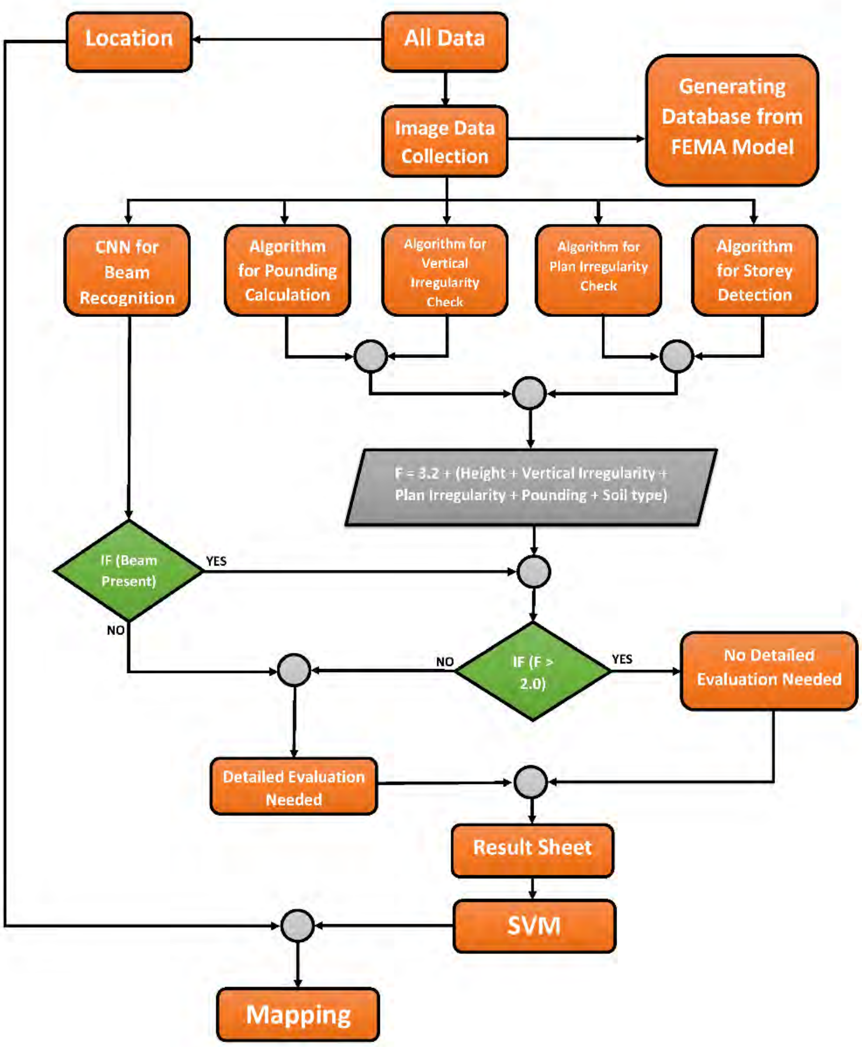


Figure 4.1.0: System Work Flow.

4.1 Image data collection

As our thesis is about risk attribute calculations of buildings of the capital Dhaka, we needed to collect authentic risk studies from RAJUK or any Civil Engineering paper. Unfortunately, government did not have sufficient data for implementing any computer science related automation. There are many building codes a particular building needs to follow for certain amount of strength to prevent incidents like major Earthquakes. Though we did not find any precise already available data, we followed civil engineering model to evaluate a building. The model is called FEMA 154 model for urban buildings. We collaborated with Dr. Mehedi Hasan Ansari Secretary General, Bangladesh Earthquake Society (BES) & Linkage Director to authenticate our building evaluation model for earthquake damageable building predictions. He helped us to modify FEMA 154 model for buildings of Dhaka. That particular model consists of many attributes to examine. We planned to collect those attribute's value of each building by inspecting and collecting particular photos for each building. We manually took around 1500 image data of 283 different building of 11 different areas of two different soil types of Dhaka.

4.2 Numeric Data Extraction from Images

We have six attributes to evaluate a building risk. These attributes are building height, vertical irregularity, plan irregularity, beam, pounding and soil type. For our data building height are two type, midrise and high rise and also soil has two type, D type soil and E type soil. Each attribute holds a specific value. For all building basic score is 3.2. We have to calculate the values that we have found from respected images. To calculate the final score, the expression is,

F =Basic score + (Building height + vertical irregularity +plan irregularity +soil type)

Here F is expressed as final score for a building. If final score is less than 2.0 then detailed Evaluation must be needed for that building.

Formula for determining the necessity of Detailed Evaluation is,

```
IF (Beam Status = False)
{
        Detailed Evaluation = True
}
ELSE
{
        IF (Final Score >= 2)
        {
                Detailed Evaluation = False
        }
        ELSE
        {
                Detailed Evaluation = True
        }
}
```

We calculated if detailed evaluation is needed for our 283 buildings, which prompts if a particular building is risky, or not for a certain level of earthquake to be felt nearby. Our building gives up some result that portrait the pattern of our capital, Dhaka, in case of area and building's risk factors. Here, risky buildings are plotted with red dot and non-risky buildings are plotted with green dot.

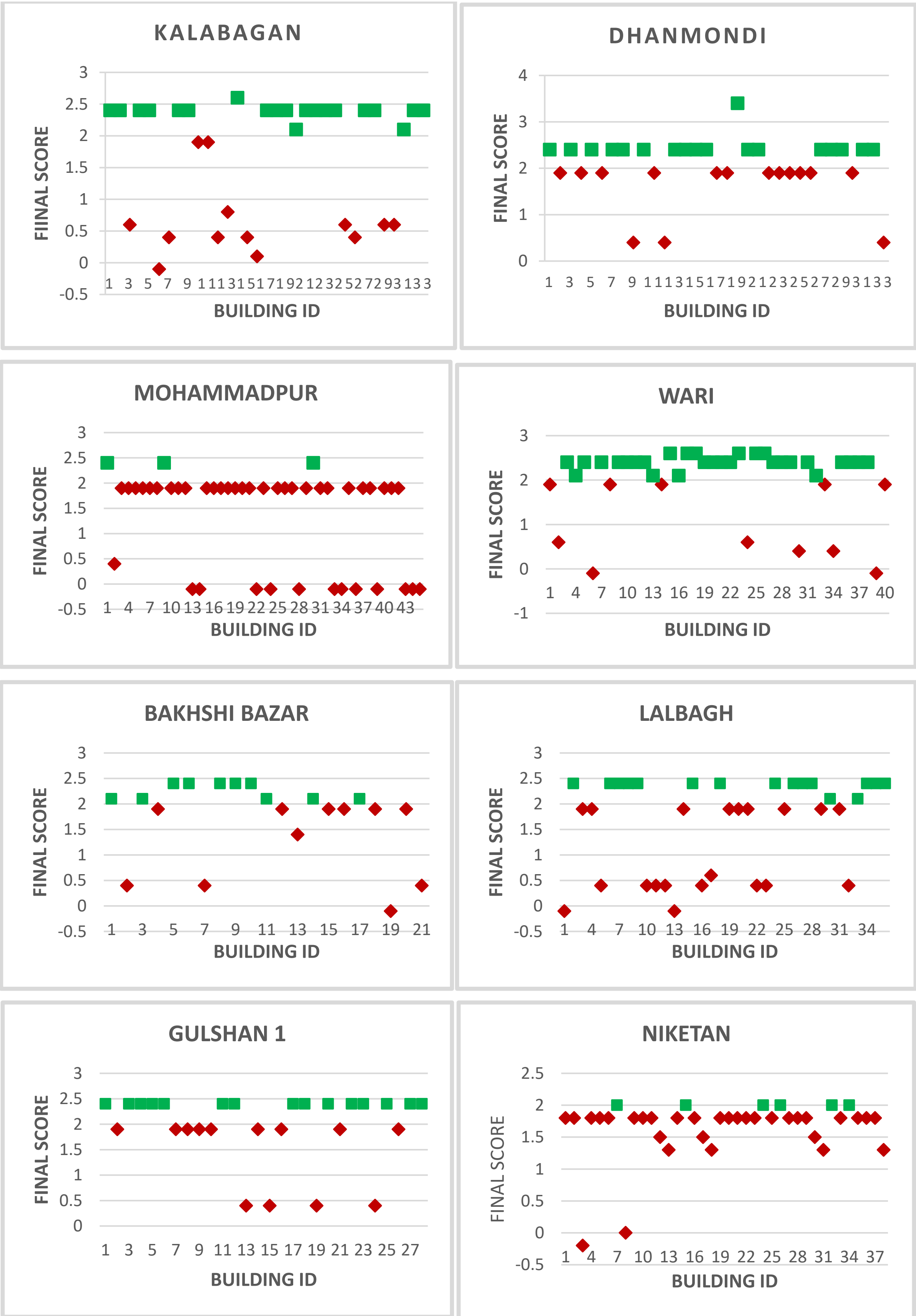


Figure 4.2 : Building's performance score distribution according to different zone of Dhaka city.

We find that different area has maintained pounding rule in particularly different pattern. Here in Mohammadpur and Mohakhali the rate for missing pounding is pretty high.

