CHAPTER 1 : INTRODUCTION

1.1 Introduction

At present, when one suffers from particular disease, then the person has to visit to doctor which is time consuming and costly too. Also if the user is out of reach of doctor and hospitals it may be difficult for the user as the disease can not be identified. So, if the above process can be completed using a automated program which can save time as well as money, it could be easier to the patient which can make the process easier.

There are other Heart related Disease Prediction System using data mining techniques that analyzes the risk level of the patient. Disease Predictor is a web based application that predicts the disease of the user with respect to the image given by the user. Disease Prediction system has data sets collected from different health related sites. With the help of Disease Predictor the user will be able to know the probability of the disease with the given input.

In this Disease Predictor multiple health diseases can be predicted using image as an input.The user need to upload the image which will be input to predict the disease and the output will be the probability of that disease.Multiple diseases like Malaria, Lung Cancer, Breast Cancer, Hepatitis, Primary tumor, X-rays can be predicted using the Disease Predictor.

As the use of internet is growing every day, people are always curious to know different new things. People always try to refer to the internet if any problem arises. People have access to internet than hospitals and doctors. People do not have immediate option when they suffer with particular disease. So, this system can be helpful to the people as they have access to internet 24 hours.

1.2 Problem Statement

There are many tools related to disease prediction. But particularly heart related diseases have been analyzed and risk level is generated. But generally there are no such tools that are used for prediction of general diseases. So Disease Predictor helps for the prediction of the general diseases. In this we have consider Malaria for prediction.

1.3 Objective

1.3.1 General Objective

To implement ResNet that can predict and analyze the disease as per the input of the user.

1.3.2 Specific Objective

To develop web interface platform for the prediction of the disease.

1.4 Scope And Limitations

1.4.1 Scope

This project aims to provide a web platform to predict the occurrences of disease on the basis of various symptoms. The user will upload image and can find the diseases with their probabilistic figures.

1.4.2 Limitations

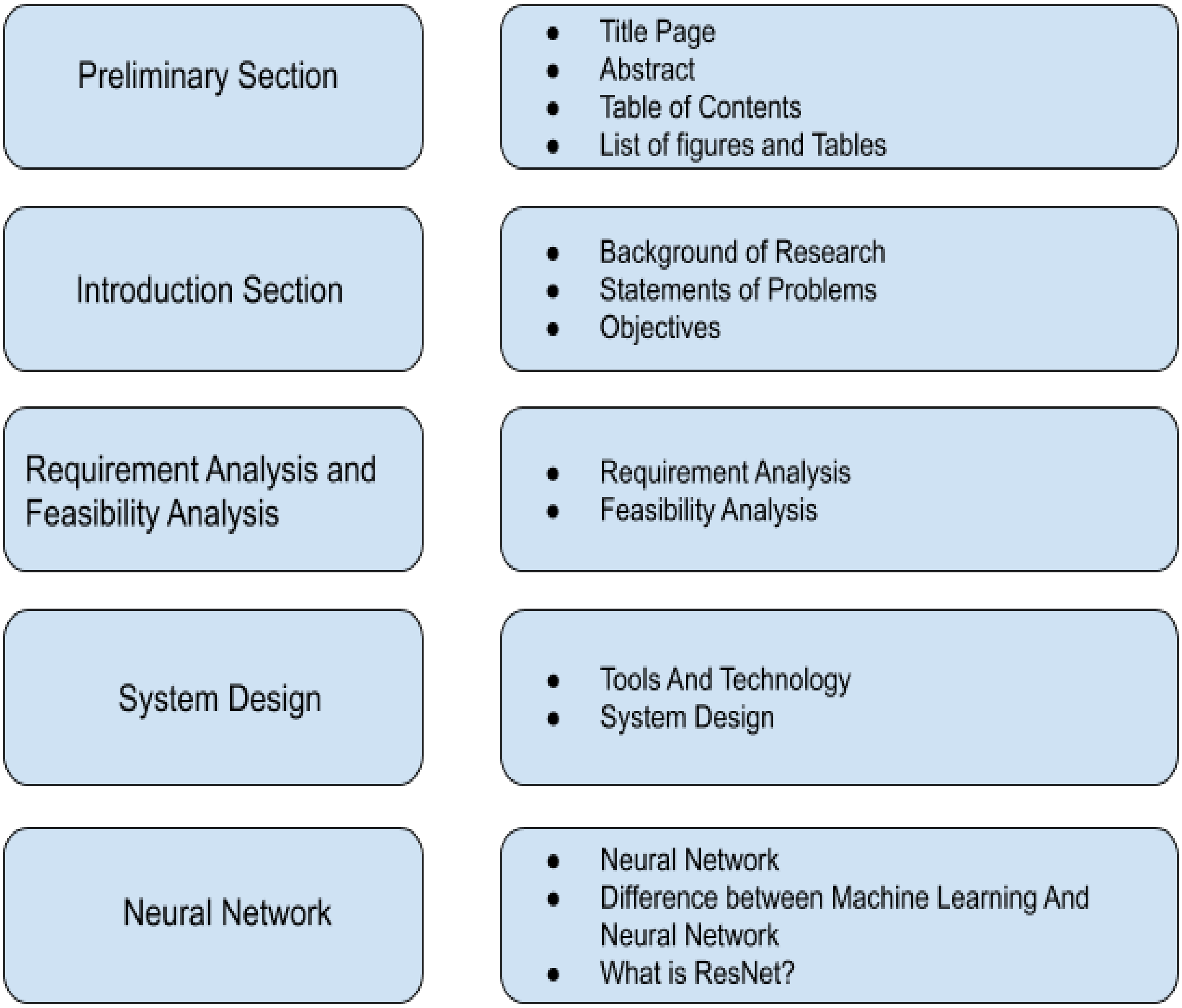
The limitations of this project are:

