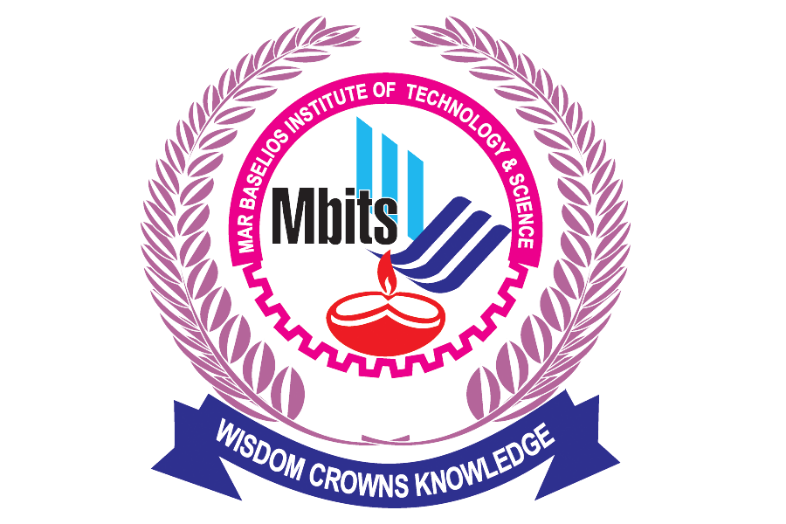
### MAR BASELIOS INSTITUTE OF TECHNOLOGY AND SCIENCE

### Nellimattom, Kothamangalam

### (Affiliated to APJ Abdul Kalam Technological University, TVM)

****

**Department of Computer Applications**

Mini Project Report

**HEART ATTACK PREDICTION**

Done by

**ANN MARIYA JOY**

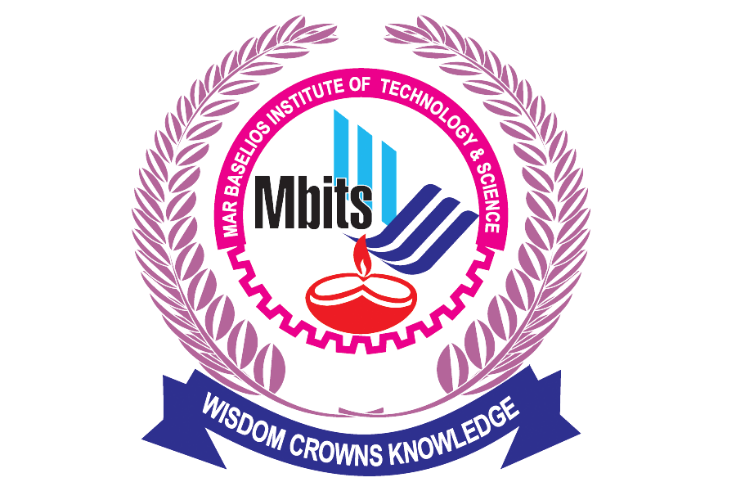
**MBI23MCA-2013**

Under the guidance of

**Prof. REENU SEBU**

**2023-2025**

**CERTIFICATE**



**HEART ATTACK PREDICTION**

Certified that this is the bonafide record of project work done by

### ANN MARIYA JOY

### MBI23MCA-2012

During the academic year 2023-2025, in partial fulfillment of requirements for award of the degree,

**Master of Computer Applications of**

**APJ Abdul Kalam Technological University Thiruvananthapuram**

### Faculty Guide Head of the Department

Prof. Reenu Sebu Prof. Reshma S

### Project Coordinator Internal Examiner

Prof. Merin Joy M

# ACKNOWLEDGEMENT

First and foremost, I thank God Almighty for his divine grace and blessings in making all this possible. May he continue to lead me in the years to come. No words can express my humble gratitude to my beloved parents who have been guiding me in all walks of my journey.

I am also grateful to Prof.Reshma S, Head of Computer Applications Department, Ms.Merin Joy M, Project Coordinator for their valuable guidance and constant supervision as well as for providing necessary information regarding the project & also for their support.

I would like to express my special gratitude and thanks to my Project Guide Ms. Reenu Sebu, Assistant Professor, Department of Computer Applications for giving me such attention and time.

I profusely thank other Professors in the department and all other staffs of MBITS, for their guidance and inspirations throughout my course of study. My thanks and appreciations also goes my friends and people who have willingly helped me out with their abilities.

# ABSTRACT

In the medical field, the diagnosis of heart attack is the most difficult task. The diagnosis of heart attack is difficult as a decision relied on grouping of large clinical and pathological data. Due to this complication, the interest increased in a significant amount between the researchers and clinical professionals about the efficient and accurate heart attack prediction. In case of heart attack, the correct diagnosis in early stage is important as time is the very important factor. Heart attack is the principal source of deaths widespread, and the prediction of heart attack is significant at an untimely phase. Machine learning in recent years has been the evolving, reliable and supporting tools in medical domain and has provided the greatest support for predicting attack with correct case of training and testing. The main idea behind this work is to study diverse prediction models for the heart attack and selecting important heart attack feature using Random Forest algorithm. Random Forest is the Supervised Machine Learning algorithm which has the high accuracy compared to other Supervised Machine Learning algorithms such as logistic regression etc. By using Random Forest algorithm, we are going to predict if a person has heart attack or not.

**REFERENCES:**

* Lui, C. K., Kerr, W. C., Li, L., Mulia, N., Ye, Y., Williams, E., ... & Lown, E. A. (2020). Lifecourse Drinking Patterns, Hypertension, and Heart Problems Among US Adults. American Journal of Preventive Medicine.

# INTRODUCTION

Heart disease is a significant health concern globally, with heart attacks posing serious risks that can lead to life-threatening complications if not detected early. To facilitate faster and more accurate diagnosis, a Heart Attack Prediction Model using the RandomForest algorithm has been developed. This model analyses patient data, including risk factors such as age, smoking, blood pressure, cholesterol levels, and diabetes, to classify individuals into low-risk or high-risk categories for heart attacks. By training the model on a dataset of historical health records, it automatically learns to identify critical patterns and distinguishing features associated with heart disease. The Random Forest algorithm, known for its robustness and ability to handle complex datasets, offers a reliable, automated solution for heart attack prediction. This model supports healthcare professionals in diagnosing and managing patient risks more efficiently, ultimately aiding in early intervention and improving patient outcomes in cardiovascular health.

# SUPPORTING LITERATURE

### Literature Review

* **Paper 1: Lui, C. K., Kerr, W. C., Li, L., Mulia, N., Ye, Y., Williams, E., ... & Lown, E. A. (2020). Lifecourse Drinking Patterns, Hypertension, and Heart Problems Among US Adults. American Journal of Preventive Medicine.**

In the study "Lifecourse Drinking Patterns, Hypertension, and Heart Problems Among US Adults," Lui et al. (2020) explore the impact of alcohol consumption patterns over the lifespan on hypertension and heart-related issues in U.S. adults. The researchers categorize drinking behaviors into different life stages and identify how these patterns correlate with increased blood pressure and cardiovascular risk. They find that both excessive and inconsistent drinking are associated with higher rates of hypertension, while long-term heavy drinking significantly raises the likelihood of developing heart problems. Additionally, the study highlights the influence of demographic factors, such as age and socioeconomic status, on drinking patterns and health outcomes. The authors emphasize the need for public health initiatives to address alcohol consumption as a modifiable risk factor for cardiovascular diseases, advocating for increased awareness and targeted interventions to improve heart health.

### Findings and Proposals

The research on heart attack prediction utilizing a Random Forest algorithm reveals several key findings that underscore its effectiveness in assessing cardiovascular risk. The model demonstrates robust classification performance by accurately predicting heart attack likelihood based on critical health features such as age, cholesterol levels, blood pressure, and lifestyle factors. It effectively manages imbalanced datasets, which are common in medical predictions, and provides valuable insights into feature importance, highlighting the significant roles of various risk factors. To enhance the prediction of heart attack risk, the study proposes developing a comprehensive Random Forest model that integrates additional clinical data, including patient history and lifestyle choices. Hyperparameter optimization is recommended to refine the model’s performance, and validation through clinical trials will ensure its effectiveness in real-world settings. Furthermore, the authors suggest exploring advanced ensemble techniques and developing user-friendly decision support systems based on the Random Forest model to aid healthcare providers in making informed patient management decisions.

# SYSTEM ANALYSIS

### Analysis of Dataset

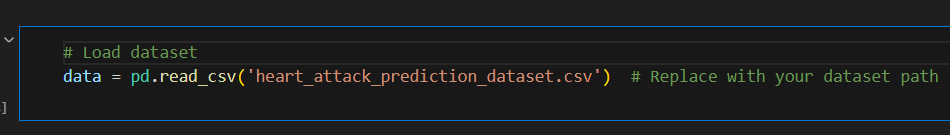
* + - 1. **About the Dataset**

The dataset consists of 8,763 entries with 26 attributes that provide valuable insights into heart health and associated risk factors. Key features include age, sex, and cholesterol levels, as well as essential health indicators such as blood pressure and heart rate. Lifestyle factors like diabetes, family history of heart disease, smoking, and obesity are also included, alongside details on alcohol consumption, exercise hours per week, and sedentary hoursperday. Additional attributes encompass dietary preferences, socio-economic indicators like income, and measurements such as BMI. The dataset's target variable, heart attack risk, serves to assess the likelihood of a heart attack, making it a rich resource for predictive modeling and analysis. This comprehensive collection of health metrics and lifestyle data is instrumental in understanding and evaluating heart disease risk, facilitating the development of machine learning models aimed at improving health outcomes.