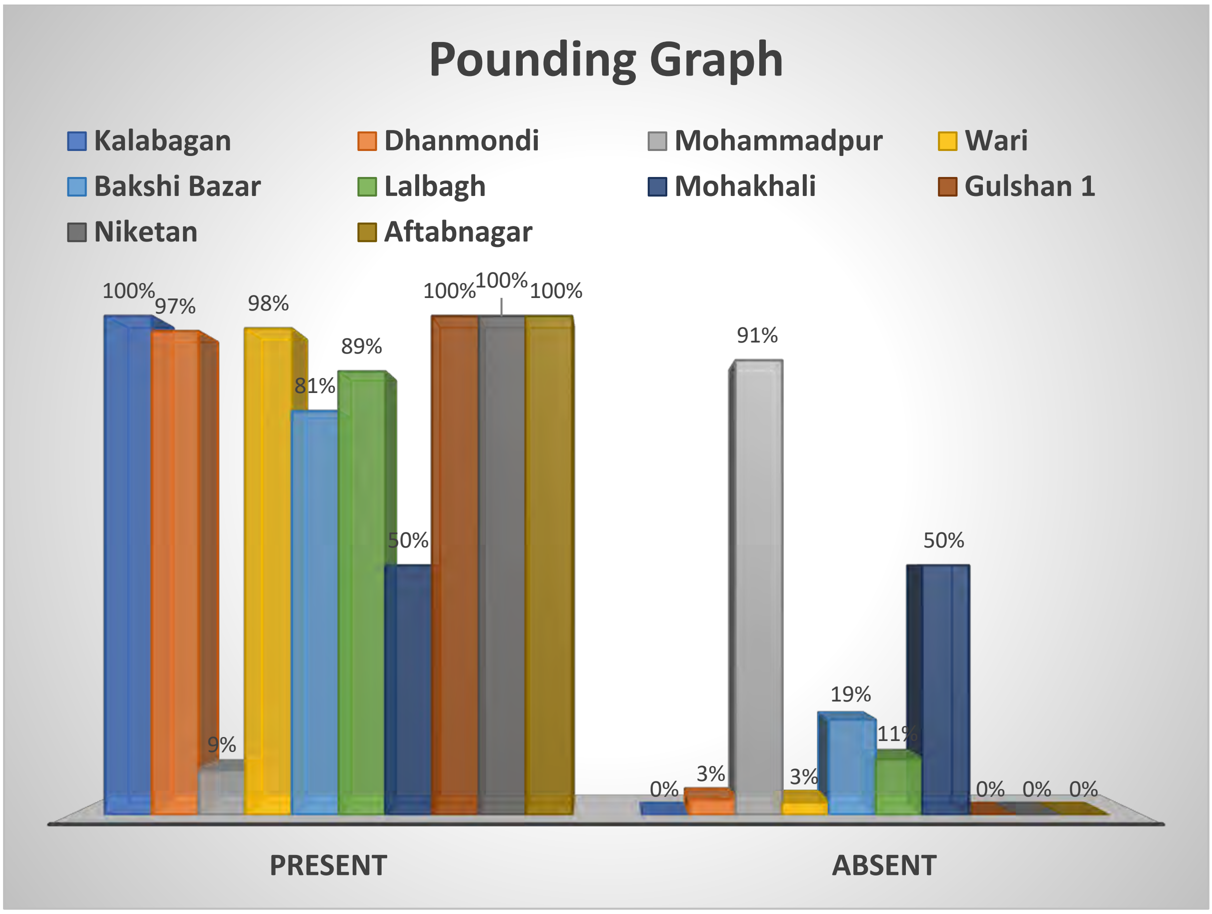


Figure 4.3 : Pounding rate of buildings in different area of Dhaka city.

Vertical and plan irregularities are both given bellow:

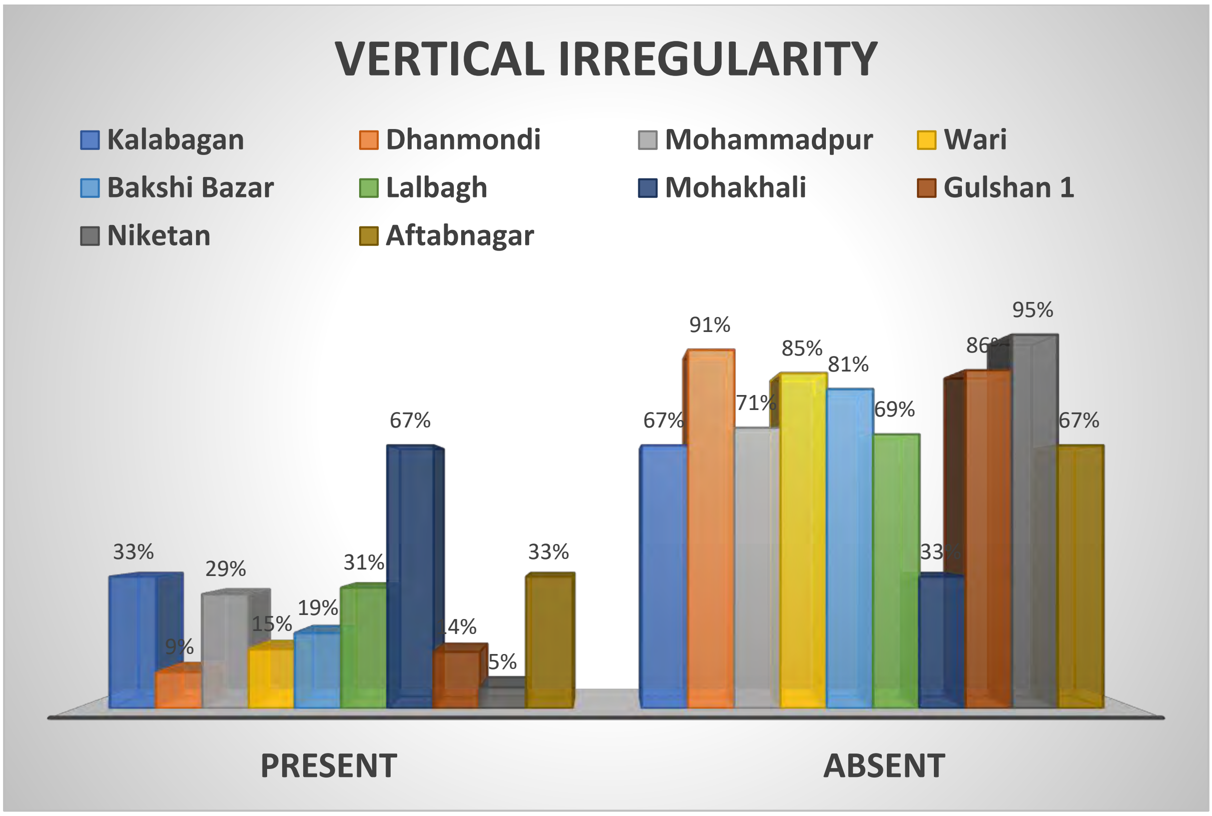


Figure 4.4 : Vertical irregularity bar of different zone of Dhaka respectively.

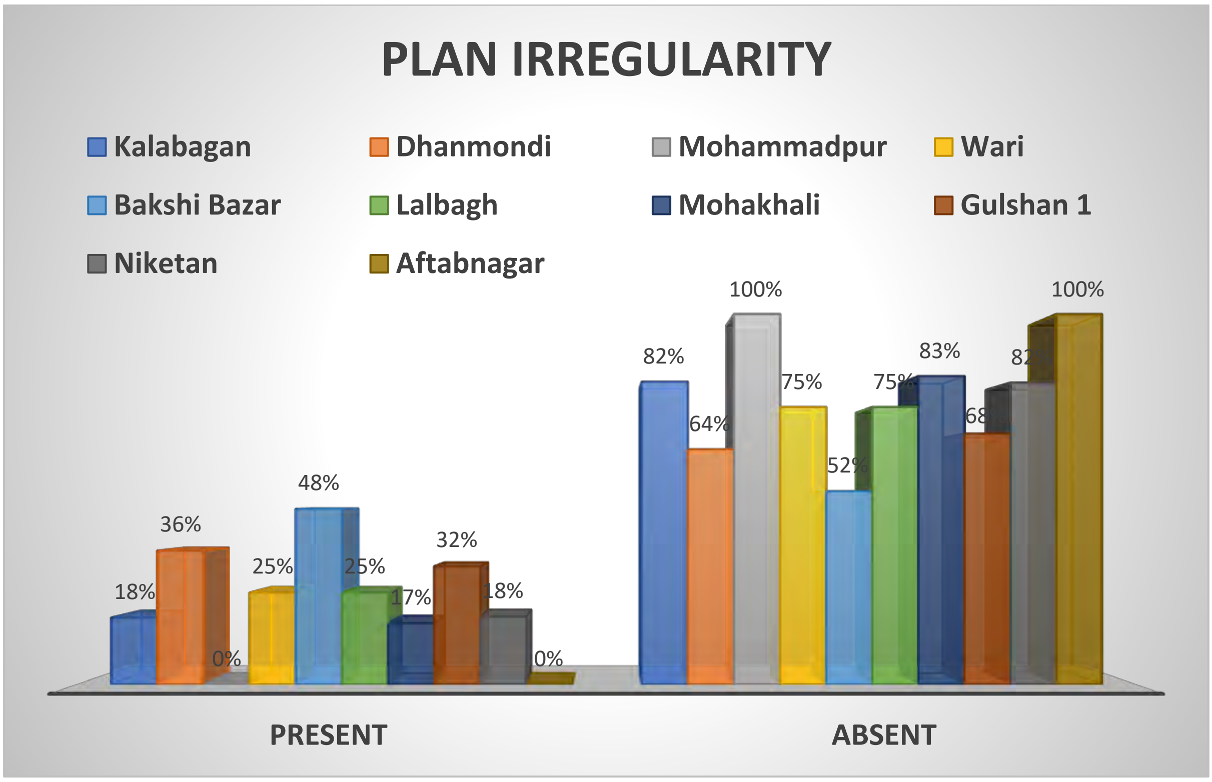


Figure 4.5 : Plan irregularity bar of different zone of Dhaka respectively.

4.3 SVM training

To train SVM we use a subset of our extracted data from the images. Since SVM is a supervised learning model which analyze data for classification and regression analysis. After feeding the data, SVM will build a model based on the training data. Then a large subset of data will used to test the SVM and output would be given based on training data.

4.4 CNN classification for beam recognition

Building data has pillar-beam pictures for each building of its parking lot. RC type buildings are supposed to have beam-pillar cross in its parking roof. However, some of the buildings do not have the beam-pillar cross. FEMA 154 tier 1, the civil engineering model that we are following for evaluating building's performance says that having beam-pillar cross is a must. If any building does not have the beam-pillar status true details evaluation is must for that particular building [as described in 4.1.2]. That means it is a risky building when there is no beam-pillar cross or the beam attribute is false.

