# SURFACE CRACK DETECTION ON USING IMAGE PROCESSING

***A Project report submitted in partial fulfilment of the requirements for***

***the award of the degree of***

**BACHELOR OF ENGINEERING IN**

MECHANICAL ENGINEERING

**Submitted by**

**MIRAJUL HOQUE (14200719059) SHOUVIK KHAN (14200719057) SATTRAJIT SAHA(14200719061) SOMNATH DAS(14200719063)**

**PARTHAJIT RANA(14200719060) PANKAJ KUMAR(14200719062)**

**

*Under the guidance of*

ANIRBAN BOSE

*ASSISTANT PROFESSOR*

Meghnad Saha Institute Of Technology

Table of Contents

ABSTRACT........................................................................................................................................................3LIST OF FIGURES: ..........................................................................................................................................3 **LIST OF CHAPTER** 1:INTRODUCTION..................................................................................................... 4

1.1 Problem Definition........................................................................................................................... 5 1.2 Motivation ........................................................................................................................................5 1 1.3 Objectives...................................................................................................................................... 6

CHAPTER 2: RELATED WORKS............................................................................................................ 7

2.1 CRACK CLASSIFICATION..................................................................................................................7

CHAPTER 3:PROPOSED ALGORITHM...................................................................................................9

3.1 GAUSSIAN FILTERING..........................................................................................................9

3.2 CANNY EDGE DETECTION...................................................................................................9

3.3 MORPHOLOGICAL APPROACH............................................................................................10

3.4 PARAMETER ESTIMATION ..................................................................................................10

CHAPTER4:DATASETS........................................................................................................................... 10

CHAPTER 5: MODEL BUILD...................................................................................................................12

CHAPTER6:EXPERIMENTS.....................................................................................................................14

6.1 DATA PREPARATION..........................................................................................................................14

6.2 ACCURACY & VAL LOSS...................................................................................................................14

6.3 EXPLORATORY ANALYSIS...............................................................................................................................................15

6.4 TRAINING AND TESTING...................................................................................................................15

CHAPTER 7: EVALUATION MATRIX.....................................................................................................................................................15

7.1CONFUSION MATRICES.................................................................................................................................................15

7.2ACCURACY.............................................................................................................................15

7.3RECALL & PRECISION..........................................................................................................16

CHAPTER8:DISCATION&RESULT........................................................................................................16

8.1 EXPERIMENTAL RESULT & ACCURACY...............................................................................................................................................17

CHAPTER 9: CODE...................................................................................................................................18

9.1 LIBRARIESUSED...................................................................................................................18

9.2LAUNCH APP.PYINTERFACE..............................................................................................19

CHAPTER 10: CONCLUSION .................................................................................................................19

REFERENCES.........................................................................................................................................20

# ABSTRACT

Due to environmental changes and poor quality of construction materials, cracks may develop in the walls of the building which are critical for maintenance as the continuous exposure will cause severe damage to the environment. One of the initial signs of the degradation of a concrete surface or a material is cracks. Manual inspection has many drawbacks like invisibility of the crack, time consuming and it completely depends upon specialist’s knowledge and experience. So automatic image- based crack detection is used as replacement for manual inspection which reduces the cost when compared to the manual approach. The proposed algorithm has been tested against various concrete crack images to study how effectively the algorithm is working, and how effectively it is overcoming the drawbacks of the present manual approach. The algorithm is projected to serve the purpose of various people by estimating the parameters of the crack so the bricklayer can take the necessary actions from the degradation of concrete structures.

**LIST OF FIGURES:**

Fig (1) Cracks Classification

Fig(2): ‘General structure of image processing:based crack detection model.

Fig (3) : Steps for Proposed Algorithm

Fig (4): Sample images with cracks

Fig (5): Sample images without cracks

Fig(6): Training dataset

Fig (7): Val dataset

Fig(8): Testing dataset

Fig (9) : Bar Graph of the Target Classes before Dropping

Fig (10) : Accuracy & Val loss

Fig (11)**:** confusion matrix

Fig(12): Training and Validation Loss Over Time

Fig (13): Training and Validation Accuracy Over Time

Fig (14) : app.py Interface Positive Crack Detected

Fig (15) : app.py Interface Negative No Crack Detected

**CHAPTER 1: INTRODUCTION**

**INTRODUCTION:-** Image processing is any form of processing for which the input is an image, such as a photograph or video frame. The output of Image Processing can be either an image or a set of characteristics or parameters related to an image. The fundamental principle of Image processing operations carried out will assist us in its greater perception and vision but doesn’t add any information content. The recent availability of sophisticated semiconductor digital devices and compact powerful computers coupled with advances in Image processing algorithms has brought Digital Image processing to the forefront. Digital Image processing has a broad spectrum. It has varied applications such as remote sensing via satellites and other spacecraft image transmission and automates inspection of industrial paths storage for business applications, medical processing, radars and acoustic image processing robotics. Image processing is necessary because human beings are adept at interpreting images of a certain threshold beyond which we cannot detect just noticeable differences in the imagery. Human beings can detect only 8 to 16 shades of grey, even when data is recorded with 256 shades of grey. Therefore, one may not be able to interpret data in the remaining shades of grey. Also, it is necessary to continuously track large amounts of data and its storage is a problem. To avoid all these difficulties, one shall prefer processing of images by digital computers which processes at a much faster rate than human beings do.

Major requirement of image enhancement is to restore a captured image from degradations arising from imperfect acquisition conditions. For example, to remove the noise imposed, to correct the colour cast and to sharpen the objects that appear in the image. To restore an image to improve. Its contrast so that it is pleasant to a human viewer is one of the most demanding features. Image processing is a method to convert an image into digital form and accomplish some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is an image like a video frame or photograph and output may be an image or characteristics associated with that image. Usually an image processing system includes treating images as two-dimensional signals while applying already set signalling processing to them.

**1.1 Problem Definition**

The objective of the project is to identify the cracks on the concrete surfaces and to estimate the parameters of the crack. Using this calculated information builders can easily estimate the strength of any concrete structure and take immediate necessary action. Concrete is a quasi- brittle material with a low tensile strength. Applied loadings, deleterious chemical reactions and environmental effects can result in the development of tensile stress in concrete. When these tensile stresses exceed the concrete tensile strength, the concrete will crack. The extent and size of cracks influence the performance of the bridges and buildings. Although this cracking can be reduced by proper selection of concrete constituent materials, some cracking is inevitable.

**1.2 Motivation**

Image Processing at present is used in various applications. It is a fascinating and exciting area to be involved in today. Visual Information, transmitted in the form of digital images, is becoming a major method of communication in the modern age. Image Processing in future can be widely recognized and aware of cancer tumors and helps in prevention of diseases by making the person aware. Digital Image Processing deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focuses particularly on images. Digital Image Processing focuses on developing a computer system that is able to perform processing on an image. The input of the system is a digital image and the system processes that image using efficient algorithms, and gives us an output.