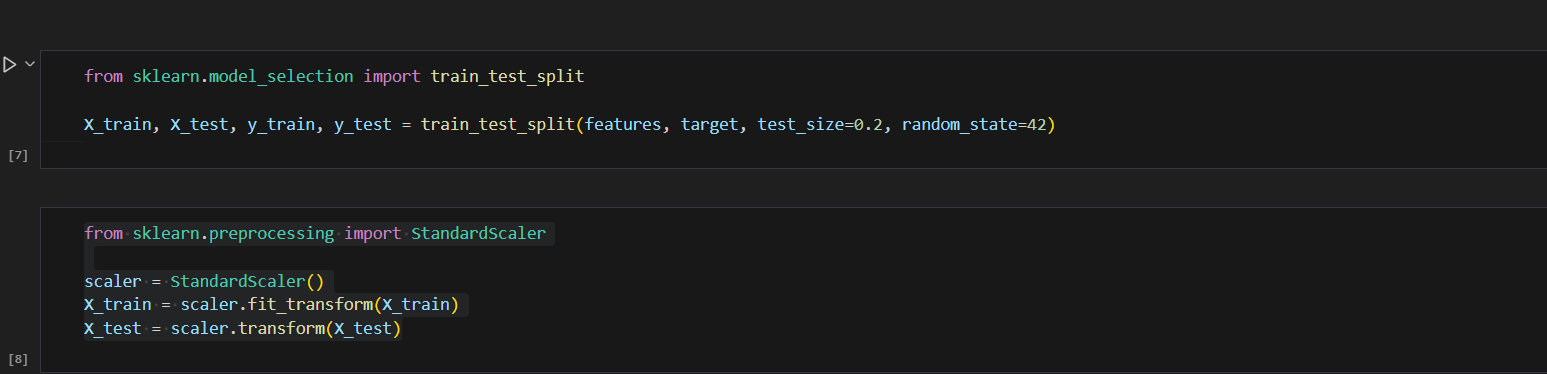
### Explore the Dataset

The dataset comprises 8,763 entries and 26 columns, detailing patient data relevant to heart attack risk. It includes demographic information such as Patient ID, Age, and Sex, alongside medical indicators like Cholesterol, Blood Pressure, and Heart Rate. Lifestyle factors are recorded, including Diabetes, Family History, Smoking, Obesity, and Alcohol Consumption. It also features metrics for physical activity and dietary habits, along with socioeconomic factors like Income and BMI. The target variable, Heart Attack Risk, indicates whether a patient has experienced a heart attack (1) or not (0).

### Loading the dataset.



### Splitting Training and Testing Data

****

* + 1. **Data Pre-Processing**

Data preprocessing for heart attack prediction involves several key steps to prepare the dataset for analysis. First, the dataset is loaded and explored to understand its structure and distributions. Any missing values are checked for and addressed through removal or imputation. Categorical variables, such as sex and diet, are converted into numerical formats using techniques like one-hot encoding and label encoding. Feature scaling is then applied to ensure all features are on a similar scale, using standardization or min-max scaling. The dataset is subsequently divided into features and the target variable, with a split into training and testing sets to evaluate model performance. Finally, the processed data is reviewed for consistency, ensuring that it is clean and ready for predictive modeling. This systematic approach to data preprocessing is crucial for building an effective heart attack prediction model.

### Data Cleaning

Data cleaning is a critical step in preparing the heart attack prediction dataset for analysis and modeling. It begins with identifying and handling missing values, as they can skew the results and lead to inaccurate predictions. Techniques such as imputation—replacing missing values with the mean, median, or mode—are commonly used. Additionally, the removal of duplicate entries is essential to ensure that each observation is unique, thereby preventing bias in the analysis. Furthermore, outlier detection is crucial, as extreme values can disproportionately influence model performance; these outliers may be addressed through transformation or removal, depending on their nature and impact. Data type verification is also important, ensuring that numerical features are correctly formatted for analysis and that categorical variables are encoded appropriately. Finally, standardization or normalization of numerical features may be applied to bring all data to a common scale, enhancing the model’s learning process. Through these comprehensive cleaning procedures, the dataset becomes more reliable, paving the way for accurate predictive modeling.

* + - 1. **Analysis of Feature Variable**

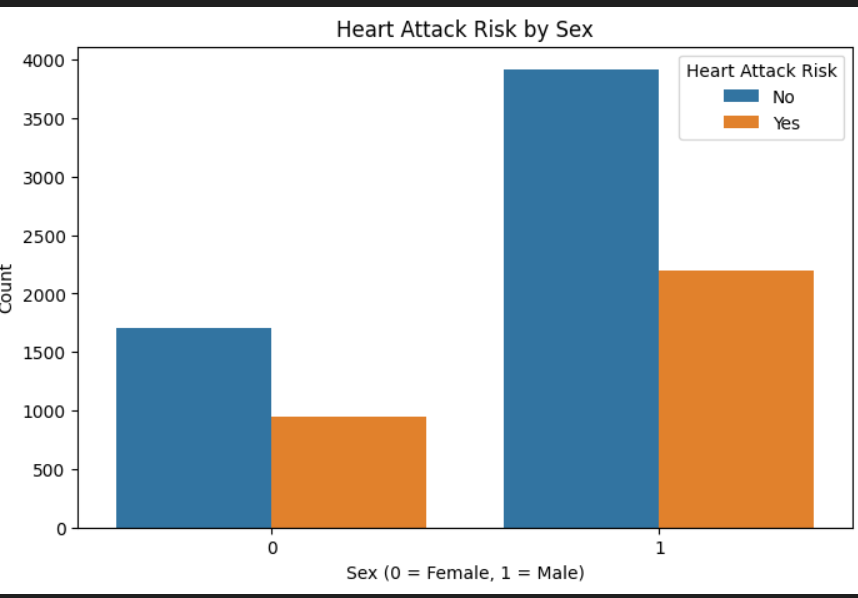
The feature variables are essential for assessing an individual's risk of cardiovascular events. Age is a significant risk factor, with older individuals facing a higher likelihood of heart issues. Sex also influences risk, as men generally have a higher risk than women at younger ages, though this shifts post-menopause. Cholesterollevels, particularly elevated low-density lipoprotein (LDL), contribute to arterial plaque buildup, while bloodpressure is vital, as hypertension can strain the heart and damage blood vessels. Additional factors include heartrate, the presence of diabetes, and familyhistory of heart disease, all of which play important roles in risk assessment. Lifestyle choices such as smoking, obesity, and physicalactivity, along with diet, alcoholconsumption, and stresslevels, further impact cardiovascular health. Analysing these diverse features enables healthcare professionals to evaluate heart attack risk accurately and develop targeted preventive strategies for personalized care.

### Analysis of Class Variables

### The class variable in a heart attack prediction dataset is typically the target variable that indicates whether or not a person is at risk of a heart attack. In this case, the "Heart Attack Risk" column represents the class variable, which likely contains binary values (0 or 1), where:0 indicates no heart attack risk,1 indicates the presence of heart attack risk