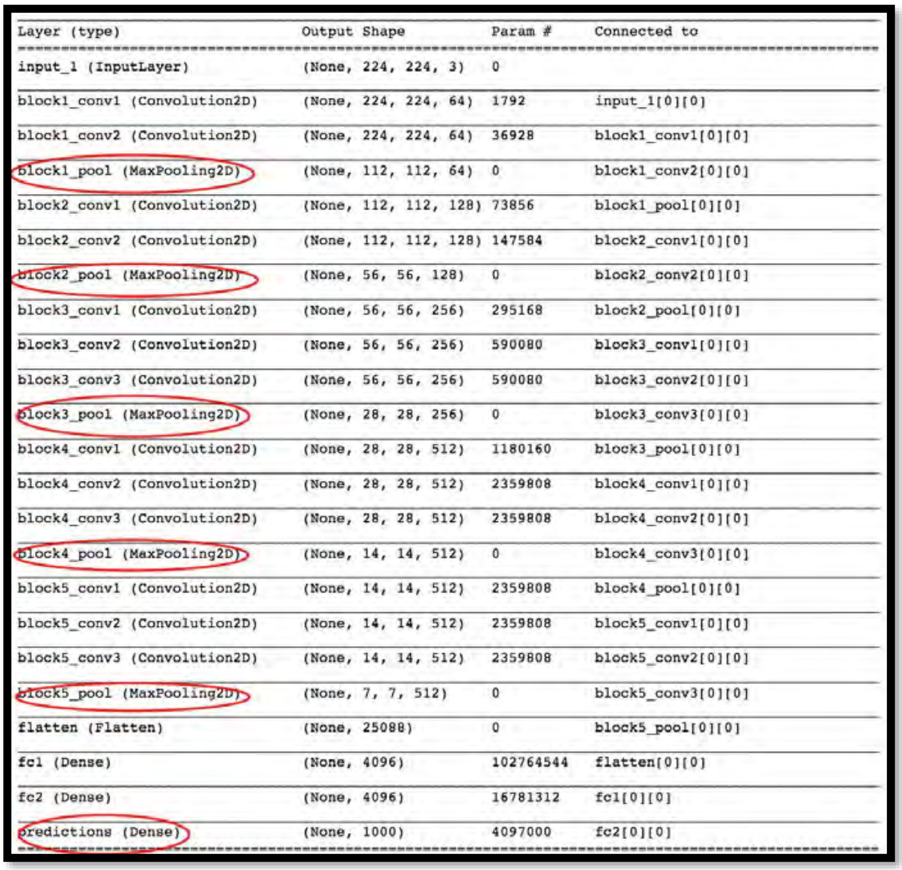
Although we did different image processing algorithm to find out different attributes of a building like pounding, vertical irregularity, plan irregularity and rise (high/mid), we used deep learning algorithm for beam attribute detecting. We discussed about CNN and VGG16 architectural model that is used in our model. We altered VGG16 model according to our data pattern and image type.

Alternation of VGG16 Architecture

1. In Keras we used "Sequential" model in spite of "Training" model of VGG16's data structure. We altered each layer of VGG16 and turned it into "Sequential" model. In sequential model, the result of a layer's training depends on the previous layer's result. Sequential model is straightforward and clear than any other data model. Therefore, we altered VGG16's each layer to Sequential.
2. Later, we cleared any previous learning of the model and made it ready for our image data. We removed any previous training memory and make the algorithm ready for beam recognition.
3. We altered the maxpooling layer from 2*2 to 4*4 matrix size in each convolution
4. VGG16 generally uses 1000 as the value of dense function. Dense function is used on fully connected step of CNN. VGG16 has 1000 outputs predefined. In our case, we

have two outputs; beam or no beam. We needed to change the last layer of VGG16 according to our dataset and expectation.

Our prediction result will give us the way of determination beam attribute to continue our process for evaluating building's strength for earthquake. If the result is "no-beam”, the particular building will need more result.



Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 224, 224, 3)	0	
block1_conv1 (Convolution2D)	(None, 224, 224, 64)	1792	input_1[0][0]
block1_conv2 (Convolution2D)	(None, 224, 224, 64)	36928	block1_conv1[0][0]
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0	block1_conv2[0][0]
block2_conv1 (Convolution2D)	(None, 112, 112, 128)	73856	block1_pool[0][0]
block2_conv2 (Convolution2D)	(None, 112, 112, 128)	147584	block2_conv1[0][0]
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0	block2_conv2[0][0]
block3_conv1 (Convolution2D)	(None, 56, 56, 256)	295168	block2_pool[0][0]
block3_conv2 (Convolution2D)	(None, 56, 56, 256)	590080	block3_conv1[0][0]
block3_conv3 (Convolution2D)	(None, 56, 56, 256)	590080	block3_conv2[0][0]
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0	block3_conv3[0][0]
block4_conv1 (Convolution2D)	(None, 28, 28, 512)	1180160	block3_pool[0][0]
block4_conv2 (Convolution2D)	(None, 28, 28, 512)	2359808	block4_conv1[0][0]
block4_conv3 (Convolution2D)	(None, 28, 28, 512)	2359808	block4_conv2[0][0]
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0	block4_conv3[0][0]
block5_conv1 (Convolution2D)	(None, 14, 14, 512)	2359808	block4_pool[0][0]
block5_conv2 (Convolution2D)	(None, 14, 14, 512)	2359808	block5_conv1[0][0]
block5_conv3 (Convolution2D)	(None, 14, 14, 512)	2359808	block5_conv2[0][0]
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0	block5_conv3[0][0]
flatten (Flatten)	(None, 25088)	0	block5_pool[0][0]
fc1 (Dense)	(None, 4096)	102764544	flatten[0][0]
fc2 (Dense)	(None, 4096)	16781312	fc1[0][0]
predictions (Dense)	(None, 1000)	4097000	fc2[0][0]

Figure 4.6 : Marked layers for alternating VGG16

4.5 Mapping

As we already know that, it is quite impossible to predict an earthquake and there is no established technology or methodology for that. Therefore, in that case we cannot prevent or take action beforehand. However, we can take necessary initiatives in order to minimize the chaos after an earthquake hit. For that reason, our mapping strategy comes in place as shown in the Figure [4.7]. Now, from the previous topics it is obvious that each and every building must maintain some code and conducts to make a building rigid and less affected to earthquakes. Unfortunately, in our country there are many cases where those codes are not being maintained during the building process. As a result, those buildings and the nearby area falls into the risk of being heavily damaged after an initial tremor of earthquake, which is a very concerning issue. As our climate is changing, hardly in a positive manner, the probability of a bigger earthquake hit is also increasing every day. For example, if we look at recent years, we can see earthquakes are occurring frequently. In addition, the population and the excessive building count made the building evaluation really hard. That is how we came up with this proposed methodology. We built a map, which will visually represent all the significant factors of a building and color-coded the buildings depending on its risk factors. In the process of building the map, we had to go through some major and minor phases. Some major processes are mentioned below.

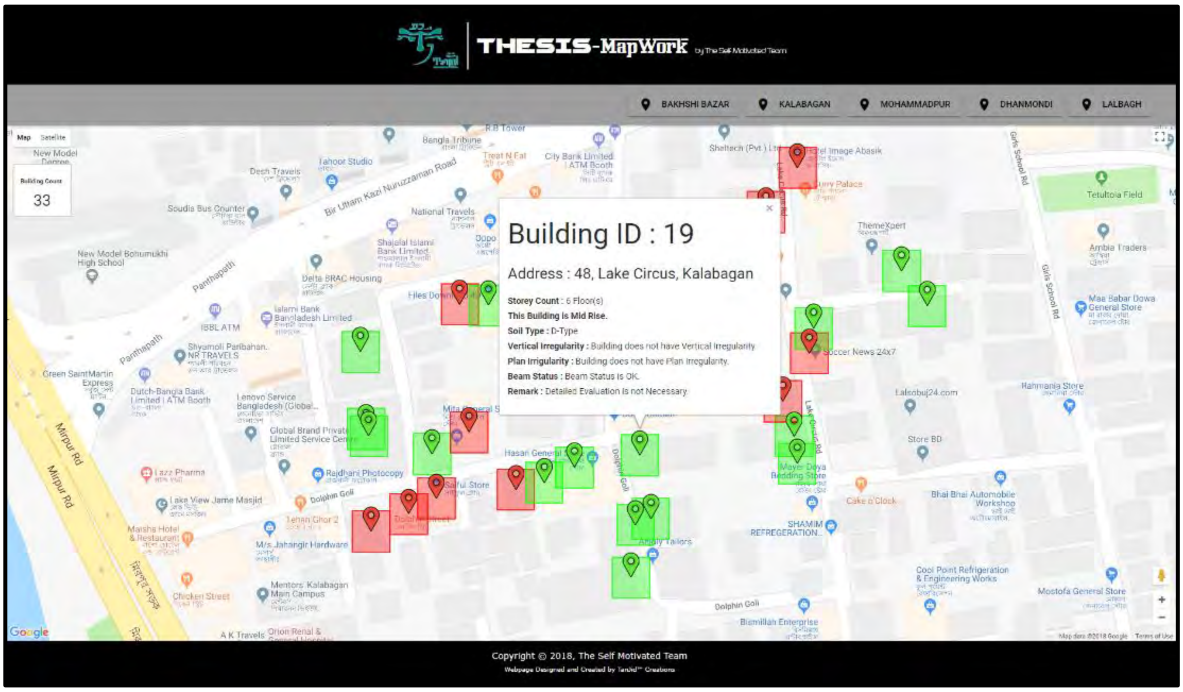


Figure 4.7 : Mapping Site.

4.5.1 Data Generation

Data generation for the Mapping portion of the project was the most crucial part. From the field work and the meetings with the Civil engineers we formed a strategy of collecting building data and images both physical and satellite. From them we put the images through Learning and testing via CNN (Convolutional neural network) and Ramer-Douglas-Peucker's Split and

Merge algorithm in order to determine the values of Pounding, Vertical Irregularity, Story and Plan irregularity. After that we computed the Final score and the Detailed Evaluation using some defined formula provided by FEMA and running SVM (Support Vector Machine) Algorithm for further verification. Shown in Figure [4.8].