a. Disease Predictor does not recommend medications of the disease.

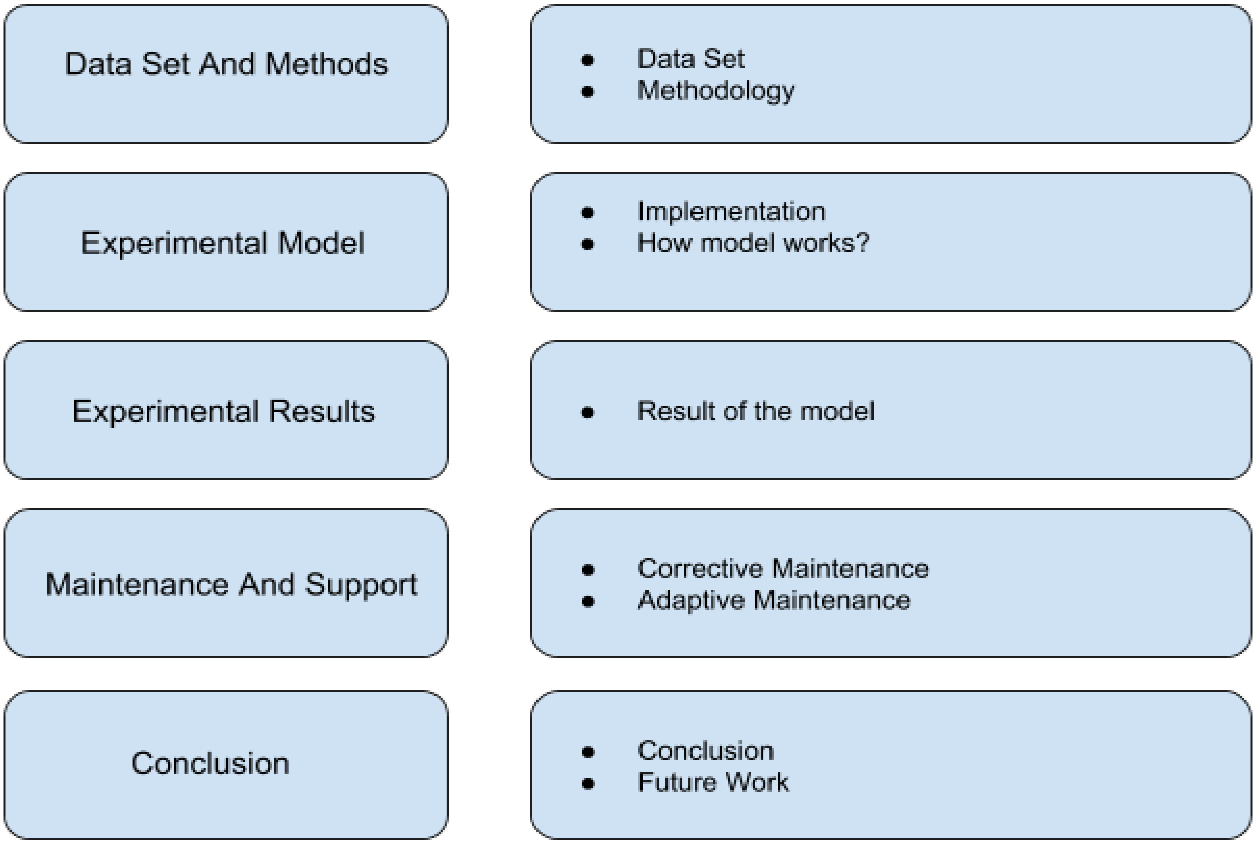
b. Past history of the disease has not been considered.