### 

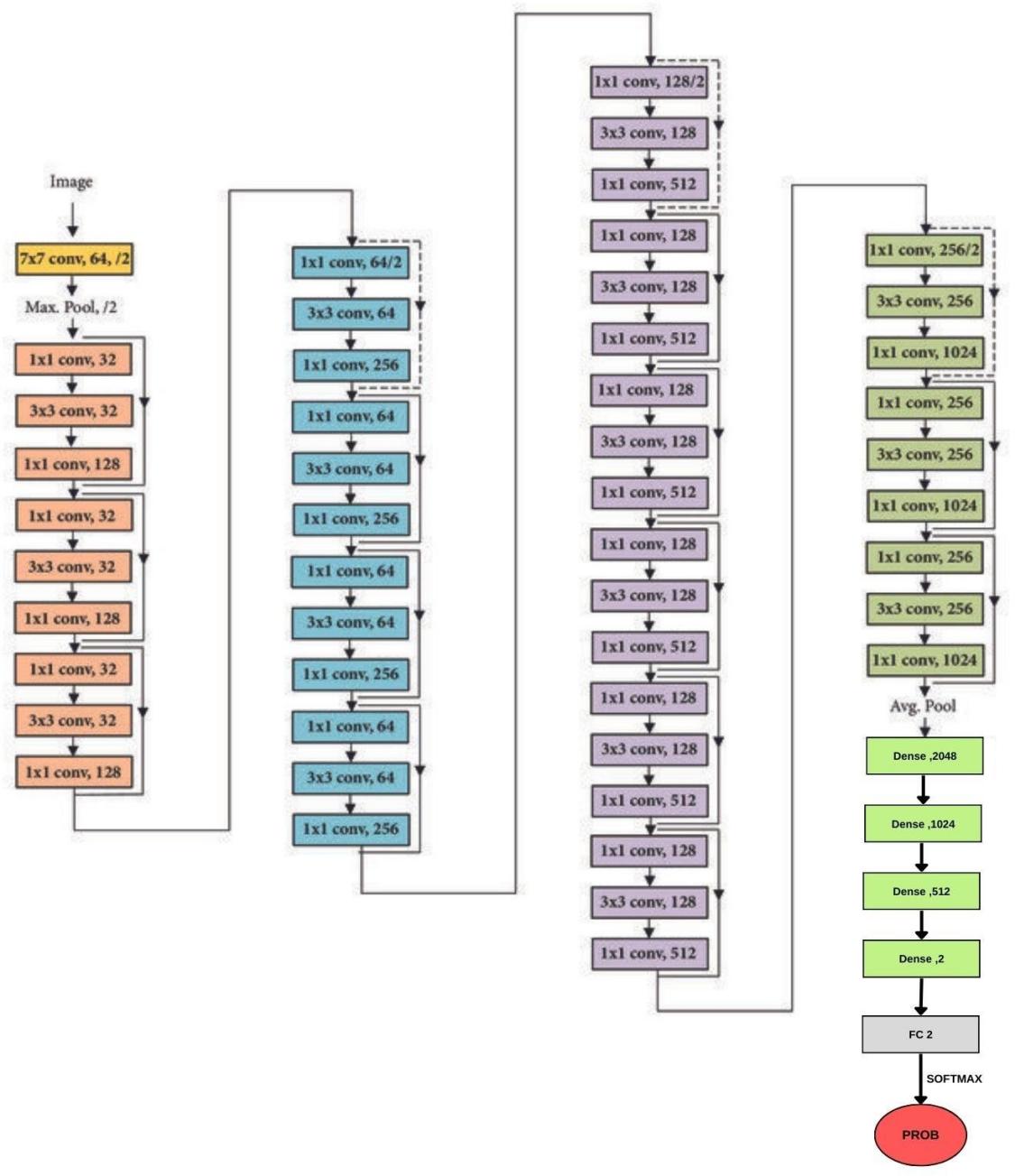
### Data Visualization

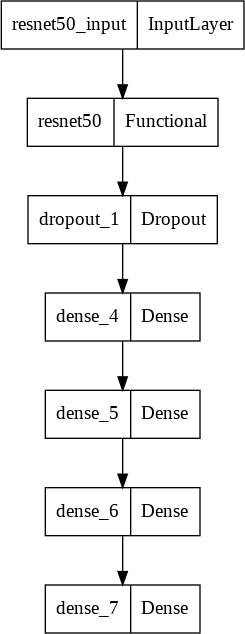


### Analysis of Architecture

The Random Forest algorithm is a robust machine learning technique ideal for heart attack risk prediction. It creates an ensemble of decision trees, each trained on random data subsets, and combines their outputs for the final prediction. This method handles complex datasets with multiple features like Age, Sex, Cholesterol, and Blood Pressure, effectively managing missing values and reducing overfitting. Random Forests also offer insights into feature importance, enhancing the understanding of risk factors. Its reliability, accuracy, and efficiency make it a strong choice for predicting heart attack risks.

### Block Diagram



The system is based on the ResNet 50 architecture, a variant of [ResNet model](https://iq.opengenus.org/resnet/) which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations. It is a widely used ResNet model and we have explored **ResNet50 architecture** in depth.

Because of the framework that ResNets presented it was made possible to train ultra- deep neural networks and by that i mean that i network can contain hundreds or thousands of layers and still achieve great performance. The ResNets were initially applied to the image recognition task but as it is mentioned in the paper that the framework can also be used for non-computer vision tasks also to achieve better accuracy. Many of you may argue that simply stacking more layers also gives us

better accuracy why was there a need of Residual learning for training ultra-deep neural networks.

* A convoultion with a kernel size of 7 \* 7 and 64 different kernels all with a stride of size 2 giving us **1 layer**.
* Next we see max pooling with also a stride size of 2.
* In the next convolution there is a 1 \* 1,64 kernel following this a 3 \* 3,64 kernel and at last a 1 \* 1,256 kernel, These three layers are repeated in total 3 time so giving us **9 layers** in this step.
* Next we see kernel of 1 \* 1,128 after that a kernel of 3 \* 3,128 and at last a kernel of 1 \* 1,512 this step was repeated 4 time so giving us **12 layers** in this step.
* After that there is a kernal of 1 \* 1,256 and two more kernels with 3 \* 3,256 and 1 \* 1,1024 and this is repeated 6 time giving us a total of **18 layers**.
* And then again a 1 \* 1,512 kernel with two more of 3 \* 3,512 and 1 \* 1,2048 and this was repeated 3 times giving us a total of **9 layers**.
* After that we do a average pool and end it with a fully connected layer containing 1000 nodes and at the end a softmax function so this gives us **1 layer**.
* We don't actually count the activation functions and the max/ average pooling layers.

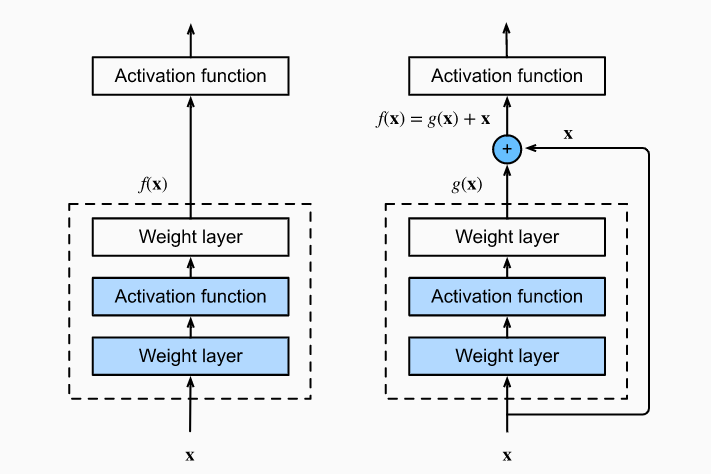
So totalling this it gives us a 1 + 9 + 12 + 18 + 9 + 1 = 50 layers Deep Convolutional network.

Here in our system, Avg Pool layer if followed by a drop out layer, then 4 dense

block layer, which optimises the number of values in the fully connected layer to 2, which is the count of the class variable of our system. So the final output will be an array of length 2.

### Diagrams and Details of Each Layer

**Convolutional Layer:** The key building block in a convolutional neural network is the convolutionallayer. This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to thesize of the filter (MxM). The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image. Size of the feature map = [(input\_ size — kernel size + 2 x padding) / stride] +1.

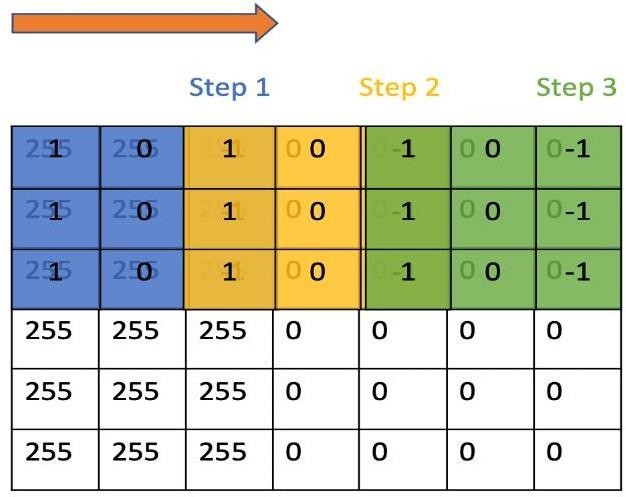


**Kernel/Filter:** In Convolutional neural network, the kernel is nothing but a filter that is used to extract the features from the images. The kernel is a matrix that moves over the input data, performs the dot product with the sub-region of input data, and gets the output as the matrix of dot products. Kernel moves on the input data by the stride value. If the stride value is 2, then kernel moves by 2 columns of pixels in the input matrix. In short, the kernel is used to extract high-level features like edges from the image.