Table 4.1 : Database Generation

ID	Address	Storey	Mid Rise	High Rise	Vertical Irregularity	Plan Irregularity	D Type	Pounding	Final Score	Beam Status	Detailed Evaluation
1	H#27, UD Road, Bakshi bazar	9	0	0.4	0	-0.5	-1	0	2.1	1	0
2	H#32 UD Road, Bakshi Bazar	7	0.2	0	-2	0	-1	0	0.4	0	1
3	31 Girdda Urdu Road, Bakshi Bazar	8	0	0.4	0	-0.5	-1	0	2.1	1	0
4	H#33A, Giridi Urdu, Bakhshi Bazar	5	0.2	0	0	0	-1	-0.5	1.9	1	1
5	H#7, Bakshi Bazar Lane	6	0.2	0	0	0	-1	0	2.4	1	0
6	4A, BakshiBazar Lane, Chalk Bazar	6	0.2	0	0	0	-1	0	2.4	1	0
7	4\K BakshiBazar Lane, Chalk Bazar	7	0.2	0	-2	0	-1	0	0.4	1	1
8	32 Bakshi Bazar Lane, Chalk Bazar	6	0.2	0	0	0	-1	0	2.4	1	0
9	16C, Bakshi Bazar Lane, Chalk Bazar	7	0.2	0	0	0	-1	0	2.4	1	0
10	103 Bakshi Bazar Lane, Chalk Bazar	6	0.2	0	0	0	-1	0	2.4	1	0
11	H#56-57 Hossain-e-Dalan Road	9	0	0.4	0	-0.5	-1	0	2.1	1	0
12	63A Hossain-e-Dalan Road, Ward-27	6	0.2	0	0	-0.5	-1	0	1.9	1	1
13	64A Hossain-e-Dalan Road	7	0.2	0	0	-0.5	-1	-0.5	1.4	1	1
14	H#14, Bakshi Bazar Road	9	0	0.4	0	-0.5	-1	0	2.1	1	0
15	131 Bakshi Bazar Road	7	0.2	0	0	-0.5	-1	0	1.9	1	1
16	132 Bakshi Bazar Road, Lal Bag	6	0.2	0	0	-0.5	-1	0	1.9	1	1
17	H#12 Bakshi Bazar Road, Lalbag	9	0	0.4	0	-0.5	-1	0	2.1	1	0
18	111B Nabab Bagicha, HD Road	6	0.2	0	0	0	-1	-0.5	1.9	1	1
19	9G Nabab Bagicha, HD Road	6	0.2	0	-2	0	-1	-0.5	-0.1	0	1
20	1 No Abul Hasnat Road	7	0.2	0	0	-0.5	-1	0	1.9	1	1
21	31 Girdda Urdu Road, Bakshi Bazar	8	0.2	0	-2	0	-1	0	0.4	1	1

4.5.2 Mapping Process

4.5.2.1 Database Management

After Database is finally being generated and verified, then we feed the final database to our mySQL server by converting our Excel database to mySQL database format as shown in Figure [4.9]. With the database initiated in the server it's ready to apply in our mapping phase.

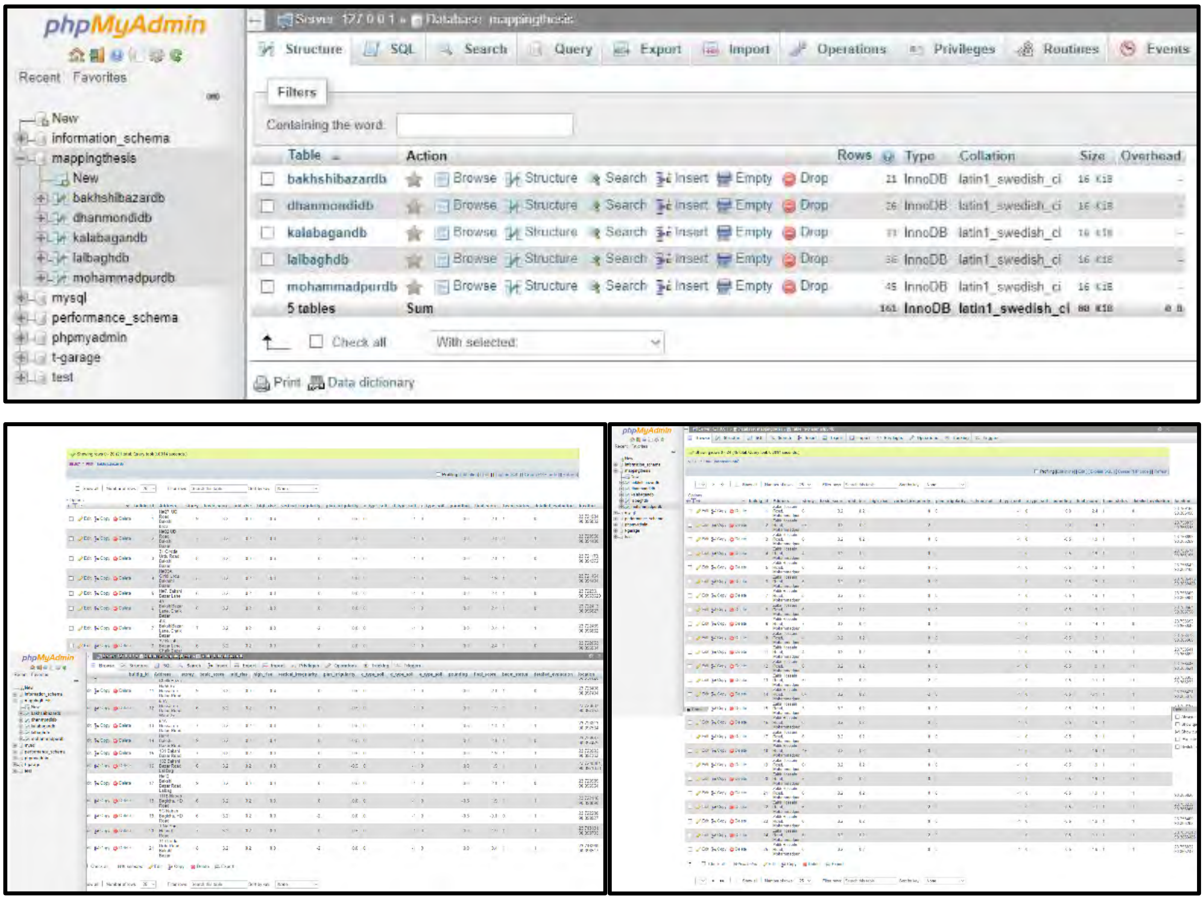


Figure 4.8 : Porting Generated Database mySQL Server

4.5.2.2 Tools and Resources

We built our webpage for the map representation using HTML[22], CSS[23] and Javascript[21] for the front-end UI, PHP[20] and mySQL for the back-end property. In order to create a virtual PHP and mySQL server we used XAMPP[24] and used custom ports for building and testing the final webpage. Google Map API[14] came handy while initiating and customizing the map's properties for our convenience.

4.5.2.3 Initiation

Using Google Map API[15] we placed custom markers and rough building object over the existing live Google Map. Markers and the building has different color codes basing on the generated database and its attributes. Every marker represents a separate building of our sample area and they have different characteristics. Hovering on the marker shows the building ID and Clicking or selecting a marker shows the properties of that building basing on the database and provides recommendation for that building.

4.5.3 Mapping Advantages

This mapping will open a completely new window to the people from various field. This mapped site will be published online, which will be accessible to everyone who needs it. For example, Fire Service, Civil Engineering Personals, RAJUK, Medical Personals, Government Officials etc. In addition, normal peoples can preview their living place's current condition by visiting this real time map.

Chapter 5

Experimentation and Result

5.1 Image Processing for Building's Heights and gap calculation

In cities like Dhaka measuring the heights and gap between buildings are so difficult because of less gaps between buildings and color complexity in the image. To begin with, the data needed to be taken from a fix and known distance, and the range of length should be known. Now, image needed to be processed into grayscale image. Then, the Topmost, bottom most, leftmost, rightmost images points are gained.

These all point of the building, which needs to be marked with a red color for tracing. However, RGB objects on the gray image are placed for color segmentation using k-mean cluster. Using the nearest neighbor algorithm, color of the red objects is being masked. After that, adaptive threshold based binarization has been applied to make the image binary. The binary image has only two parts of pixels' values now, white or black. Based on the theorem of moment of image is applied to recognize regions. The border values are stored and color is defined as RGB value. And also, the center value using moments of image can be gained. Now, based to distance of building's top to bottom or one building's left post point to right for gap calculation.

$D_{is} = (x1 - x2 + (y - y2))$ -------------------------------------- (5.1)

RESULT

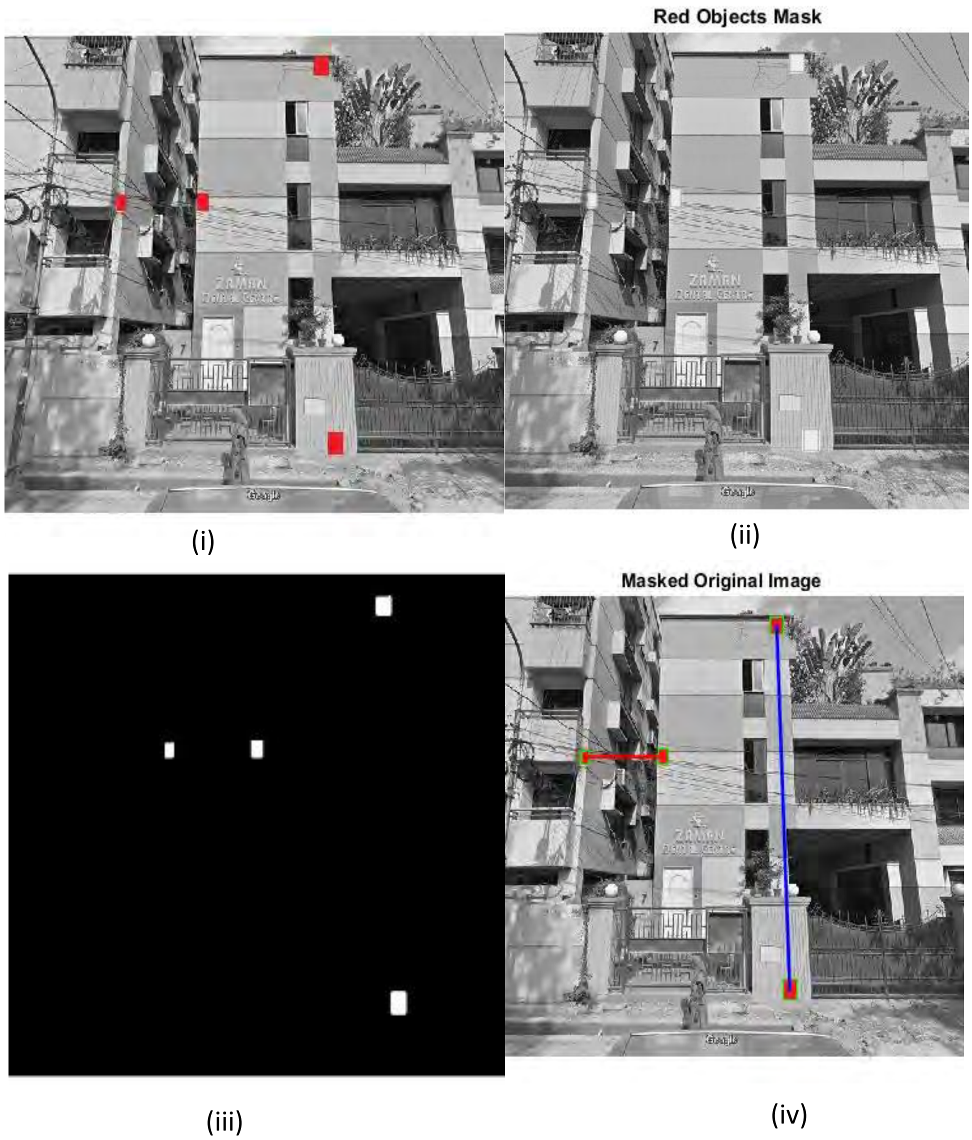


Figure 5.1: (i) grayscaled with marking, (ii) Red masked color (iii) Binarization, (iv) Calculating distance joining The centers.