c. Input image should be not more than given size.

1.5 Outline of Document



Preliminary Section	• Title Page • Abstract • Table of Contents • List of figures and Tables
Introduction Section	• Background of Research • Statements of Problems • Objectives
Requirement Analysis and Feasibility Analysis	• Requirement Analysis • Feasibility Analysis
System Design	• Tools And Technology • System Design
Neural Network	• Neural Network • Difference between Machine Learning And Neural Network • What is ResNet?



Data Set And Methods

- Data Set
- Methodology

Experimental Model

- Implementation
- How model works?

Experimental Results

- Result of the model

Maintenance And Support

- Corrective Maintenance
- Adaptive Maintenance

Conclusion

- Conclusion
- Future Work

Fig 1 : Outline of the Document

CHAPTER 2 : REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

2.1 Requirement Analysis

2.1.1 Function Analysis

a. Predict disease with the given input image .

b. Compare the given input image with the input datasets .

2.1.2 Non-Functional Analysis

a. Display the list of symptoms where user can select the symptoms.

b. ResNet is used to predict the disease sets.

2.2 Feasibility Analysis

2.2.1 Technical Feasibility

- The project is technically feasible as it can be built using the existing available technologies.
- It is a web based applications that uses Flask Framework.
- The technology required by Disease Predictor is available and hence it is technically feasible.

2.2.2 Economic Feasibility

- The project is economically feasible as the cost of the project is involved only in the hosting of the project.
- As the data samples increases, which consume more time and processing power. In that case better processor might be needed.

2.2.3 Operational Feasibility

- The project is operationally feasible as the user having basic knowledge about computer and Internet.
- Disease Predictor is based on client-server architecture where client is users and server is the machine where datasets are stored.

CHAPTER 3 : SYSTEM DESIGN

3.1 Tools And Technology

3.1.1 Python3

Python is a fun programming language with great libraries for machine learning and computer vision.

3.1.2 OpenCV

OpenCV is a popular framework for computer vision and image processing. We'll use OpenCV to process the images. It has a Python API so we can use it directly from Python.

3.1.3 Keras

Keras is a deep learning framework written in Python. It makes it easy to define, train and use deep neural networks with minimal coding.

3.1.4 TensorFlow

TensorFlow is Google's library for machine learning. We'll be coding in Keras, but Keras doesn't actually implement the neural network logic itself. Instead, it uses Google's TensorFlow library behind the scenes to do the heavy lifting.

3.1.5 Flask

Flask is the framework which supports Python which is used for designing Web Interface.