### Stride:

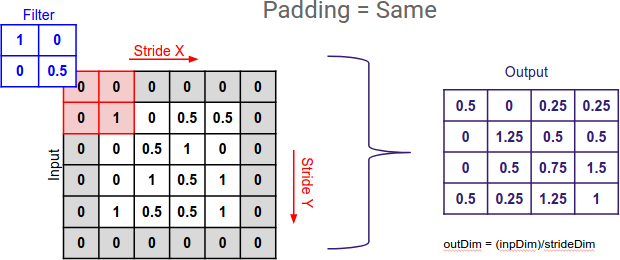
Stride is a component of convolutional neural networks, or neural

networks tuned for the compression of images and video data. Stride is a parameter of the neural network‘s filter that modifies the amount of movement over the image or video. For example, if a neuralnetwork's stride is set to 1, the filter will move one pixel, or unit, at a time. The size of the filter affects the encoded output volume, so stride is often set to a whole integer, rather than afraction or decimal



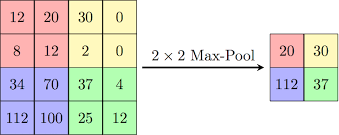
### Padding:

Padding is a term relevant to convolutional neural network as it refers to the numberof pixels added to an image when it is being processed by the kernel of a CNN. For example, if the padding in a CNN is set to zero, then every pixel value that is added will be of value zero. If, however, the zero padding is set to one, there will be a one-pixel border added to theimage with a pixel value of zero. A Finer Padding value is ‗same‘.



### Pooling Layer:

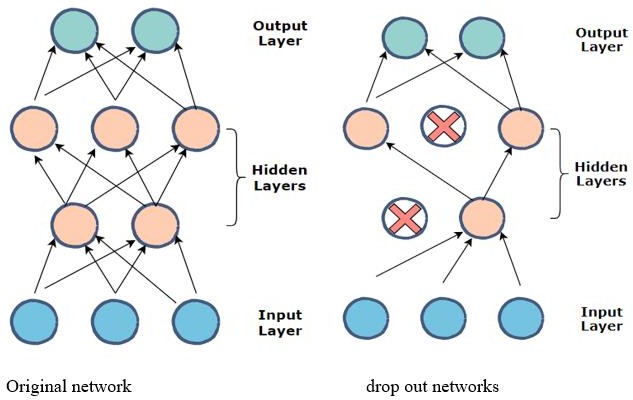
The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of Pooling operations. In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section are computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the Fully Connected Layer. The pooling layer is also called subsampling layer. Max pooling provides much better performance than average pooling. In max pooling layer, the maximum value among all the values in a matrix is chosen. Here we are using MaxPooling.



### Dropout Layer:

Another typical characteristic of CNNs is a Dropout layer. Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the

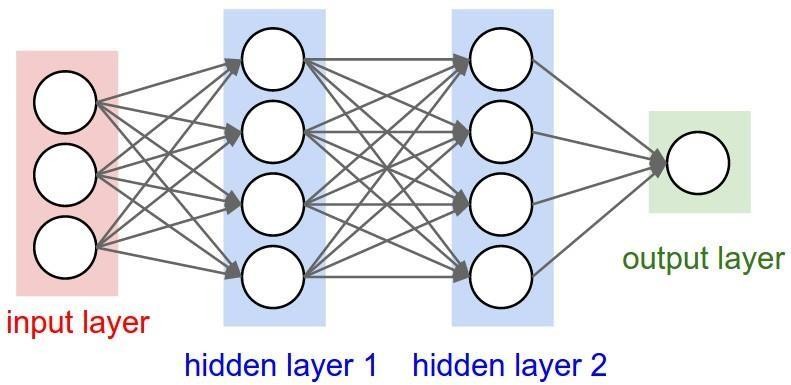
training data causing a negative impact in the model‘s performance when used on a new data. To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropoutof 0.5, 50% of the nodes are dropped out randomly from the neural network.



### Fully connected Layer:

They typically were included as the last few layers of most CNNs, appearing after several convolution and subsampling operations were performed. Fully connected layers were independent neural networks that possessed one or more hidden layers. Their operations involved multiplying their inputs by trainable weight vectors, with a trainable bias sometimes summed to those results. The output of these layers was traditionally sent through activation functions, similarly to convolution layers.

These layers take the output of the previous layers, ―flattens‖ them and turns them intoa single vector that can be an input for the next stage.



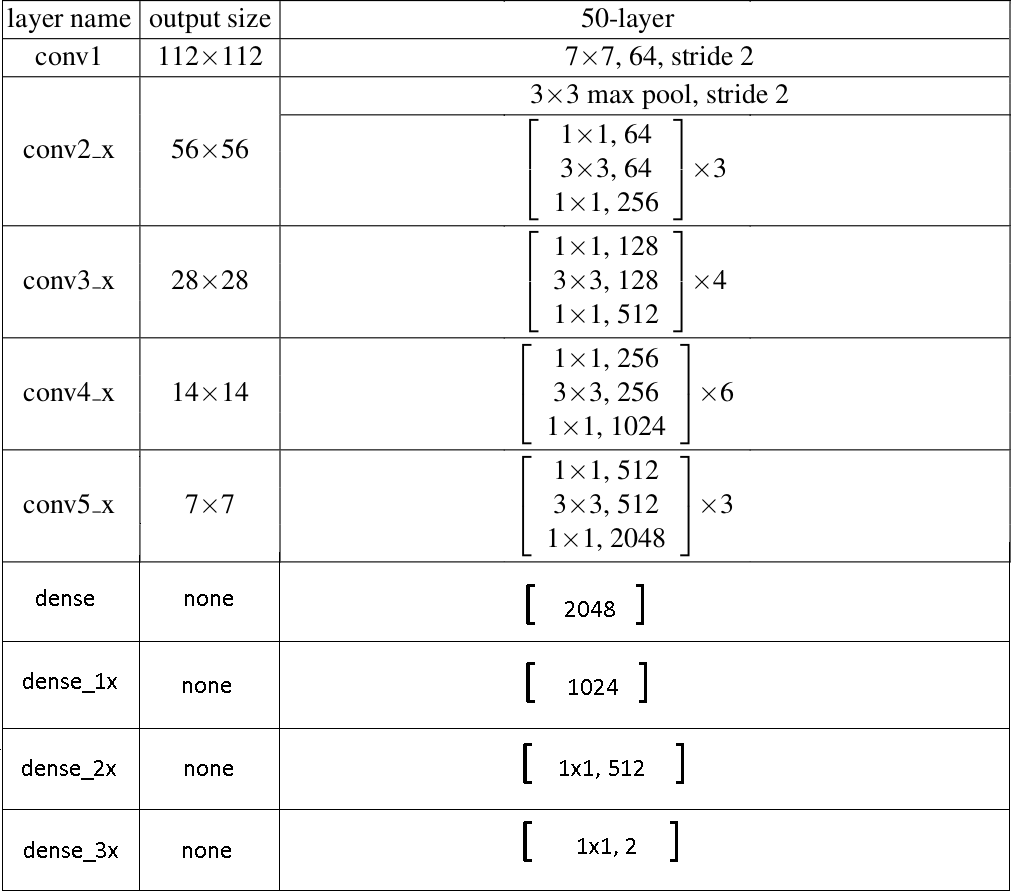
**Activation Functions:** Finally, one of the most important parameters of the CNN model is theactivation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network. It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, SoftMax, tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and SoftMax functions are preferred and for multi-class classification, generally SoftMax us used. In our work we used two activation functions ReLu, and Softmax.

**ReLU** (rectified linear activation function) is a linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance. The usage of ReLU helps to prevent the exponential growth in the computation required to operate the neural network.

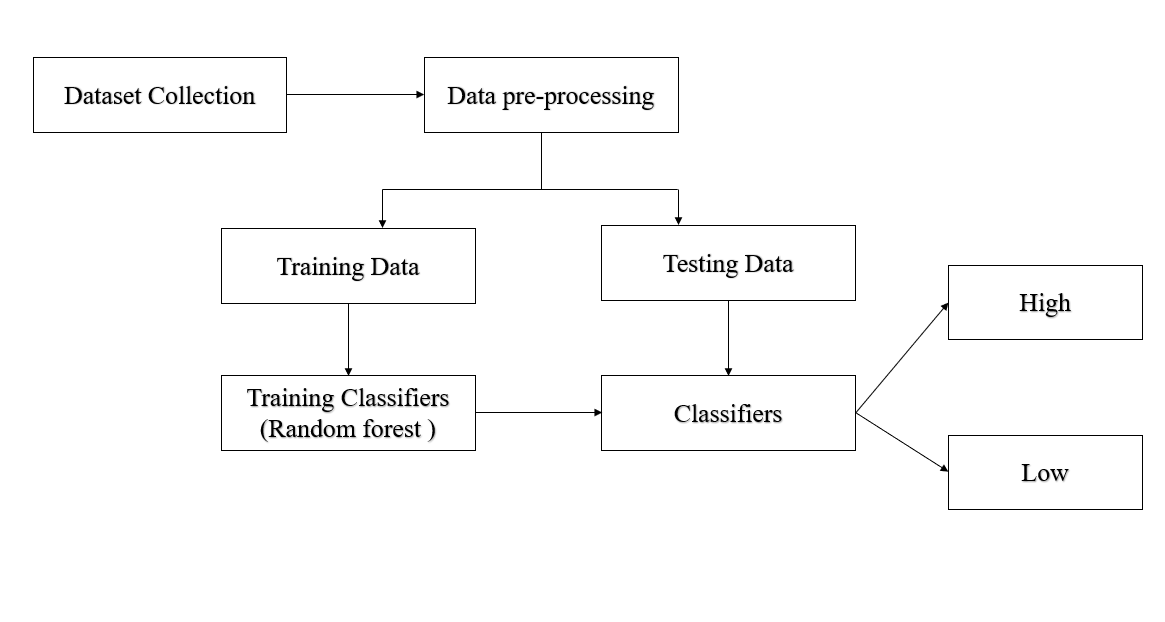
**Softmax** is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector.