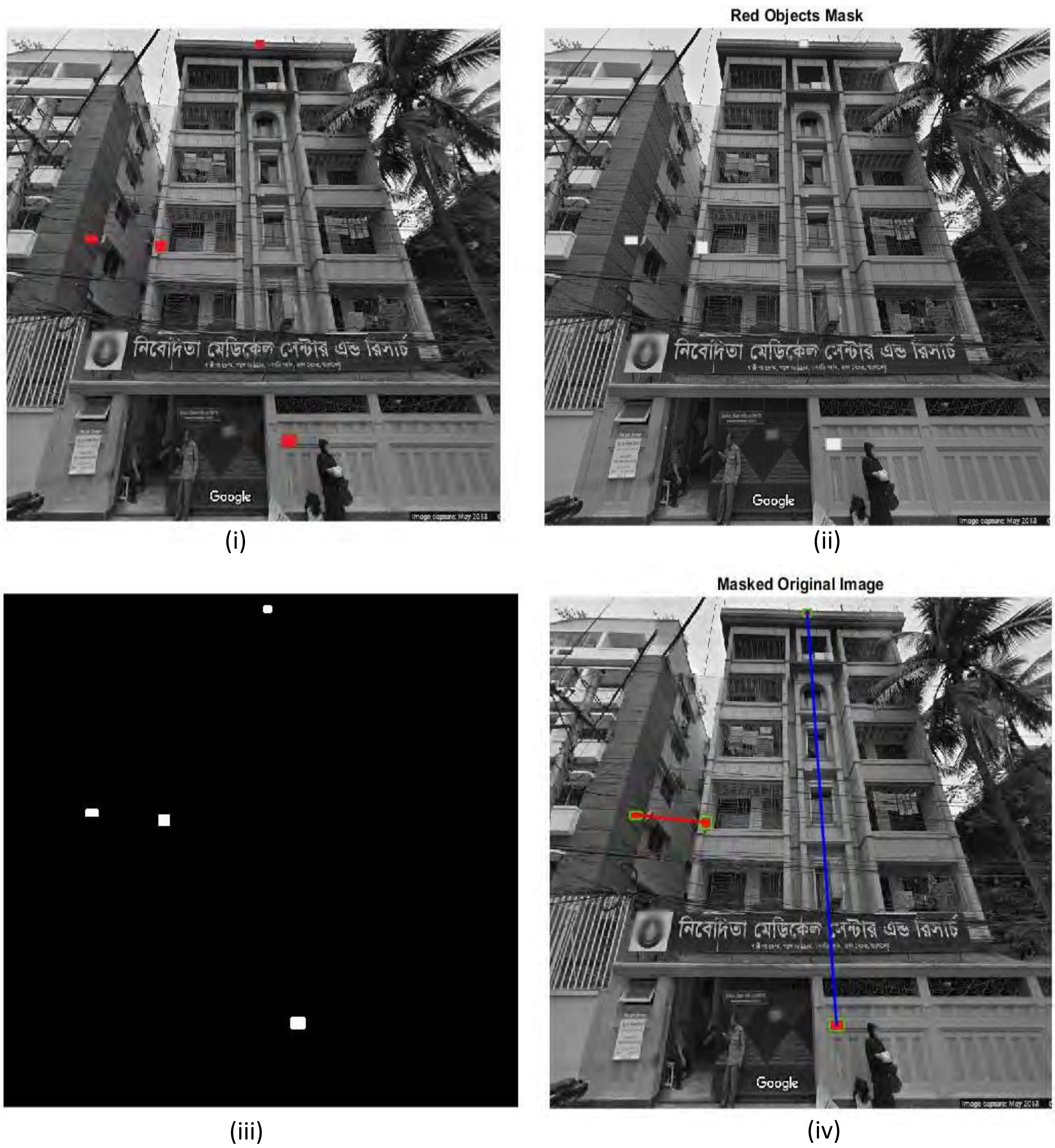


Figure 5.1: (i) grayscaled with marking, (ii) Red masked color (iii) Binarization, (iv) Calculating distance joining The centers.

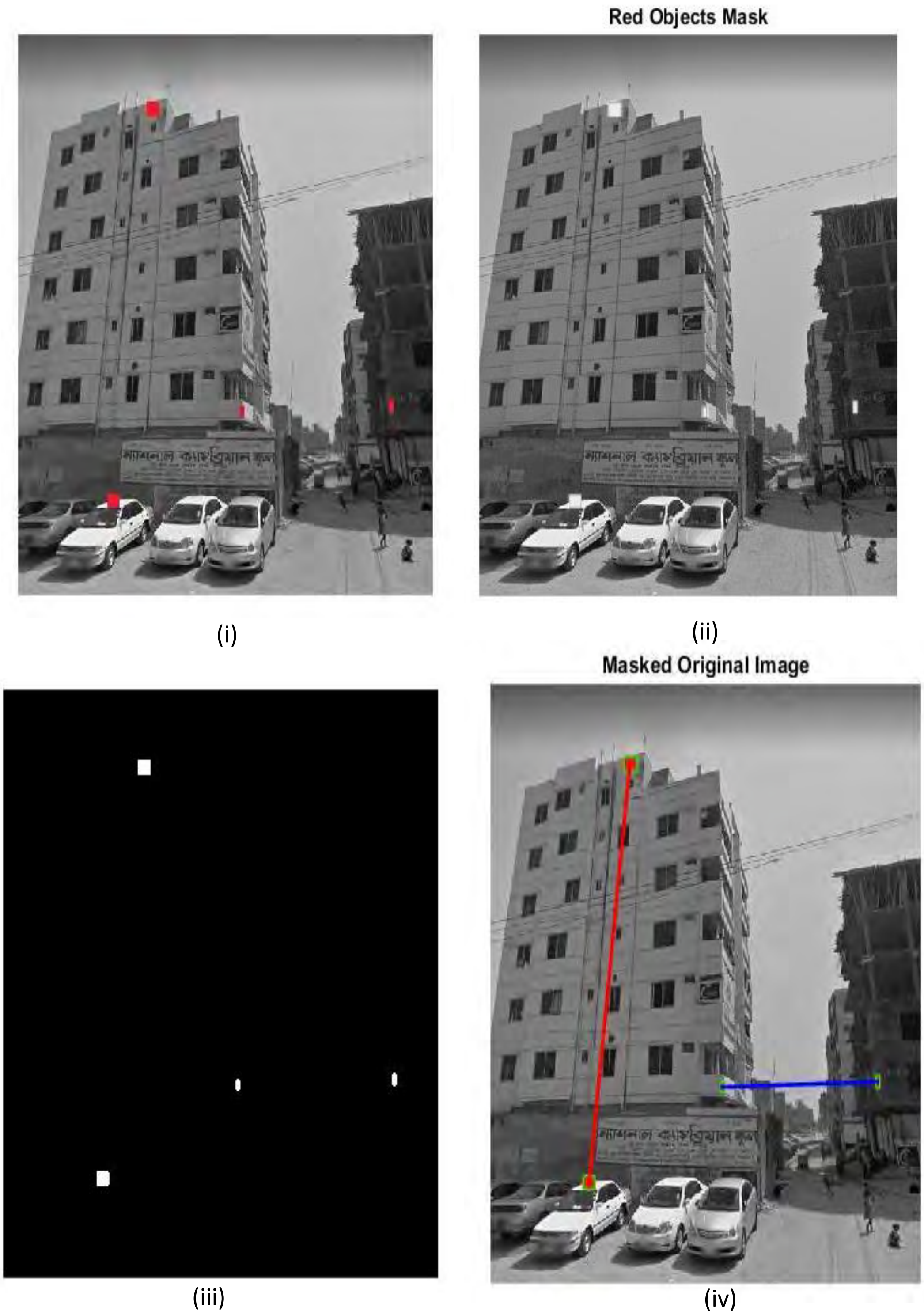


Figure 5.1: (i) grayscaled with marking, (ii) Red masked color (iii) Binarization, (iv) Calculating distance joining The centers.

The unit of distance is in pixel, and need to convert in meter. By the known distance from the building and resolution of image we can measure per pixel meter unit. As a result, real value can be shown.

5.2 Plan Irregularity

Shape based analysis is implemented to a satellite image to clarify whether the build's shape is rectangular or square as it facts in earthquake study. In order to perform shape detection, we will be using contour approximation, which is the implementation of Ramer Douglas Peucker algorithm. Contour approximation is predicated on the assumption that a curve can be approximated by a series of short line segments. This leads to a resulting approximated curve that consists of a subset of points that were defined by the original curve. Contour approximation is actually already implemented in OpenCV via the cv2.approxPolyDP method. In order to perform contour approximation, we first compute the perimeter of the contour followed by constructing the actual contour approximation. Common values for the second parameter to cv2.approxPolyDP are normally in the range of 1-5% of the original contour perimeter.

If a contour has four vertices, then it must be either a square or a rectangle. To determine which, we compute the aspect ratio of the shape, which is simply the width of the contour bounding box divided by the height. If the aspect ratio is ~1.0, then we are examining a square (since all sides have approximately equal length). Otherwise, the shape is a rectangle.