There are various types of tasks in Image Processing which includes Image Acquisition, Storage, Transmission; Image enhancement and Restoration; Image understanding and Recognition. All of these play an important role in the real world. Out of them, Image enhancement plays a vital role in various fields. Enhancements are used to make it easier for visual interpretation and understanding of imagery. The advantage of digital imagery is that it allows us to manipulate the digital pixel values in an image. An image ‘enhancement’ is basically anything that makes it easier or better to visually interpret an image. Also, an enhancement is performed for a specific application. This enhancement may be inappropriate for another purpose which would demand a different type of enhancement.

**1.3 Objectives**

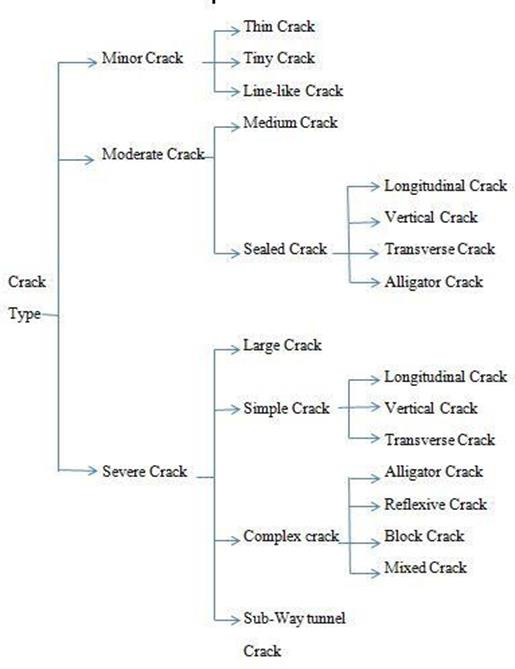
The main objective of developing this project are:

* To identify if the surface is cracked or not.
* It can be modified to see other parameters on the surface.
* It can be used to determine cracked and non-cracked concrete bridge decks, walls, and pavements.
* It can be used in manufacturing processes like casting, welding, forging,etc to find the surface crack or other parameter.

CHAPTER 2: RELATED WORKS

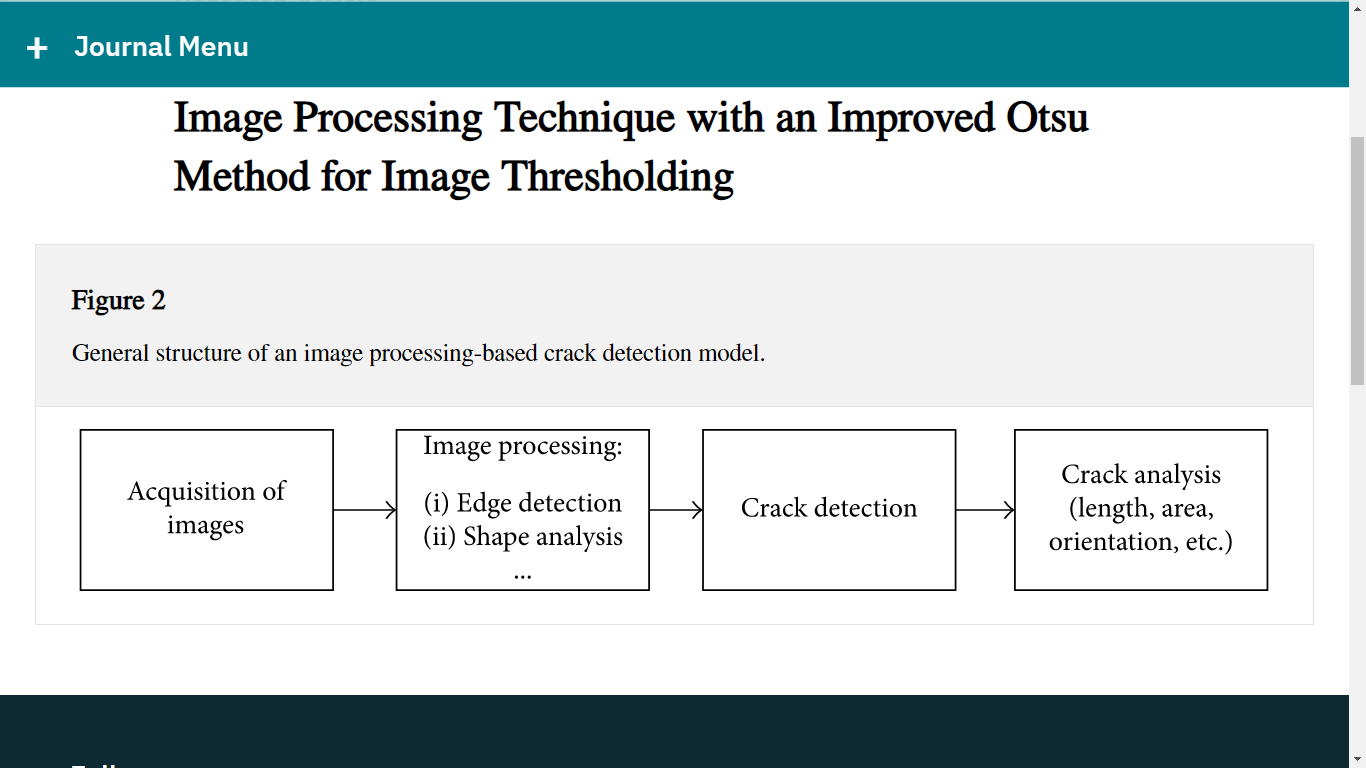
**2.1 CRACK CLASSIFICATION**

Cracks can be broadly classified into two categories namely active and dormant. In active cracks, the change in direction, width or depth occurs over a measured period whereas in dormant cracks it remains unchanged. If left unrectified, both active and dormant cracks provide passage for moisture penetration, which can lead to future damage. Some of the active cracks are longitudinal crack, transverse crack, miscellaneous crack, crocodile crack and reflection crack. Dormant cracks are very fine in nature and auto healing occurs over a time period. The various types of crack based on their structure are micro crack, thin crack, sealed crack, mixed crack, line-like crack, minor crack, tiny crack, medium crack, large crack and complex crack. Crack classification is an approach to find the specific crack type using machine learning algorithms. Crack detection identifies or recognizes the presence of crack whereas crack classification classifies the crack based on the feature extracted from the crack region. Machine learning is a subfield of Artificial Intelligence (AI), useful to perform classification, prediction and clustering of the dataset depending on the application. Classification/Prediction is carried out using supervised learning algorithms whereas clustering is carried out using unsupervised algorithms. The different types of supervised learning algorithms applied for crack classification are Support Vector Machine (SVM), K Nearest Neighbours algorithm (KNN). In underwater dams, it is difficult to detect and classify the cracks. Hence, solar images are used to detect and classify cracks into tiny, medium and large using tensor voting method. Salari and Ouyang stated that images not only contain a concrete segment, but also include other complicated background components. Crack in the pavement images with complicated background components such as trees, houses, etc. are also detected and classified using SVM, fractal thresholding and radon transform. Some of the images require pre-processing techniques for effective results. It includes a wiener filter to remove the blurriness and reduction method to reduce the noise. Chen et al. classifies the bridge crack into vertical, longitudinal, reflexive and crocodile cracks using SVM. Fuzzy clustering method, Nouha Ben et al. is useful when the input regions can’t be defined clearly and precisely. As a combination of the Fuzzy clustering method, k-means thresholding, segmentation, de-nosing, morphological operation and skeletonization gives an accuracy of 82% for pavement crack. A detailed review based on crack type has been done for crack detection and classification. The first level crack types (minor, moderate and severe) and its appropriate subtypes are shown in Figure 1.