The most common use of the softmax function in applied machine learning is in its use as an activation function in a neural network model. Specifically, the network is configured to output N values, one for each class in the classification task, and the softmax function is used to normalize the outputs, converting them from weighted sum values into probabilities that sum to one. Each value in the output of the softmax function is interpreted as the probability of membership for each class.

### Dimension Table



* + 1. **Project Pipeline**

****

### Feasibility Analysis

A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing system or proposed system, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success.

Evaluated the feasibility of the system in terms of the following categories:

* Technical Feasibility
* Economical Feasibility
* Operational Feasibility

### Technical Feasibility

The application "Heart Attack Risk Prediction" is technically feasible because all the technical resources required for the development and functioning of the application are easily available and reliable. The hardware requirement for this application is a computer system with internet access. The code is written in Visual Studio Code (VS Code), which allows for creating a separate environment for the project with all necessary libraries installed. The user interface is developed using HTML, which is simple to understand and implement. These requirements are readily available, reliable, and help make the system efficient, saving time and reducing the need for manpower. The system will be easy to develop, manage, and modify since the technologies used are common and widely accessible.

### Economic Feasibility

The cost to manage this system will be lesser. The system requires only a computer for working. The code is working on Visual Studio Code, so it consumes no amount of internet. The development of the system will not need a huge amount of money. It will be economically feasible. And the money spend for the application will be worth.

### Operational Feasibility

The developed system is completely driven and user friendly. Since the code is written on Visual Studio Code, the system has a separate environment without having to utilize resources in a common environment. There is no need of skill for a new user to open this application and use it. The interface contains only a file upload option

and a Detect button. Users also need to be aware of the application initially. Then they can use it easily. So, it is feasible.

### System Environment

System environment specifies the hardware and software configuration of the new system. Regardless of how the requirement phase proceeds, it ultimately ends with the software requirement specification. A good SRS contains all the system requirements to a level of detail sufficient to enable designers to design a system that satisfies those requirements. The system specified in the SRS will assist the potential users to determine if the system meets their needs or how the system must be modified to meet their needs

### Software Environment

The system environment specifies both software and hardware specifications.

* + - * + Tool: Visual Studio Code Python: version3
        + Operating System: Windows 7 or later Front End: CSS, HTML
        + Back end: Flask, Python

Various software used for the development of this application are the following:

* **Python**: Python is a high-level programming language that lets developers work quickly and integrate systems more efficiently.

This application is developed by using many of the Python libraries and packages such as:

* **Matplotlib**: is a cross-platform, data visualization and graphical plotting library for Python. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals.

In this application, its used for plotting the images from datasets as well as to plot the training graph

* **NumPy** is a Python library used for working with arrays. NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays.

In this application, its used for handling arrays and reshaping.

* **Seabron** a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
* **The OS** module in Python provides functions for interacting with the operating system. OS comes under Python ‘s standard utility modules. This module provides a portable way of using operating system-dependent functionality.

### OpenCV :

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.. The library has more than 2500 optimized algorithms

In this application, its used for reading and saving images.

### HTML and CSS

Hyper Text Markup Language is used for creating web pages.HTML describes the structure of the web page. Here, the user interface of my project is done using HTML. Cascading Style Sheet is used with HTML to style the web pages.

### Github

Git is an open-source version control system that was started by Linus Torvalds. Git is similar to other version control systems Subversion, CVS, and Mercurial to name a few. Version control systems keep these revisions straight, storing the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute. Git is the preferred version control system of most developers, since it has multiple advantages over the other systems available. It stores file changes more efficiently and ensures file integrity better.

The social networking aspect of GitHub is probably its most powerful feature, allowing projects to grow more than just about any of the other features offered. Project revisions can be discussed publicly, so a mass of experts can contribute knowledge and collaborate to advance a project forward

### Hardware Environment

Selection of hardware configuration is very important task related to the software development

* + - * + Processor: 2 GHz or faster (dual-core or quad-core will be much faster)
        + Memory: 8 GB RAM or greater
        + Disk space: 40 GB or greater Good internet connectivity
        + GPU :2GB

# SYSTEM DESIGN

### Model Building

* + - 1. **Model Planning**

The pothole detection dataset consist of 2 folders which contains normal images and pothole roads images. Normal images are of count 352 and pothole images are of count 329.

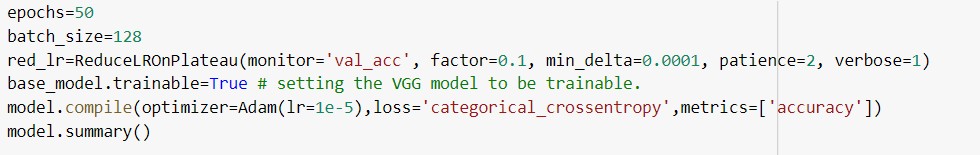
Both the labels are divided into training and testing datasets in a ratio of 70:30. The model is trained using the training dataset and tested against the test data. The model is validated against any images from the real time, can be taken from Google, or uploaded from system itself.





### Training

ResNet 50 is considered as the base model, then 1 dropout layer and 4 dense layer is added to obtain the desired architecture. The resulting fully connected layer contains 2 outputs.

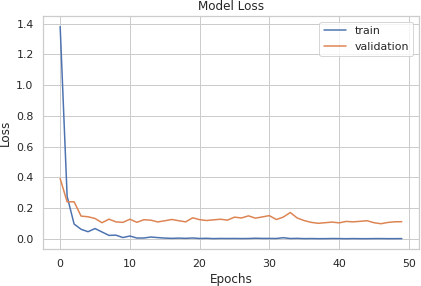
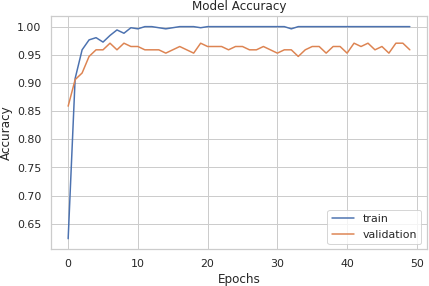


Training is performed based on different parameters like, epoch, batch size, learning rate etc.

* The number of epoch is set to 50 initially with a batch size of 128.
* Learning Rate : 0.00001
* ReduceLROnPlateau is a function that is used to monitor a value, if there is no change in the specified value, the learning rate will be adjusted automatically.
* The model gains an accuracy of 95% and a loss of 0.11.

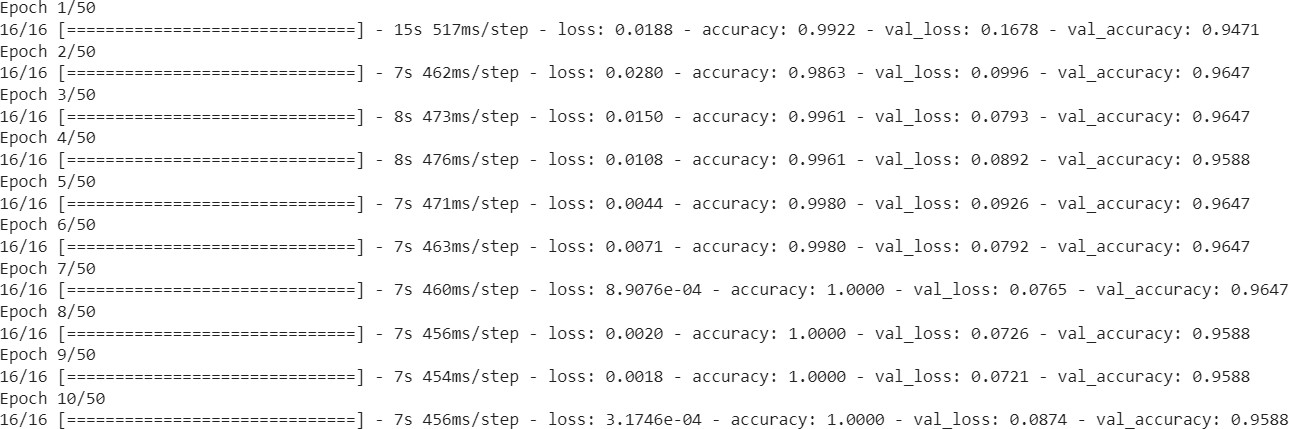
Training data is the initial dataset we use to teach an application to recognize pattern. Training dataset is used to train our model, in order to get correct predictions by the model. Images are classified into two different folders, where folder name denotes

the class. By running the 50 epochs to generate the model, and achieved an accuracy of 95% and a loss of 0.10.