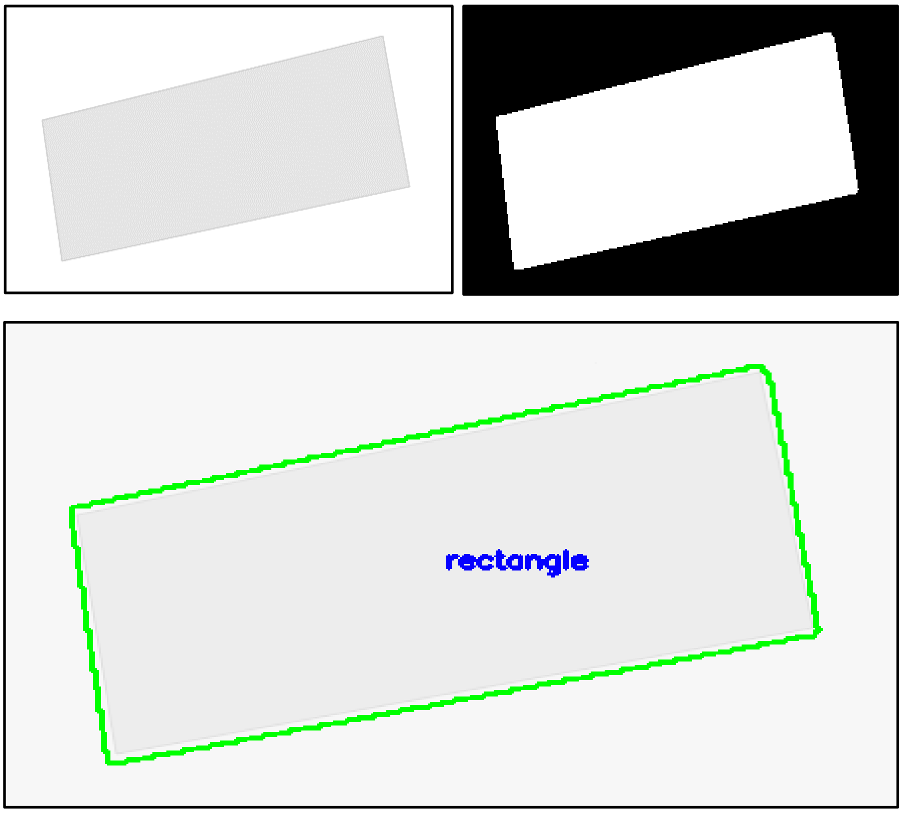


Figure 5.2 : (i) Image from google-map (ii) Binarization of that image (iii) Rectangle shape detected image

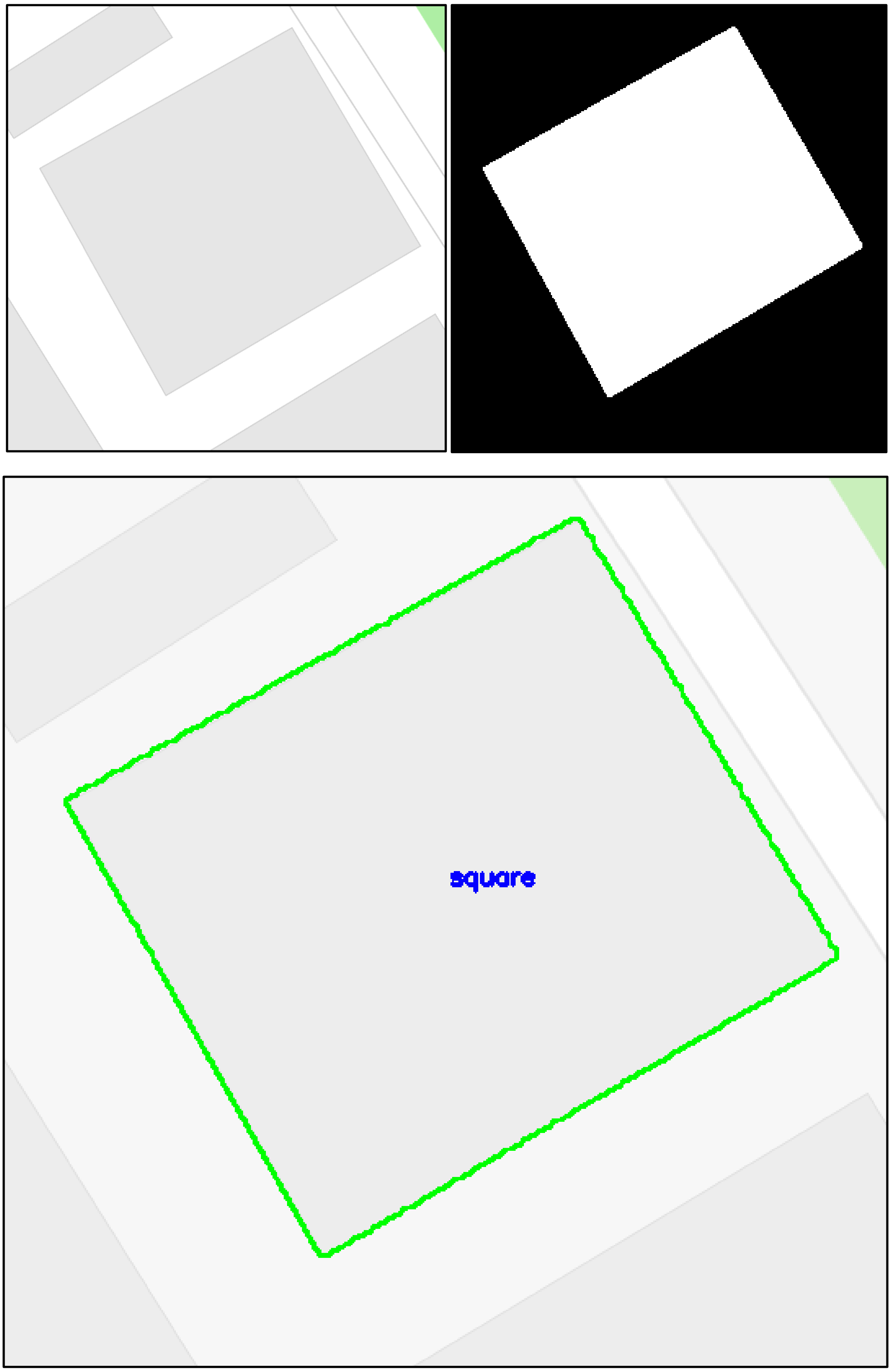


Figure 5.3 : (i) Image from google-map (ii) Binarization of that image (iii) Square shape detected image

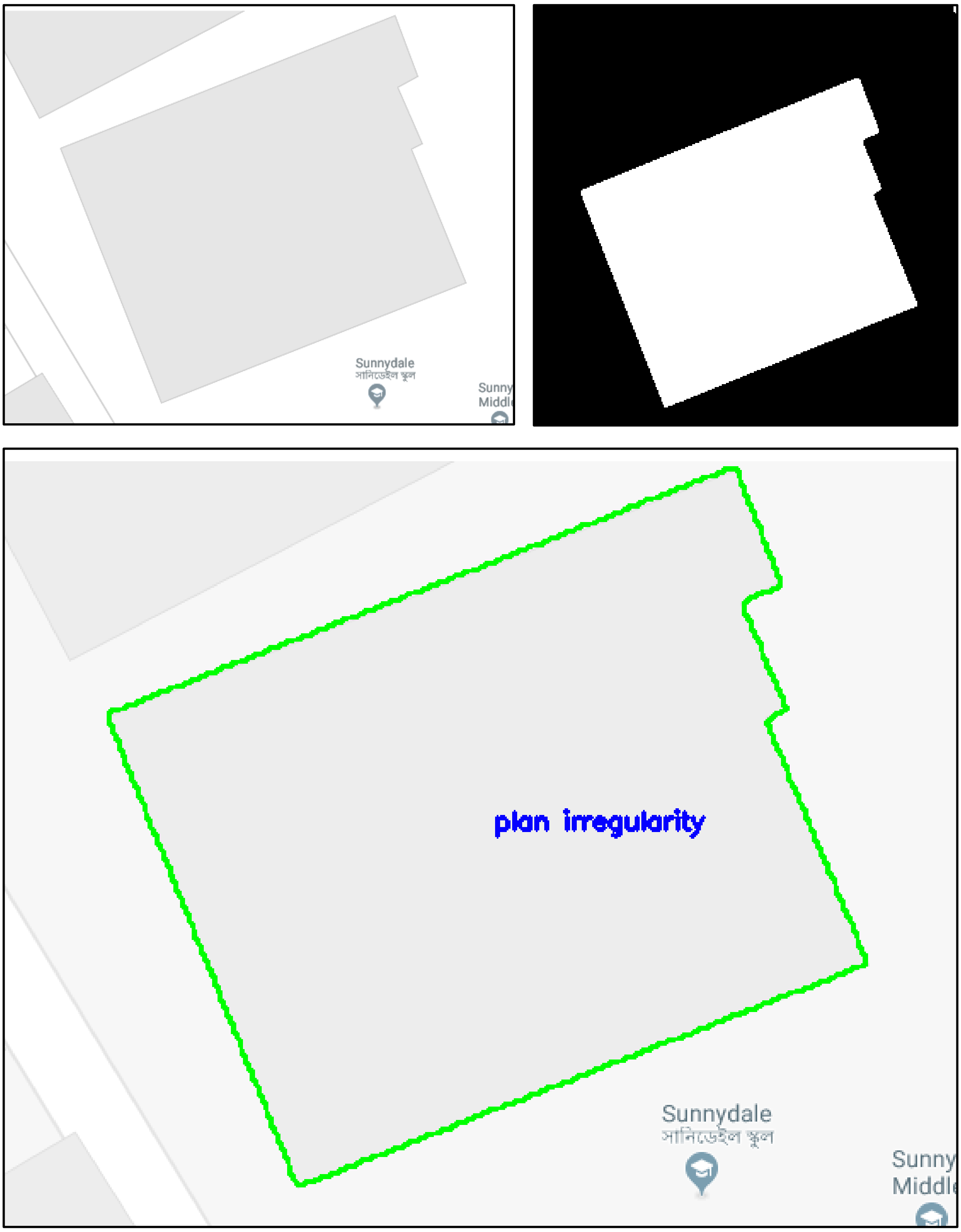


Figure 5.4 : (i) Image from google-map (ii) Binarization of that image (iii) Plan irregularity detected image

5.3 Altered VGG16 architectural model of CNN algorithm

Coding and Environment:

We coded in python language in anaconda environment of python3.6 version. We used Keras model with proper packages and Tensorflow backend. Keras is a high level API to build and train deep neural networks. In Keras we can work with either tensorflow, theano or CNTK backend. Among these well accepted backend techniques we used Tensorflow as it has first class functionality such as eager execution, data pipeline and estimators without sacrificing flexibility and performance. Though there are many worldwide known usage of Theano backend, it is declared by its developer that it won't be updated after 2017's model. So, we considered Tensorflow over other options without any doubt. In Keras we can use different models which are basically the assembly of different layers. We used sequential layer model which is basically a stack of layers in a fully connected network.