*Fig (1) Cracks Classification*

Image-based systems have several benefits for monitoring the crack propagation in different structural materials. Initially, when these systems were used to measure cracks, more attention was paid to features of objects and repeatability. Cracking is an important indication of the degradation of structures. Detection of cracks is often required in the stage of building maintenance. In addition, inspections of the structural integrity based on crack analyses become substantial for the service life prediction of structures . Since the manual process for crack measurement is painstakingly time-consuming for large-scale structures (e.g., high-rise buildings and bridges), many researchers have proposed models based on image processing, which enable a faster and more efficient way of measuring the cracks in concrete surfaces. The general framework of these models is shown in Figure [2](https://www.hindawi.com/journals/ace/2018/3924120/fig2/)



**CHAPTER 3:PROPOSED ALGORITHM**

There are four phases in our proposed algorithm:

**1. Gaussian filtering**

**2. Canny edge detection**

**3. Morphological approach**

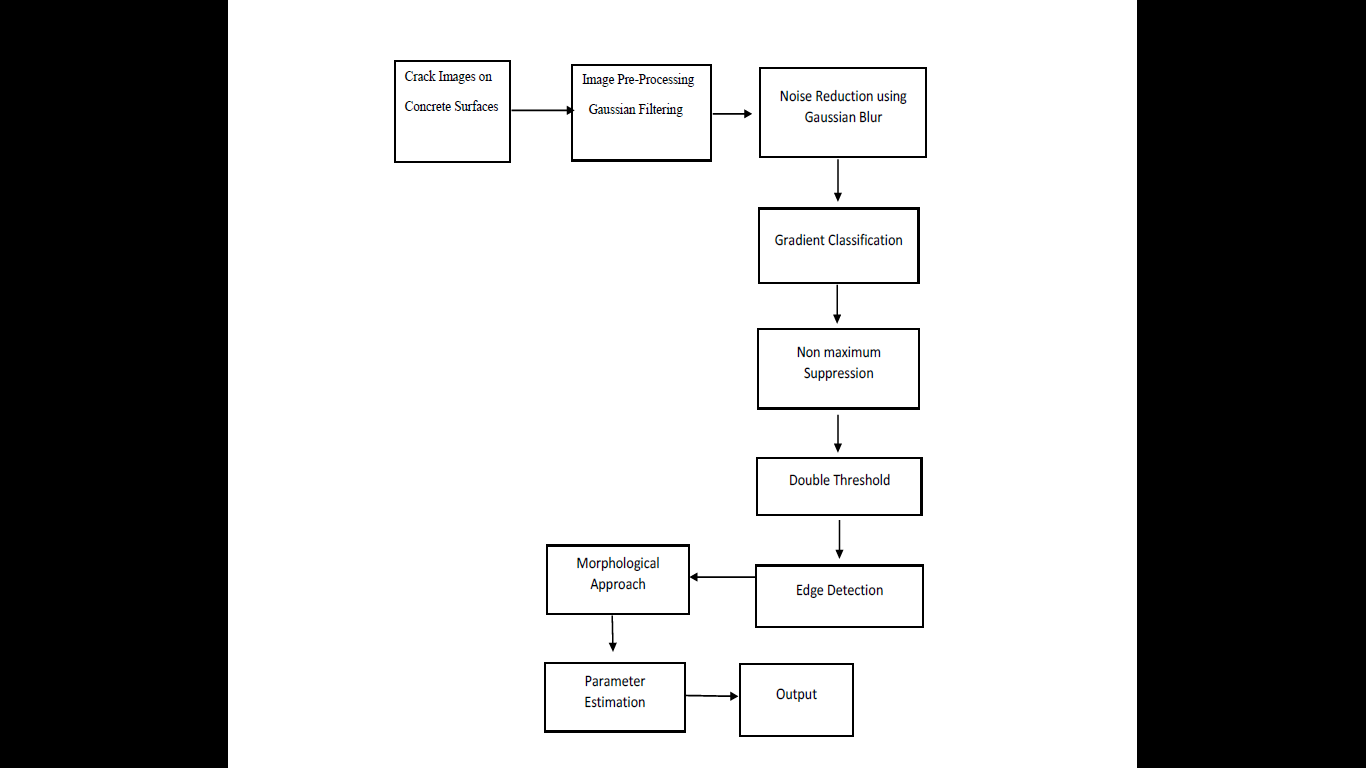
**4. Parameter estimation**

**3.1 Gaussian filtering:-** Gaussian filter is a linear filter which is used to blur the image or to reduce noise. If we apply Gaussian filtering and Median filtering to an image and subtract their outputs, final output can be used for unsharp masking (edge detection). The Gaussian filter alone will blur edges and reduce contrast. From the image perspective, during Gaussian filtering each individual pixel is modified with a Gaussian shaped blob with the same total weight as the original intensity value. This Gaussian is also referred to as the convolution kernel.

**3.2 Canny Edge Detection:-**Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the quantity of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations. Detection of edges with lower rate, which means that the detection should accurately catch as many edges. The Canny algorithm contains many adjustable parameters, which can affect the computation time and effectiveness of the algorithm. Primarily the smoothing filter used in the first stage of Gaussian filter directly affects the results of the canny edge detection algorithm. To reduce blurring effect and to detect small, sharp lines we can use small filters. But there are some disadvantages of using large filters. They increase the amount of blurring effect in the image and value of particular pixel slights out over a large area of image [9]. Larger blurring radii are more useful for detecting larger and smoother edges.

**3.3 Morphological approach:-**Morphological Image Processing is a collection of non-linear operations associated with the shape or morphology of features in an image. Morphological processing is capable of removing noise and has the ability to edit an image based on the size and shape of objects of interest. It is used in place of Linear Image Processing, because it sometimes distorts the geometric form of an image but in the case of Morphological approach the information of an image is not lost. In the Morphological Image Processing the original image can be reconstructed by using Dilation, Erosion, Opening and Closing operations for a finite number of times.

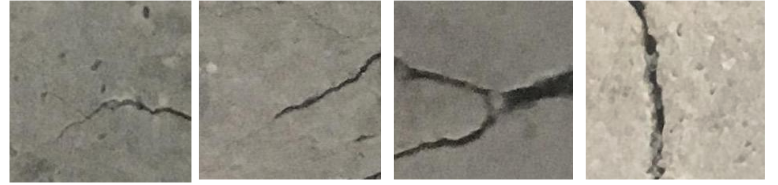
**3.4 Parameter estimation:-**Implementing the above proposed algorithm, we will calculate the height, depth, width, direction of propagation and severity of the crack.Steps involved in proposed algorithm are stated in the below Fig (3).