3.2 Related Work

K.M. Al-Aidaroos, A.A. Bakar and Z. Othman have conducted the research for the best medical diagnosis mining technique. For this authors compared Naïve Baeyes with five other classifiers i.e. Logistic Regression (LR), KStar (K*), Decision Tree (DT), Neural Network (NN) and a simple rule-based algorithm (ZeroR). For this, 15 real-world medical problems from the UCI machine learning repository (Asuncion and Newman, 2007) were selected for evaluating the performance of all algorithms. In the experiment it was found that NB outperforms the other algorithms in 8 out of 15 data sets so it was concluded that the predictive accuracy results in Naïve Baeyes is better than other techniques.

Medical Problems	NB	LR	K*	DT	NN	ZeroR
Breast Cancer wise	97.3	92.98	95.72	94.57	95.57	65.52
Breast Cancer	72.7	67.77	73.73	74.28	66.95	70.3
Dermatology	97.43	96.89	94.51	94.1	96.45	30.6
Echocardiogram	95.77	94.59	89.38	96.41	93.64	67.86
Liver Disorders	54.89	68.72	66.82	65.84	68.73	57.98
Pima Diabetes	75.75	77.47	70.19	74.49	74.75	65.11
Haeberman	75.36	74.41	73.73	72.16	70.32	73.53
Heart-c	83.34	83.7	75.18	77.13	80.99	54.45
Heart-statlog	84.85	84.04	73.89	75.59	81.78	55.56
Heart-b	83.95	84.23	77.83	80.22	80.07	63.95
Hepatitis	83.81	83.89	80.17	79.22	80.78	79.38
Lung Cancer	53.25	47.25	41.67	40.83	44.08	40
Lymphography	84.97	83.18	78.21	81.81	54.76	78.45
Postoperative Patient	68.11	61.11	61.67	69.78	58.54	71.11
Primary tumor	49.71	41.62	38.02	41.39	40.38	24.78
Wins	8/15	5/15	0	2/15	1/15	1/15

(Al-Aidaroos, Bakar, & Othman, 2012).

[1]

Table 1 : Accuracy Table of different techniques for Health Disease Prediction

Darcy A. Davis, Nitesh V. Chawla, Nicholas Blumm, Nicholas Christakis, Albert-Laszlo Barabasi have found that global treatment of chronic disease is neither time or cost efficient. So the authors conducted this research to predict future disease risk. For this CARE was used (which relies only on a patient's medical history using ICD- 9-CM codes in order to predict future diseases risks). CARE combines collaborative filtering methods with clustering to predict each patient's greatest disease risks based on their own medical history and that of similar patients. Authors have also described an

Iterative version, ICARE, which incorporates ensemble concepts for improved performance. These novel systems require no specialized information and provide predictions for medical conditions of all kinds in a single run. The impressive future disease coverage of ICARE represents more accurate early warnings for thousands of diseases, some even years in advance. Applied to full potential, the CARE framework can be used explore a broader disease histories, suggest previously unconsidered concerns, and facilitating discussion about early testing and prevention. [2]
(A.Davis, V.Chawla, Blumm, Christakis, & Barbasi, 2008).

Jyoti Soni, Ujma Ansari, Dipesh Sharma and Sunita Soni have done this research research paper into provide a survey of current techniques of knowledge discovery in databases using data mining techniques that are in use in today's medical research particularly in Heart Disease Prediction. Number of experiment has been conducted to compare the performance of predictive data mining technique on the same dataset and the outcome reveals that Decision Tree outperforms and some time Bayesian classification is having similar accuracy as of decision tree but other predictive methods like KNN, Neural Networks, Classification based on clustering is not performing well. [3]
(Jyoti Soni, Ansari, Sharma, & Soni, 2011)

Shadab Adam Pattekari and Asma Parveen have conducted a research using Naïve Bayes Algorithm to predict the heart diseases where user provides the data which is compared with trained set of values. So from this research, patients were able to provide their basic information which is compared with the data and the heart disease is predicted. [4] (Adam & Parveen, 2012)