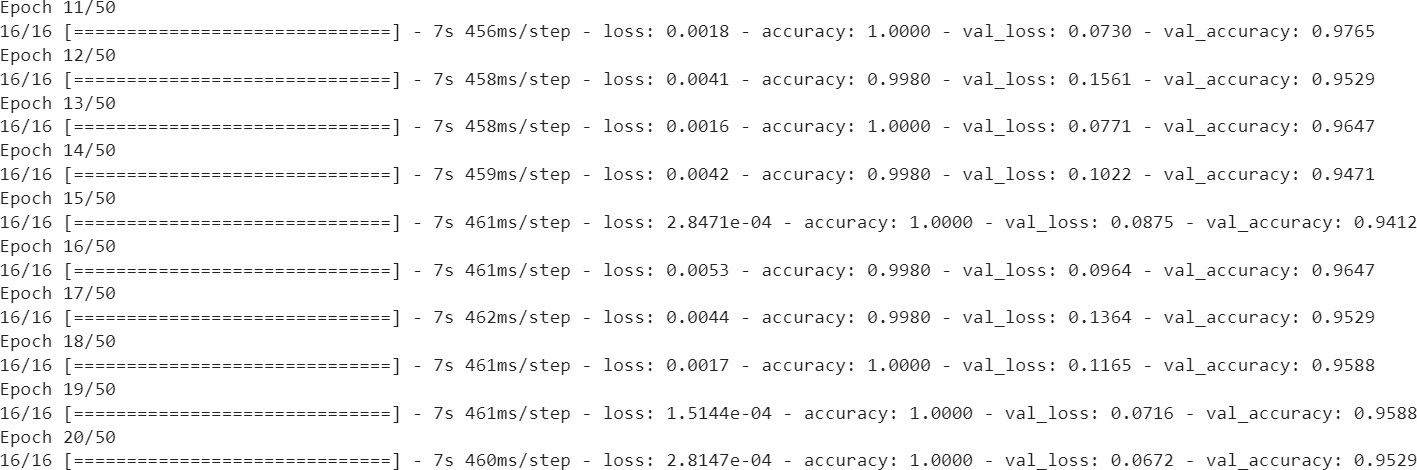


Filters/neurons detect spatial patterns such as edges in an image by detecting

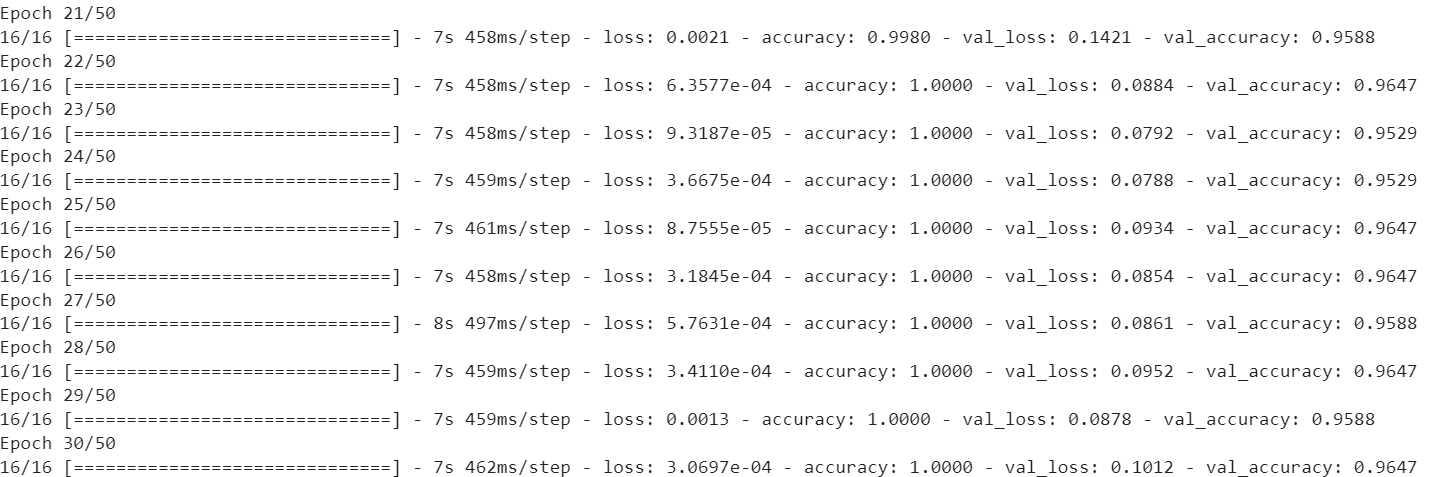
the changes in intensity values of the image. Filter size is then size of matrix using in the CNN layers. Stride is the number of pixels shifts over the input matrix. Padding is used sometimeswhen filter does not perfectly the input image, then we add zeros to make it fit. We set padding to SAME so that the input image gets fully covered by the filter and specified stride. It is called SAME because, for stride 1 , the output will be the same as the input. The featuremap captures the results of applying the filters to an input image. I.e., at each layer, the featuremap is the output of that layer.



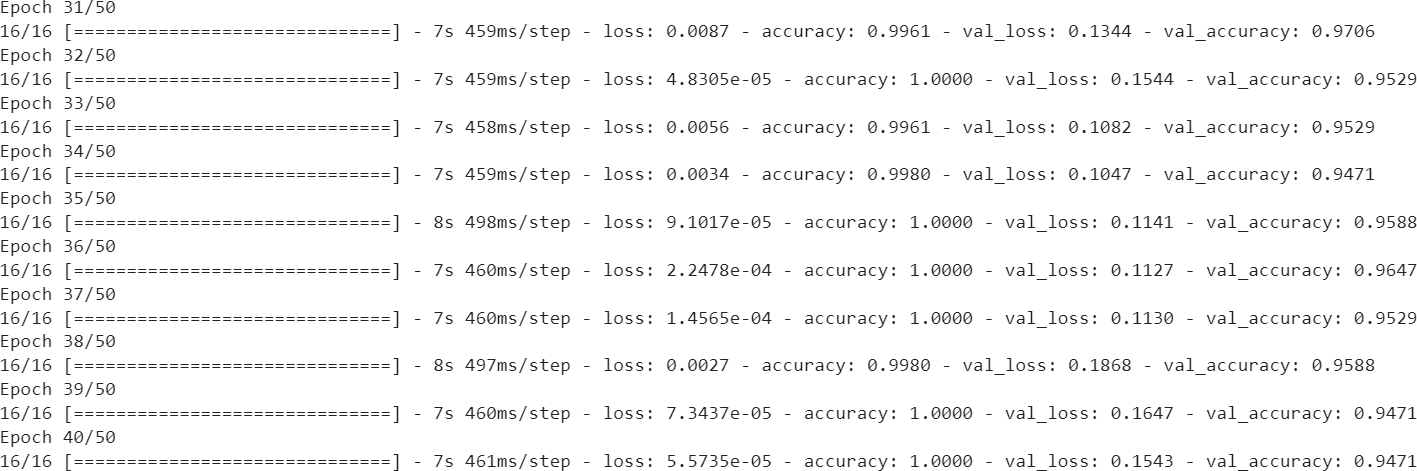
First 10 epochs of model training



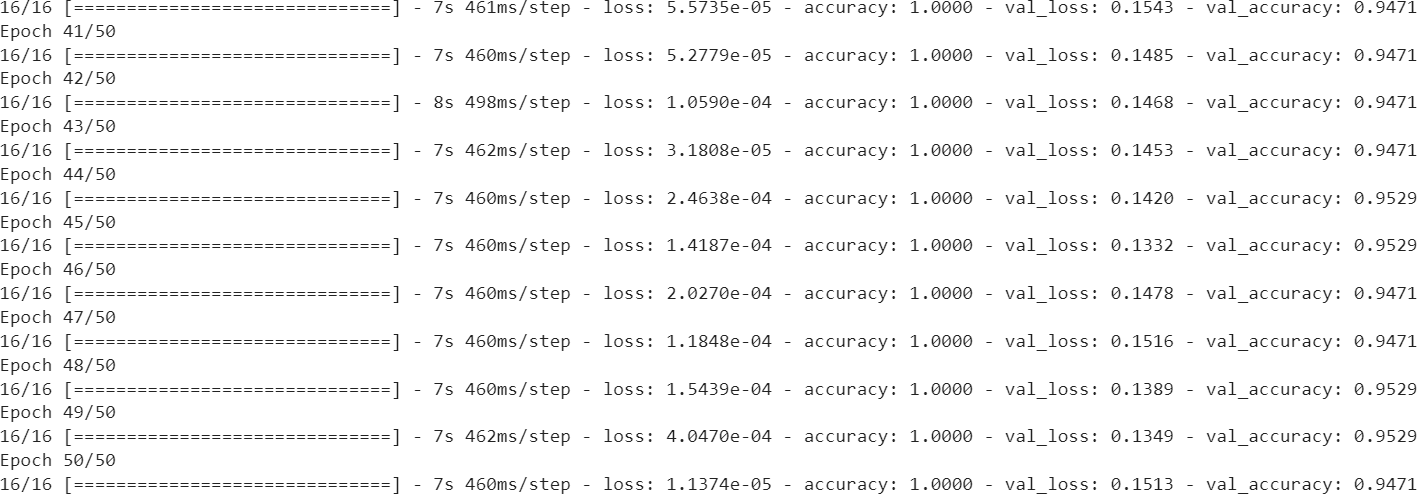
11th to 20th epochs of model training



21th to 30th epochs of model training



31th to 40th epochs of model training



41st to 50th epochs of model training

### Testing

Testing or validation data is used to evaluate our model‘s accuracy. To check whetherthe application is able to predict the output correctly. 30% of the dataset are used for testing the model.