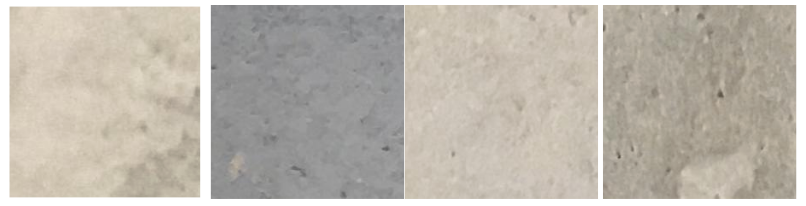


*Fig (3) : Steps for Proposed Algorithm*

**CHAPTER 4: DATASETS**

We are using the Publicly available [Concrete Crack Images data set](https://data.mendeley.com/datasets/5y9wdsg2zt/2). The data set consists of **20,000** images of concrete structures with cracks and **20,000** images without cracks. The dataset is generated from 458 high-resolution images (4032x3024 pixels). Each image in the data set is a 227 x 227 pixels RGB image. Some sample images with cracks and without cracks are shown below:

 *Fig 4: Sample images with cracks*



*Fig 5: Sample images without cracks*

As can be seen, the data set has a wide variety of images — slabs of different colours, cracks of different intensities and shapes.

# CHAPTER 5:Model Build

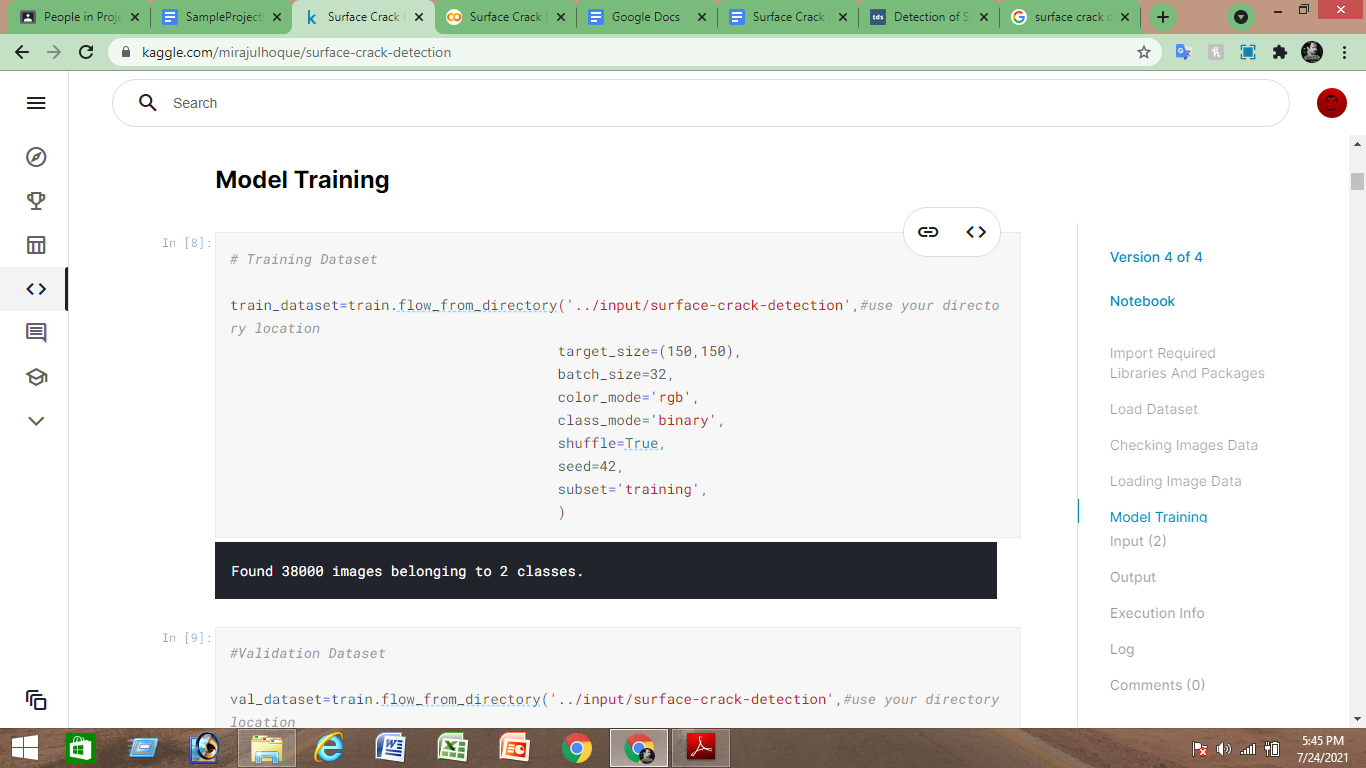
For this problem, let's build a Convolution Neural Network (CNN) in Pytorch. Since we have a limited number of images, we will use a pretrained network as a starting point and use image augmentations to further improve accuracy. Image augmentations allow us to do transformations like — vertical and horizontal flip, rotation and brightness changes significantly increasing the sample and helping the model generalize.

In this research, we chose three classic neural networks to conduct model training. First, we use the training dataset to train these three kinds of neural networks, input the training dataset into the neural network, and optimize according to the loss function.

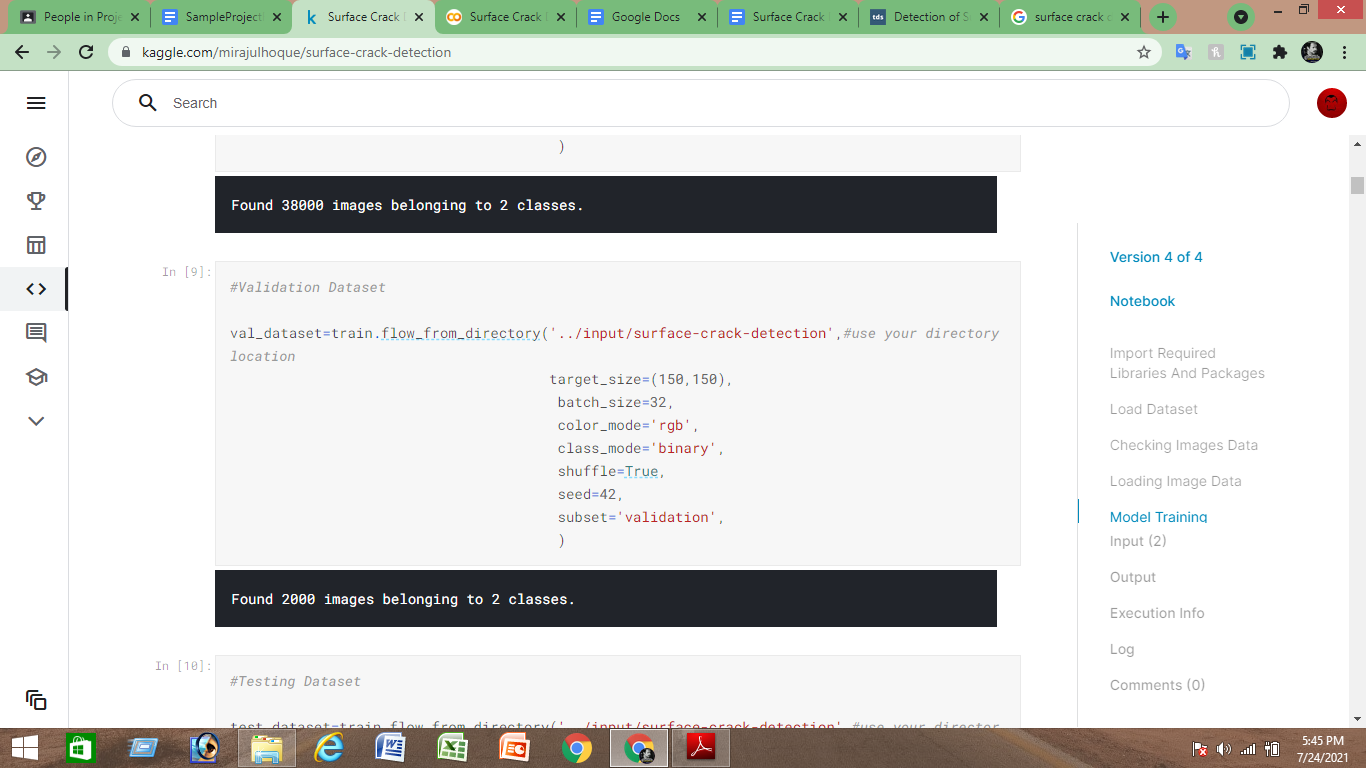
After a certain epoch, the neural network tends to converge and, then, uses the validation dataset to select the best model and, finally, compares the superiority of the training model according to the test set; the most suitable model was selected based on the comparison of the experimental results. Compared with the traditional method, the method in this paper has better generalization and robustness, but the method in this paper requires a large dataset, and the model training time is longer.