First test:

For training 30 images of with beam and 20 images of without beam, images are trained. We validated 10 images for each of the classes. We didn't use Dropout() function as our data is less. Thus, our training and loss function curves has ups and downs. We have got around 73 percent of learning accuracy with such few data with only 5 epochs.

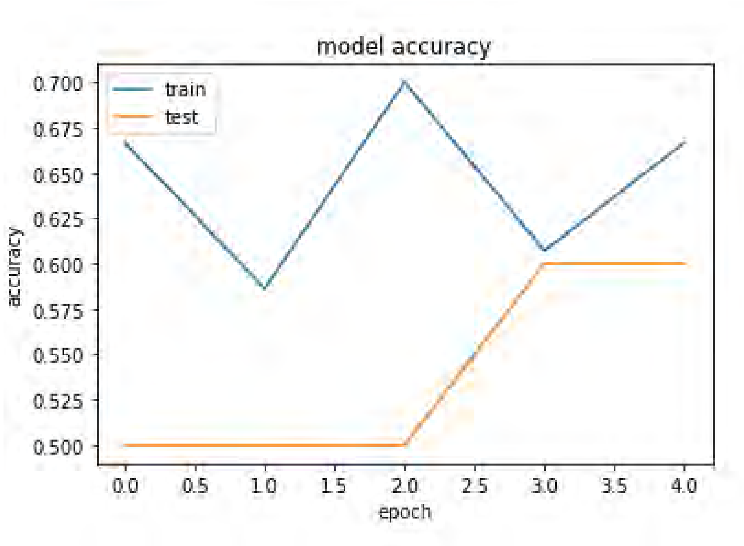


Figure 5.5 : Accuracy rate model of learning [both test and train data].

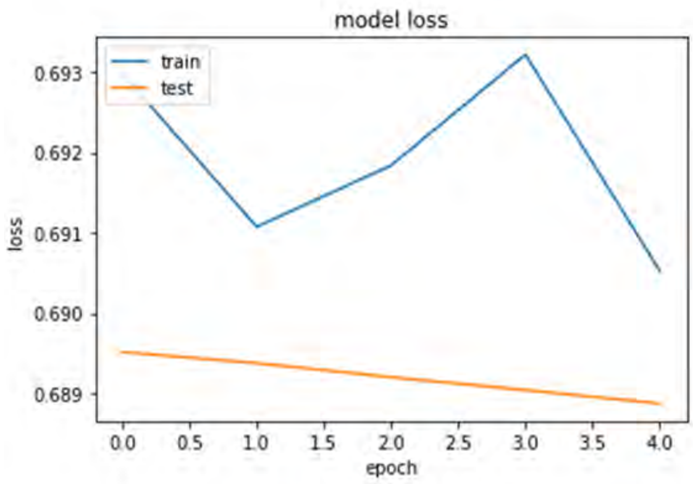


Figure 5.6 : Loss rate of learning [both test and train data].

Second test:

We enrich out limited data set with some more images. This time to get better learning curve for "test set" we trained on 50 images with beam and 28 images without beam. We validated our model on 19 images with beam and 12 without beam. This time for 200 epochs, the result is below with 80% of validation accuracy:

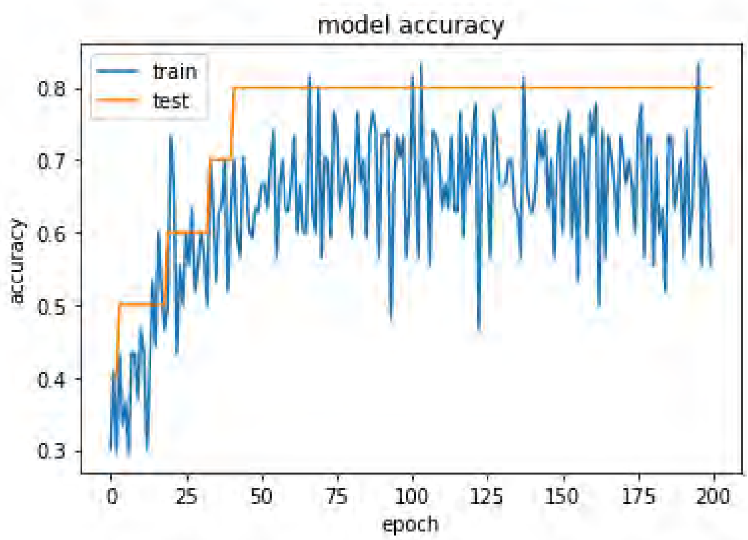


Figure 5.7: Model accuracy curve for 200 epochs.

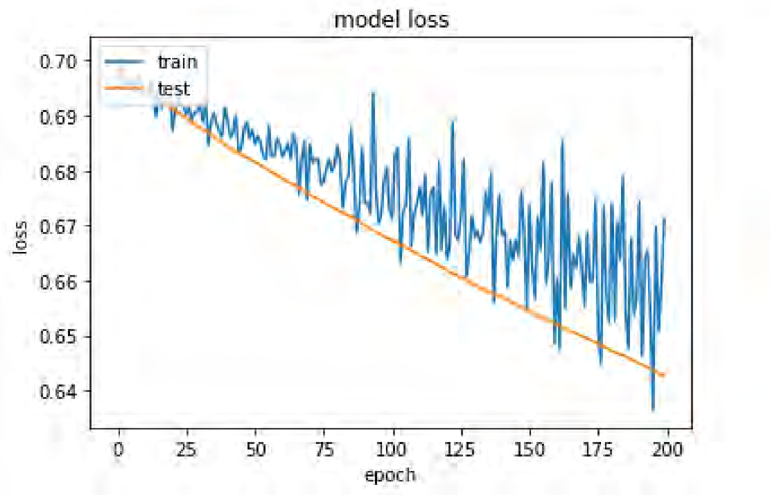


Figure 5.8: Model loss curve for 200 epochs.

We predicted/evaluated our result on 8 images consisting of 5 images with beam and 3 without beam. We labeled beam images with 0 and no beam images with 1. That gives us accuracy rate of 75%.

Table 5.1 : Accuracy table for VGG16 altered architecture.

	Beam	No beam	Total
True numbers	5	3	8
Correct Prediction	4	2	6
Total accuracy			6/8*100 = 75%

5.4 Support Vector Machine result

Finally, to evaluate the risk value of buildings we used MATLAB® version 14

Simulation tools. We have total 283 building data in which we use 200 data for training and 83 data for test. A portion of training set and test set has been shown in figure [5.2, 5.3]. As, SVM is a supervised learning model so that we labelled our data to conduct train and test our data. Here, SVM will train and test based on six attributes.

Table 5.2 : SVM training set.

Rise	**Vertical irregularity**	**Plan irregularity**	**Soil type**	**Pounding**	**Beam status**	**Detailed Evaluation**
0.2	0	0	-1.6	0	1	1
0.4	-2	0	-1.6	0	0	1
0.4	0	-0.5	-1	0	1	0
0.2	0	0	-1	0	1	0
0.4	0	0	-1.6	0	1	0
0.2	-2	0	-1	-0.5	0	1
0.4	0	0	-1	-0.5	1	1

Table 5.3 : SVM test set.

Rise	**Vertical irregularity**	**Plan irregularity**	**Soil type**	**Pounding**	**Beam status**	**Detailed Evaluation**	**Result**
0.2	0	-0.5	-1	0	1	1	0
0.4	0	-0.5	-1	0	1	1	1
0.2	0	0	-1	0	1	0	0
0.2	0	0	-1	0	1	0	1
0.2	-2	0	-1	0	1	1	1
0.2	0	0	-1	0	1	1	1
0.4	-2	0	-1.6	-0.5	1	1	1

In SVM test, we have found 78 test cases are correct comparing to our manually evaluated data. SVM error rate is 6.024%, which is shown in figure [5.6].