**Epoch 31/50**

**7s 459ms/step – loss: 0.0087 –**

**accuracy: 0.9961 –**

**val\_loss: 0.1344 –**

**val\_accuracy: 0.9706**

The models are tested along with training. At each epoch the model tries to predict the unseen test data. The accuracy on the test dataset changes with each epoch and thebest model with maximum accuracy and minimum loss is selected. In the 31sr epoch, the validation accuracy is 97%.

* 1. **RESULTS AND DISCUSSION**

The aim with this project is to develop a system that predicts whether a road is of good quality or not. That is, do the roads contain potholes or not. The model is developed using the pre-trained model of CNN, ResNet 50 as the base model, with some additional layers into it. Also the dataset used, images are of different size and of different color profiles, the system is learned to handle the data from any outside source.

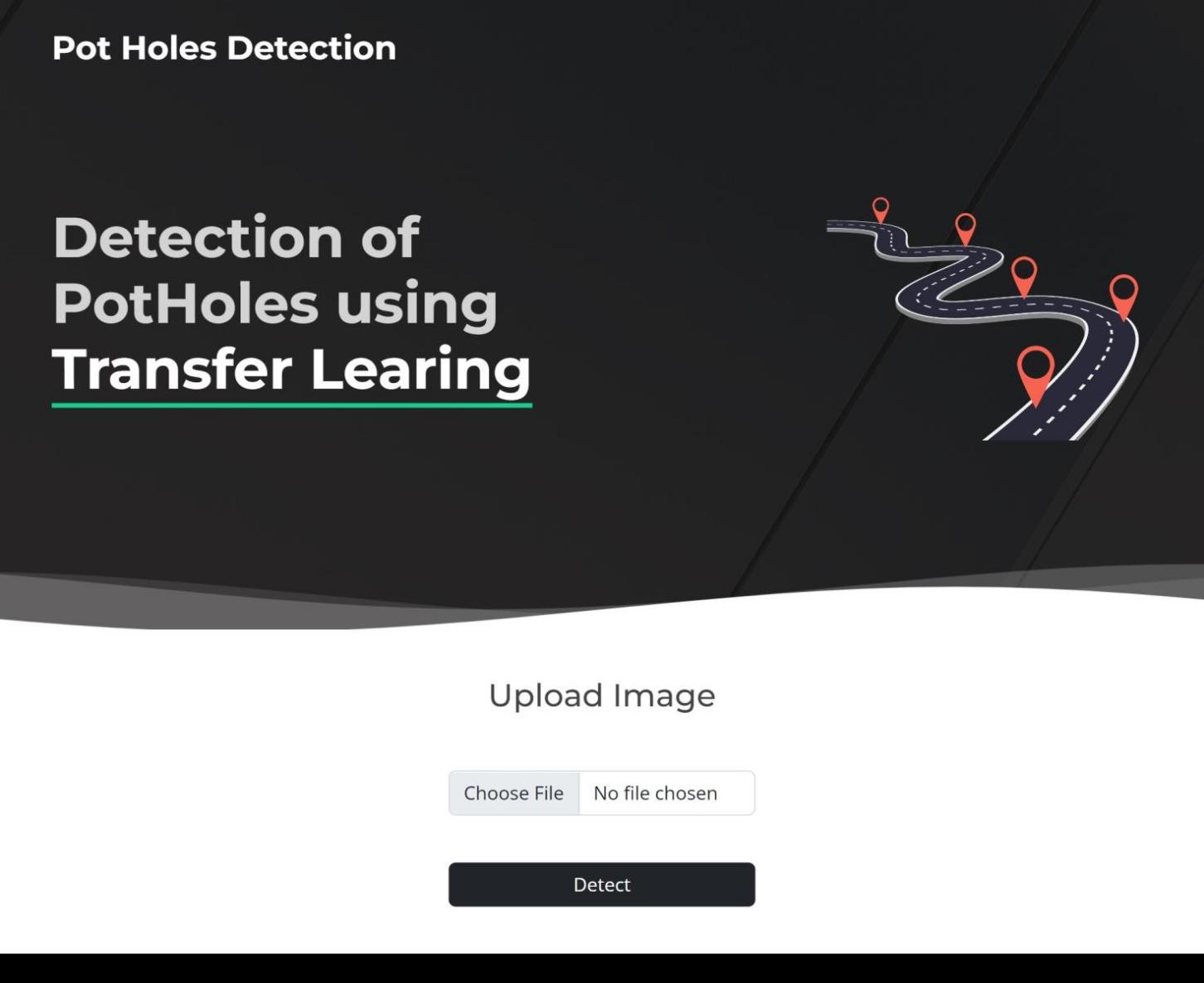
Accuracy is considered as the metrics to measure the effectiveness of the system. It is also used in the training of the dataset. Accuracy is a measurement of observational error. It defines how close or far off a given set of measurement are to their true value. Out of the total 50 epochs, I selected the model saved at the 31st epoch with validation accuracy 97.06% and minimum validation loss 0.1344. This shows that even at high validation accuracy the model may not provide the minimum validation loss.

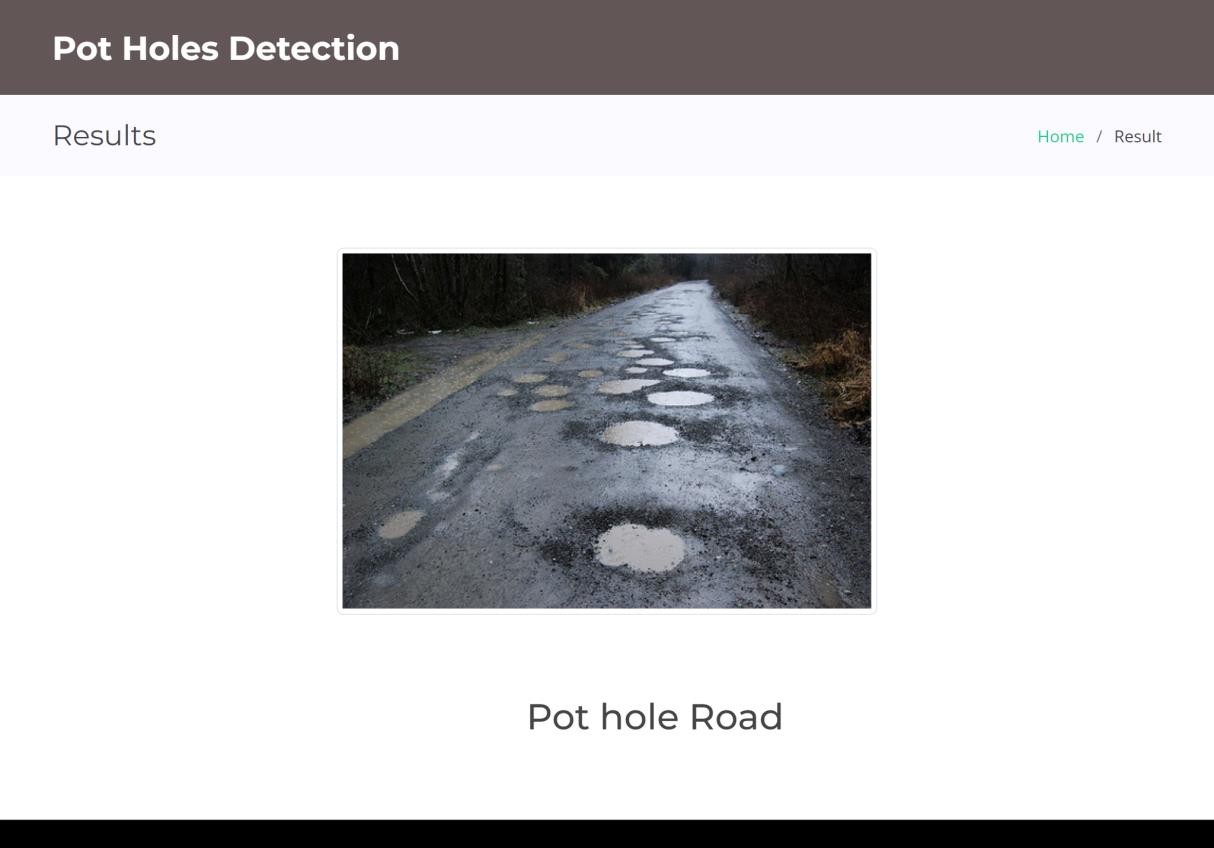
This shows that by increasing the number of epochs in training, we can improve the quality of prediction. It is like, more data for training the system is, more correct the prediction will be.