**Shuffle and Split input data into Train , Val & Test**

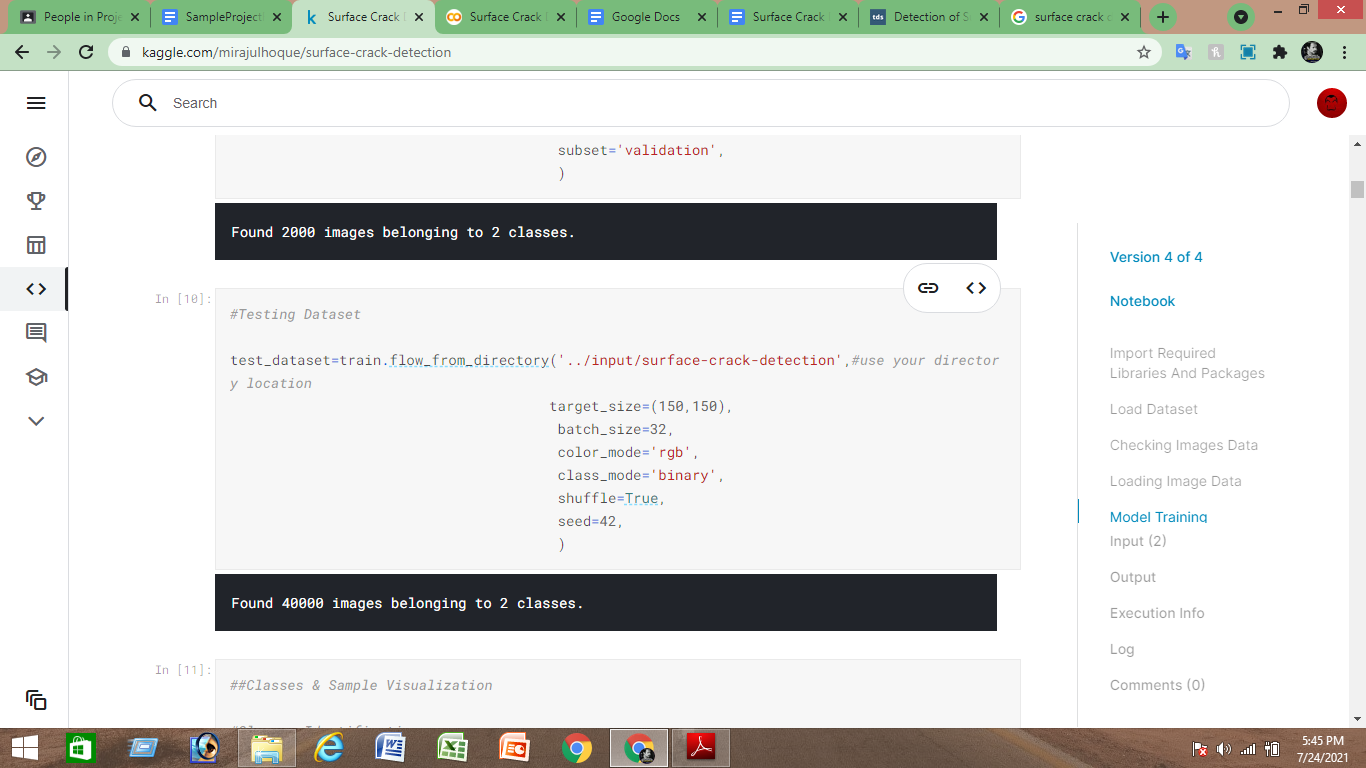
The data downloaded will have 2 folders one for **Positive** and one for Negative. We need to split this into train and val. The code snippet below will create new folders for train and val and randomly shuffle 85% of the data into train and rest into val.