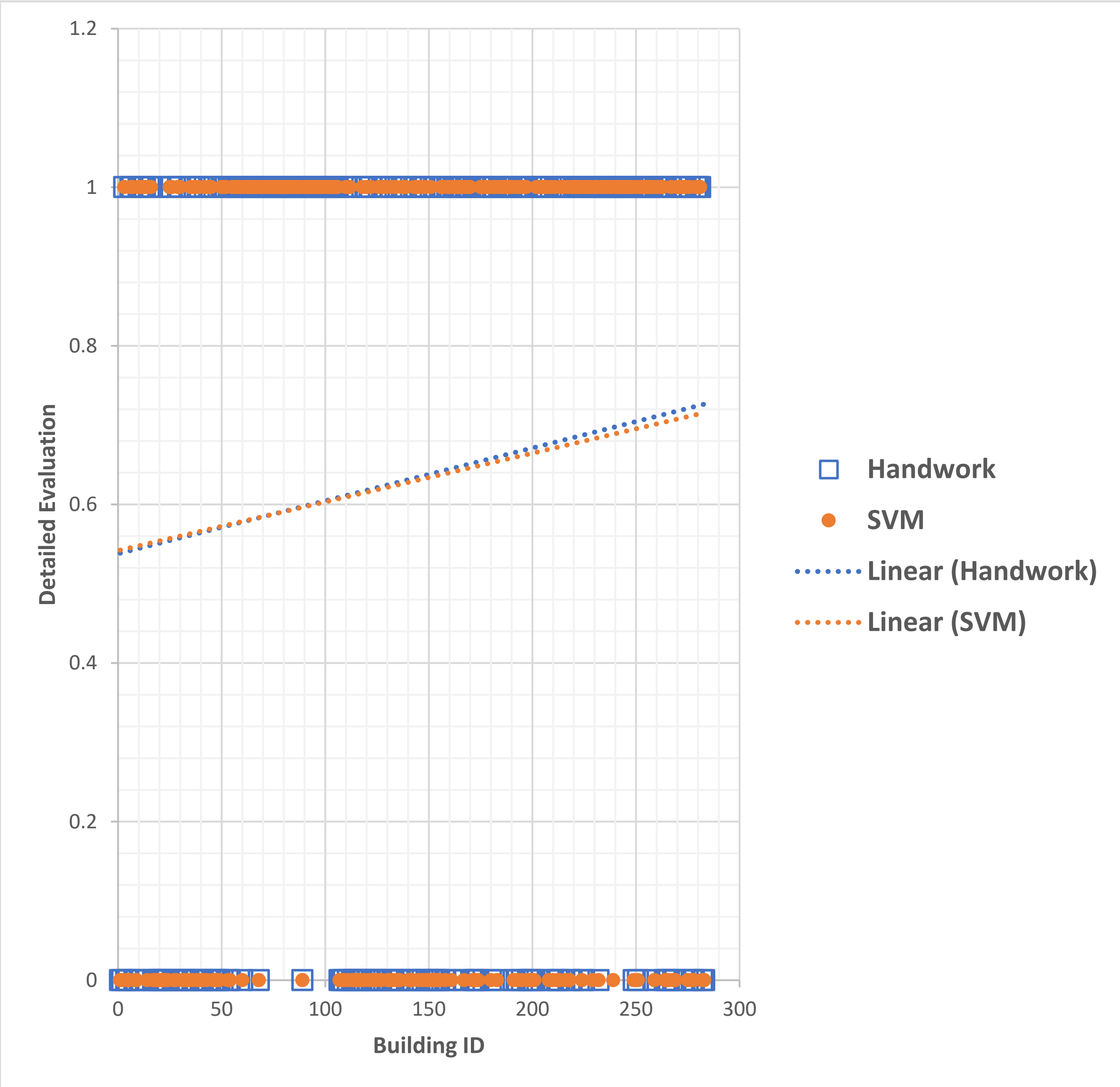


Figure 5.9 : SVM Accuracy Graph.

Chapter 6

Conclusion

Working for earthquake by taking building's picture and applying Machine Learning and Image Processing on it is completely a new process. We tried our best to cover the all the criteria for Tier-1 phase of calculating risk factors of buildings by following FEMA154 method. Our processed images gives us the answer whether the building has pounding, vertical irregularities, plan irregularities or not. Moreover, we trained for detecting beam through CNN. We hope and believe that this research will help to bring out something great in this field of earthquake safety in near future.

6.1 Challenges

Analyzing building risk using machine learning and image processing is quite new. To conduct this research, we needed a large number of dataset but we could not be able to get that number of data set from any other source and usually to perform this type of research analysts produces their own dataset. Therefor we needed to produce our own data. Without the proper length of time it was really difficult for us to collect that number of data. However, we manage to collect our data in very challenging ways.

One of the most difficulties we faced was to get the permission to collect the images of the corresponding subject. In some situations, people were concerned that where the images will be used and it might be used in some problematic way for themselves and we had to convince them it was purely for the research purpose. When we were done with capturing, another problem arose and that was, as we took images with mobile phone and some of the pictures are not appropriate. Because, in some cases, we could not manage to fit the whole building in an image so we had to take a panorama of these buildings but while taking panorama, sometimes the image got cut through a portion and sometimes the sunlight spoilt the upper portion of the image. Therefore, we were bound to throw away many pictures and again started taking pictures with perfect view. Hence, we managed to collect the dataset with lots of difficulties.

6.2 Future Plan

Our motto was to map buildings of Dhaka city with the attributes of its risk factor. Our work has many ways to upgrade and work on further progress of earthquake resistance. We can drag many aspects of urban planning that is important for earthquake. Particularly our future plans are:

1. Map entire city with the cluster of areas with risk assessment.
2. Fitting the neural network with more image data and make it more accurate.
3. Upgrading and smoothing algorithms for giving more accuracy with detecting building faults by images.
4. Training our model with future earthquake rate and damages to learn how much can a building be affected in particular magnitude of earthquake.
5. We worked with FEMA 154 Tier 1 altered model for Dhaka city's building. We will work on Tier 2 and 3, which have more attributes to work with than Tier 1.

References

1. J. Kumagai, “A Brief History of Earthquake Warnings,” *IEEE Spectrum: Technology, Engineering, and Science News*, 01-Jun-2007. [Online]. Available: https://spectrum.ieee.org/at-work/innovation/a-brief-history-of-earthquake-warnings. [Accessed: 02-Jan-2018]
2. “USGS Earthquake Hazards Program,” *U.S. Geological Survey*. [Online]. Available: https://earthquake.usgs.gov/. [Accessed: 10-Feb-2018]
3. “Earthquake History of Bangladesh,” *Assignment Point*, 05-Jan-2017. [Online]. Available: http://www.assignmentpoint.com/arts/modern-civilization/earthquake-history-of-bangladesh.html. [Accessed:10-Feb-2018]
4. “Satellites will soon predict earthquakes and volcanic eruptions from space,” *Futurism*, 06-Sep-2017. [Online]. Available: https://futurism.com/satellites-will-soon-predict-earthquakes-and-volcanic-eruptions-from-space/. [Accessed: 10-Feb-2018]
5. R. Reitherman, “International Aspects of the History of Earthquake Engineering,” thesis, 2008.
6. “Site Classification and Seismic Response of Dhaka City Soils.” [Online]. Available: http://lib.buet.ac.bd:8080/xmlui/bitstream/handle/123456789/3140/. [Accessed: 05-May-2018].
7. “Earthquake Vulnerability Assessment of Existing Buildings in Cox’s-Bazar using Field Survey & GIS”, (n.d.).
8. “Rapid Visual Screening of Buildings for Potential Seismic Hazards,” Conference Tutorials - 1906 Earthquake Conference, San Francisco. [Online]. Available: http://www.1906eqconf.org/tutorials/. [Accessed: 01-Apr-2018].
9. “Convolutional Neural Networks (CNN) Introduction,” *Algobeans*, 22-Sep-2016. [Online]. Available: https://algobeans.com/2016/01/26/introduction-to-convolutional-neural-network/. [Accessed: 15-Mar-2018]
10. F.-F. Li, J. Johnson, and S. Yeung, “CNN Architectures,” in *Lecture 9*.
11. D. K. Srivastava and L. Bhambhu, “Data Classification Using Support Vector Machine,” *Journal of Theoretical and Applied Information Technology*, 2009.
12. Saravanan M., Kalaivani B. and Geethamani R., “Image Segmentation Using K-Means Clustering Based Thresholding Algorithm,” *International Journal of Advanced Technology in Engineering and Science*, vol. 5, no. 2, Feb. 2017.

13. “Google Maps Platform - Geo-location APIs | Google Maps Platform | Google Cloud,” *Google*. [Online]. Available: https://cloud.google.com/maps-platform/. [Accessed: 19-Apr-2018]
14. *Google Maps*. [Online]. Available: https://developers.google.com/maps/documentation/javascript/. [Accessed: 18-Feb-2018]
15. “Google Maps Documentation,” *Google*. [Online]. Available: http://code.google.com/apis/maps/documentation/. [Accessed: 18-Feb-2018]
16. M. C. Brown, *Hacking Google Maps and Google Earth*. Indianapolis: Wiley Publishing, ISBN: 978-0-471-79009-9, 2006.
17. F. Rousseaux and K. Lhoste, “Rapid Software Prototyping Using Ajax and Google Map API,” *2009 Second International Conferences on Advances in Computer-Human Interactions*, 2009.
18. “Introducing Google Maps,” *Beginning Google Maps Applications with PHP and Ajax*, pp. 3–11.
19. Y.-L. Zhou and Z.-L. Fu, “Release of spatial information based on Google Maps API,” *Journal of Computer Applications*, vol. 31, no. 5, pp. 1450–1452, 2011.
20. “Documentation,” *php*. [Online]. Available: http://php.net/docs.php. [Accessed: 24-May-2018]
21. R. Nixon, *Learning PHP, MySQL, and JavaScript: A Step-By-Step Guide to Creating Dynamic Webpages*. Beijing: OReilly, ISBN: 978-0-596-12713-5, 2009.
22. “DevDocs,” *DevDocs - Bootstrap 4 documentation*. [Online]. Available: http://devdocs.io/html/. [Accessed: 24-May-2018]
23. J. Duckett, *HTML & CSS Design and Build Websites*. Crosspoint Boulevard, IN: John Wiley & Sons, ISBN: 978-1-118-00818-8, Inc., 2011.
24. “Joomla! Documentation™,” *Portal:Administrators - Joomla! Documentation*. [Online]. Available: https://docs.joomla.org/XAMPP. [Accessed: 28-Mar-2018]
25. Theodosios Pavlidis, A review of algorithms for shape analysis, Computer Graphics and Image Processing, Volume 7, Issue 2, 1978, Pages 243-258, ISSN 0146-664X, https://doi.org/10.1016/0146-664X(78)90115-6.
26. Y. M. Mustafah, R. Noor, H. Hasbi and A. W. Azma, "Stereo vision images processing for real-time object distance and size measurements," 2012 International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, 2012, pp. 659-663.doi: 10.1109/ICCCE.2012.6271270

27. Young Won Lim, Sang Uk Lee, On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques, Pattern Recognition, Volume 23, Issue 9, 1990, Pages 935-952, ISSN 0031-3203, https://doi.org/10.1016/0031-3203(90)90103-R.
28. “Share and discover research,” ResearchGate. [Online]. Available: https://www.researchgate.net/figure/. [Accessed: 11-Mar-2018]
29. R. Islam, M. N. Islam, and M. N. Islam, “Earthquake Risks in Bangladesh: Causes, Vulnerability, Preparedness and Strategies for Mitigation,” ARPN Journal of Earth Sciences, vol. 5, no. 2.
30. “Line Simplification,” Fisher–Yates Shuffle. [Online]. Available: https://bost.ocks.org/mike/simplify/. [Accessed: 02-May-2018].