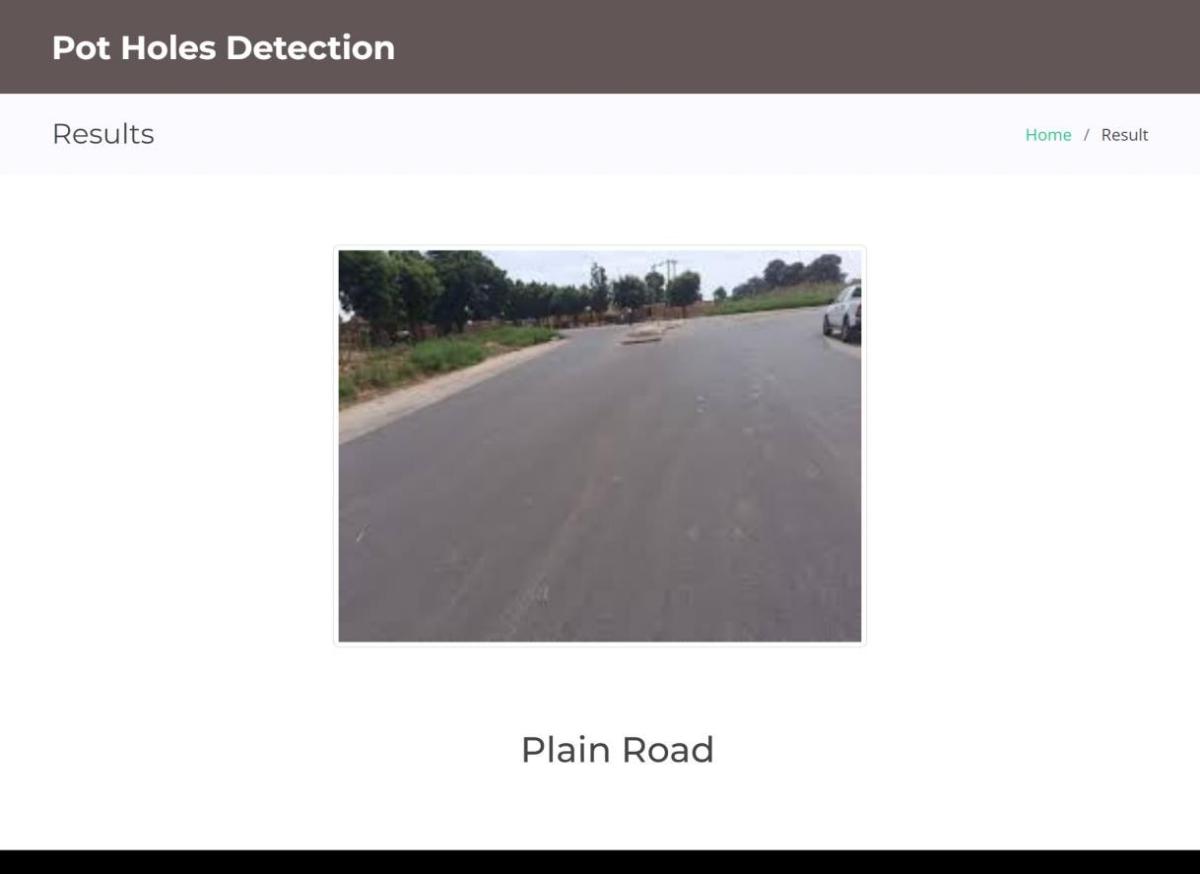
## MODEL DEPLOYMENT

This figure shows the user interface of this application. The interface is very simple and easy to understand. There are only some elements displayed on the screen. There is a file upload option provided. The user can choose image from the local storage. The prediction will be initiated when the user submits that form, the image uploaded will be used for analysis. The output will be whether the roads contain potholes or not

**UI DESIGN**

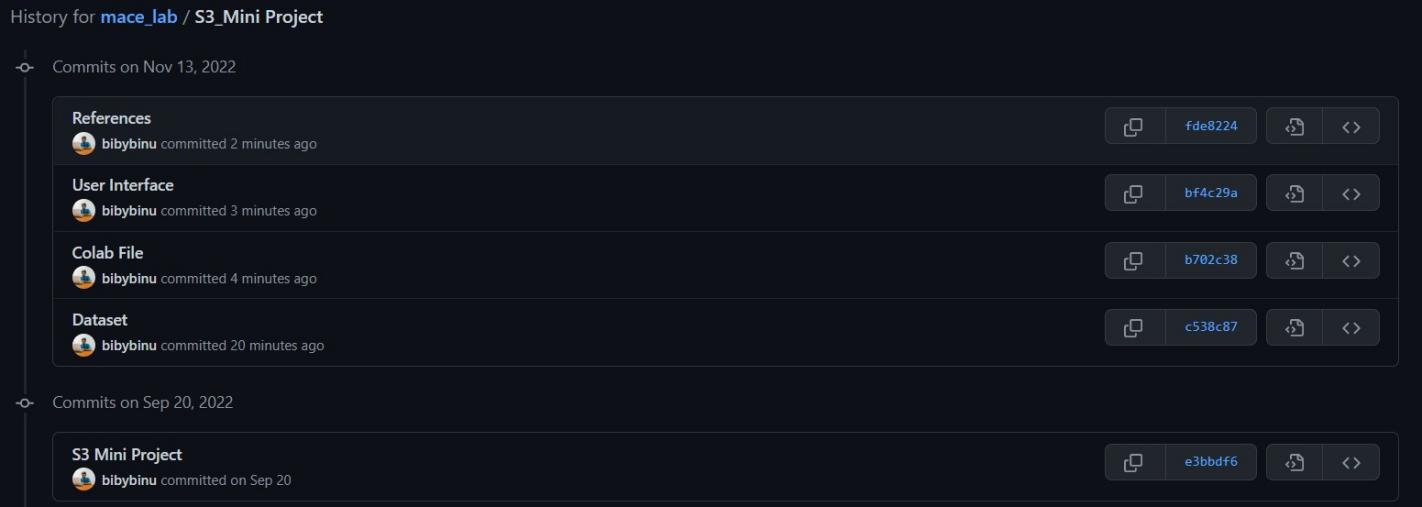


q



## GIT HISTORY

[**https://github.com/bibybinu/mace\_lab/commits/main/S3\_Mini%20Project**](https://github.com/bibybinu/mace_lab/commits/main/S3_Mini%20Project)



## CONCLUSION

As the presence of potholes can lead to various accidents which we have seen in the present days, even it leads to the death of many of lives. The system aims at reducing the number of accidents caused due to these potholes.

This is an image processing application which analyses roads images from different angles and detect whether the roads contains potholes or not. The system is developed based on the deep learning – ResNet 50 Model, which is widely used for image processing rather that image detection.

Considering the needs to detect potholes in roads accurately, this project develops efficient System based on ResNet50 to detect potholes. The system used a dataset that includes pothole images that were collected in different daylight conditions, different road conditions, and with different shapes and sizes for training which makes it accurate in all these conditions.

The model developed shows an accuracy of 97% which can detect potholes in the roads which is about 5M ahead. Training is done based on such kind of images which is somewhat 5M away from the potholes. Also the system can detect those which are too near, within the range of 1M.

The models were trainedsix to ten times for different number of epochs and finally an optimum number forty was chosenas the number of epochs for training in the final stage. Saving the model after each epoch canhelp us to choose the best model based on the validation loss and validation accuracy

## FUTURE WORK

The model is developed as a web based application, which can detect the potholes from the images we have uploaded. We can use this detection technique in Autopilot Vehicles to detect potholes. Also we can implement this in normal vehicles such that it can make the driver alert about the nearby potholes, so that he can be aware of that. Also we can change the maximum speed limit based on the roads dynamically.

Another application of this system is to analyze the satellite images to detect potholes and evaluate the conditions of the roads. For example, like in google maps, which shows the amount of traffics, from the data collected from the users, we can show the conditions of the roads, and specify whether the road is good for travelling or not

## APPENDIX

* + 1. **Minimum Software Requirements** Operating System : Windows, Linux, Google Colab

### Minimum Hardware Requirements

Hardware capacity : 256 GB (minimum)RAM : 8 GB

Processor : Intel Core i5 preferred

GPU: : 2 GB

Display : 1366 \* 768

## REFERENCES

* Ahmed, Khaled R.. (2021). Smart Pothole Detection Using Deep Learning Based on Dilated Convolution. Sensors. 21. 10.3390/s21248406.
* Rastogi, Roopak, et al. "A Comparative Evaluation of the Deep Learning Algorithms for Pothole Detection." *2020 IEEE 17th India Council International Conference (INDICON)*. IEEE, 2020.
* Becker, Yuri V. Furusho, et al. "Asphalt pothole detection in uav images using convolutional neural networks." *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019.