*Fig(6): Training dataset*



*Fig(7): Val dataset*

****

*Fig(8): Testing dataset*

**CHAPTER 6: EXPERIMENTS**

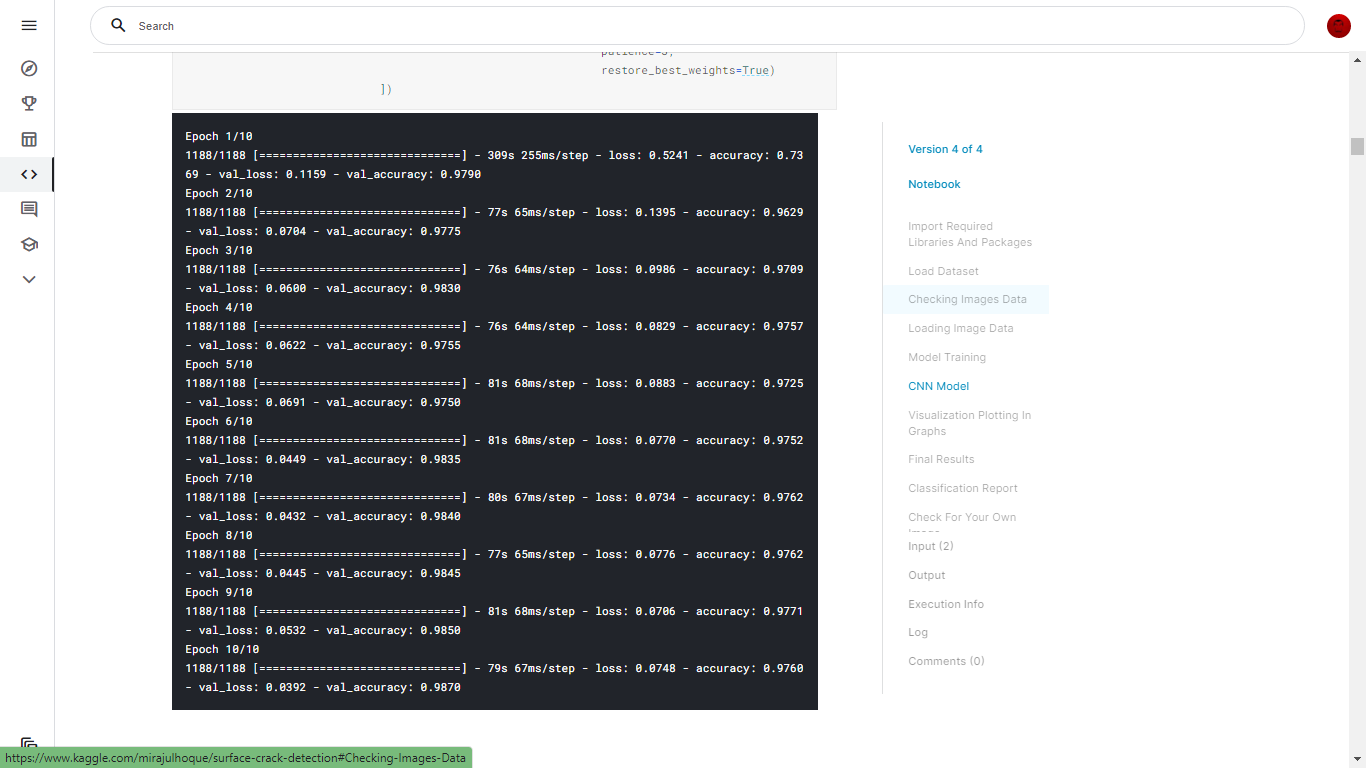
**6.1 Data Preparation:-** Since the dataset consists of 40000 observations with 20000 data Negative and 20000 data Positive, two different experiments were performed for data preparation.



*Fig 9 : Fig 9 : Bar Graph of the Target Classes before Dropping*

This leads to reduced number of the observations providing irrelevant training to our model. So, we progressed with imputation of data with the mean value of the observations and scaling them using SimpleImputer and StandardScaler modules of Sklearn.

**6.2 Accuracy & Val loss**



*Fig (10) : Accuracy & Val loss*

**6.3 Exploratory Analysis:**

Correlation Matrix visualization Before Feature Selection there is no single feature that has a very high correlation with our target value. Also, some of the features have a negative correlation with the target value and some have positive. The data was also visualized through plots and bar graphs.

**6.4 Training and testing**

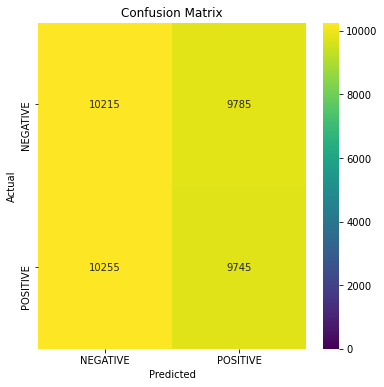
Finally, this resulting data split into 80% train,20% val & 100% test data, which was further passed to the LogisticRegression model to fit, predict and score the model.

**CHAPTER 7: EVALUATION METRICS**

For the evaluation of our output from our training the data, the accuracy was analyzed as a “**Confusion matrix**”.

**7.1 Confusion Matrix**

A confusion matrix, also known as an error matrix, is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. The key to the confusion matrix is the number of correct and incorrect predictions that are summarized with Predicted values Graph.

****

***Fig 11:*** *confusion matrix*

**7.2 Accuracy**

The accuracy is calculated as:

def evaluate\_model(model, test\_dataset):

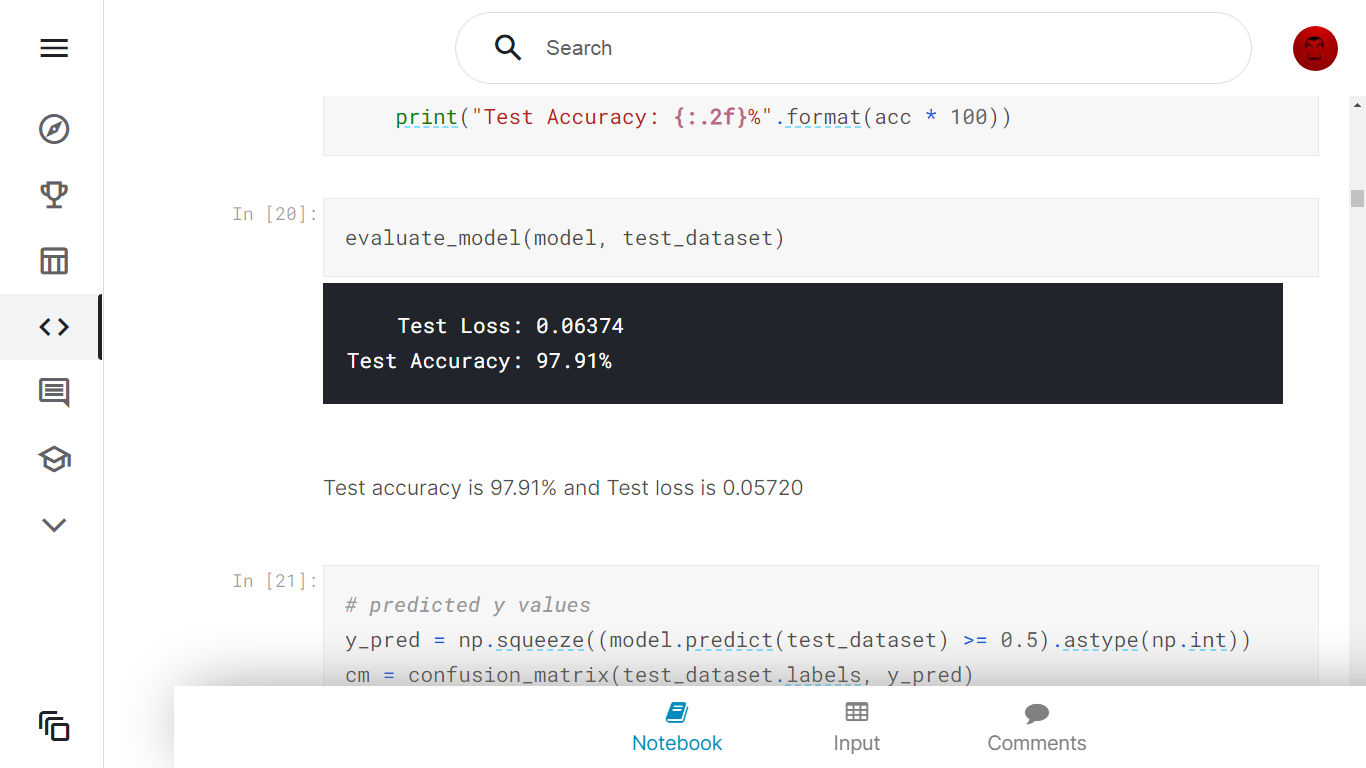
results = model.evaluate(test\_dataset, verbose=0)

loss = results[0]

acc = results[1]

print(" Test Loss: **{:.5f}**".format(loss))

print("Test Accuracy: **{:.2f}**%".format(acc \* 100))



The obtained accuracy during training the data after feature selection using backward elimination was testing was 97.91%.

**7.3 Recall & Precision**

Recall is calculated as: **Classification Report**

***#Classification Report***

**print("Classification Report:\n----------------------\n", clr)**

**Classification Report:**

**----------------------**

**precision recall f1-score support**

**NEGATIVE 0.50 0.51 0.50 20000**

**POSITIVE 0.50 0.49 0.49 20000**

**accuracy 0.50 40000**

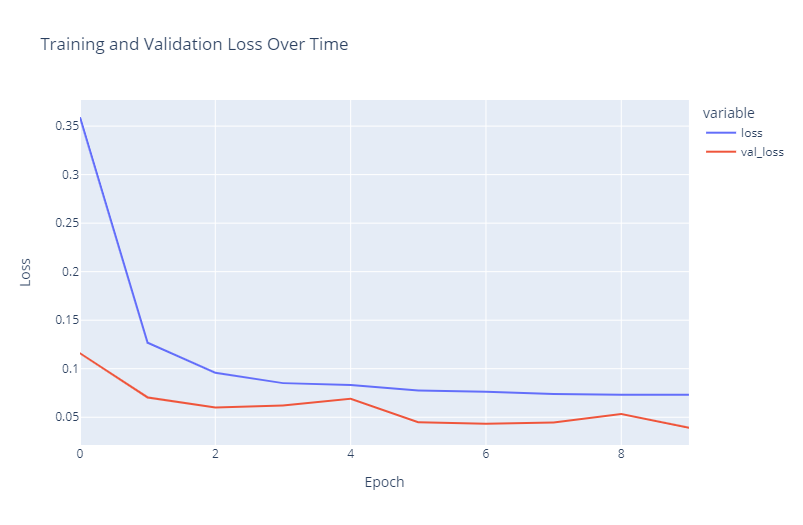
**macro avg 0.50 0.50 0.50 40000**

**weighted avg 0.50 0.50 0.50 40000**

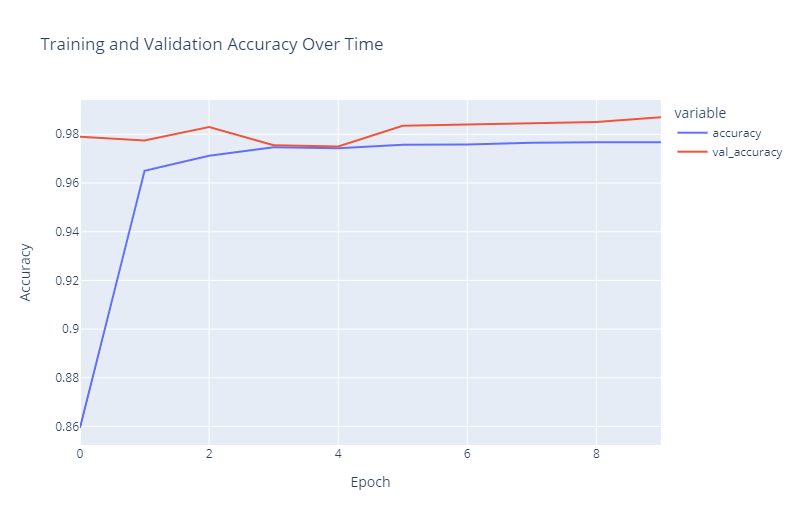
CHAPTER 8: DISCUSSION ON RESULTS

**8.1. Experimental Results and Analysis**

In this experiment, **10,000 crack and 10,000 non-crack** pictures were randomly shuffled, and **3000 pictures were selected as the validation set**, which was used to select the best model; 3000 pictures were selected as the test set, and the test set was used to test the performance of the model; **14,000 pictures were the training set**, which was used to train the model. To prevent overfitting, measures such as batch normalization, regularization, and dropout were used. Normalization normalizes the pixel value of the input image to (0,1), which is easy for neural network convergence. Regularization adds a regular term to the loss function, and the value of the loss function becomes the sum of the loss function and the regular term. Dropout indicates that a random part of the neuron parameters is not updated every time it is back propagated. The platform of this experiment is based on the Ubuntu platform, the programming language is Python, and the frameworks are TensorFlow, Keras, and OpenCV. The initial learning rate is 0.001, and epoch is 30. Figure **12** shows the accuracy curve of the training and validation dataset. There are 8 basic block modules in the red dotted area, and each module’s blue arrow is a short connection. This means that the neural network can be deeper because the gradient can transfer farther during backpropagation. The model is divided into 20 layers, including 17 convolutional layers and three fully connected layers. The pooling layer is not shown in the figure but exists after each convolutional layer. The specific parameters of the ResNet18 model are the same as those of Figure **13**.



*Fig(12): Training and Validation Loss Over Time*



*Fig (13): Training and Validation Accuracy Over Time*

**CHAPTER 9: CODE**

The coding portion was carried out to prepare the data, visualize it, pre-process it, build the model and then evaluate it. The code has been written in Python programming language using **Kaggle** **Notebook** as IDE. The experiments and all the model building are done based on python libraries. The code is available in the Git repository given in following link:

<https://github.com/miraJhq/Project-II-Machine-Learning-in-Python->

**9.1 Libraries used:**

1. NumPy

2. tensorflow

3. Matplotlib (pyplot, rcparams, matshow)

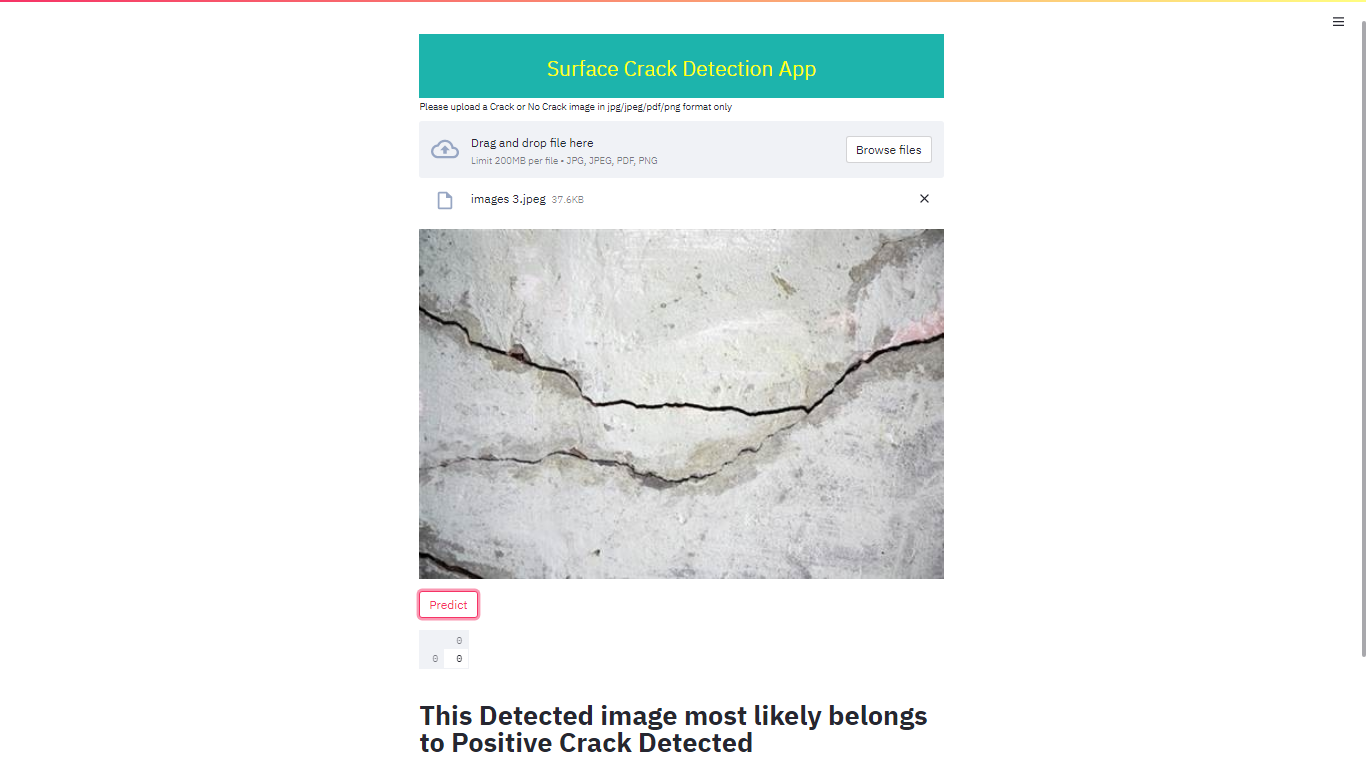
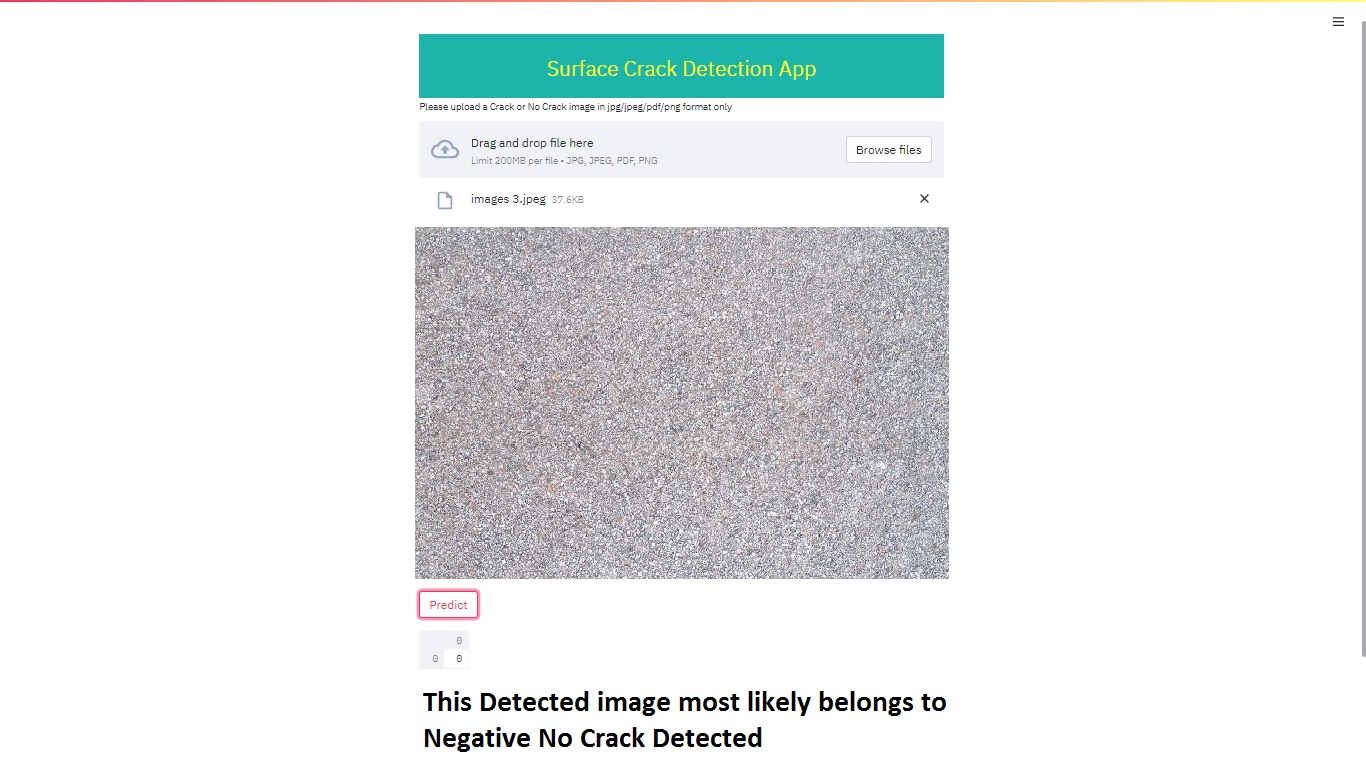
4. sklearn

5. Pandas

6. Keras

**9.2 Launch app.py Interface**

One of the most important skills you need to build as a Python is to be able to run Python scripts and code. This is going to be the only way for you to know if your code works as you planned. It’s even the only way of knowing if your code works at all.



*Fig 14 : app.py Interface Positive Crack Detected**Fig 15 : app.py Interface Negative No Crack Detected*

**CHAPTER 10: CONCLUSION**

The manual approach for the estimation of cracks in concrete surfaces is tedious. So, the proposed automatic crack detection algorithm identifies the cracks on concrete surfaces with better results compared to manual approach. In the proposed crack detection algorithm the cracks are identified using the canny edge detector algorithm which provides a better results for any kind of an image The proposed algorithm is tested with nearly 100 images and the different parameters related to crack such as length, width are measured, this can be used by various construction workers so that necessary remedies can be taken from the degradation of concrete material. Direction of Propagation and Life Span of the concrete material are measured further with different high-quality techniques.

**REFERENCES**

[1] A. H. M. S. U. Marjia Sultana, "Analysis of Data Mining Techniques for Heart Disease Prediction," 2018.

[2] M. I. K. ,. A. I. ,. S. Musfiq Ali, "Heart Disease Prediction Using Machine Learning Algorithms".

[3] K. Bhanot, "towarddatascience.com," 13 Feb 2019. [Online]. Available: https://towardsdatascience.com/predicting-presence-of-heart-diseases-using-machine learning-36f00f3edb2c. [Accessed 2 March 2020].

[4] [Online]. Available: https://www.kaggle.com/ronitf/heart-disease-uci#heart.csv.. [Accessed 05 December 2019].

[5] M. A. K. S. H. K. M. a. V. P. M Marimuthu, "A Review on Heart Disease Prediction using Machine Learning and Data Analytics Approach".