M.A.Nishara Banu, B Gomathy used medical data mining techniques like association rule mining, classification, clustering I to analyze the different kinds of heart based problems. Decision tree is made to illustrate every possible outcome of a decision. Different rules are made to get the best outcome. In this research age , sex, smoking, overweight, alcohol intake, blood sugar, heart rate, blood pressure are the parameters used for making the decisions. Risk level for different parameters are stored with their id's ranging (1-8). ID lesser than of 1 of weight contains the normal level of prediction and higher ID other than 1 comprise the higher risk levels .K-means clustering technique is used to study the pattern in the dataset. The algorithm clusters informations into k groups. Each point in the dataset is assigned to the closed cluster. Each cluster center is recomputed as the average of the points in that cluster. [5]
(NisharBanu, MA; Gomathy, B;, 2013)

3.3 SYSTEM DESIGN

3.3.1 Class Diagram

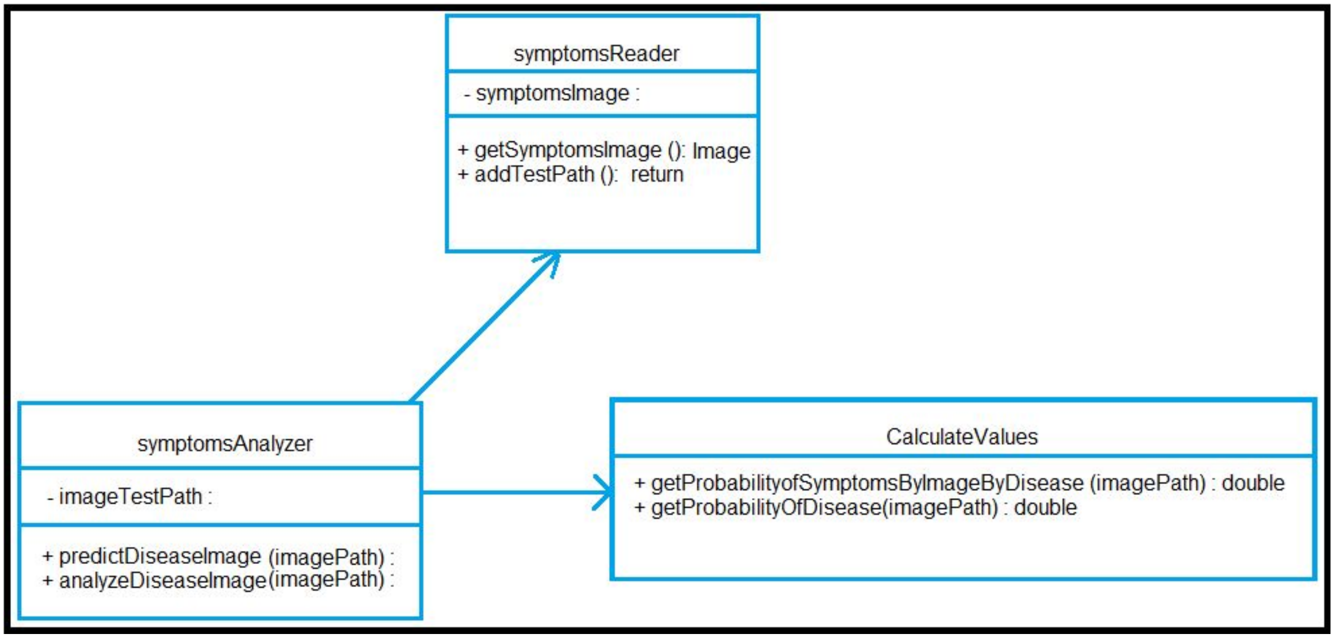

symptomsReader
- symptomsImage :
+ getSymptomsImage (): Image
+ addTestPath (): return
symptomsAnalyzer
- imageTestPath :
+ predictDiseaseImage (imagePath) :
+ analyzeDiseaseImage(imagePath) :
CalculateValues
+ getProbabilityofSymptomsByImageByDisease (imagePath) : double
+ getProbabilityOfDisease(imagePath) : double

Fig 2 : Class Diagram

It explain the classes used in the Disease Predictor. There are three classes used in total,

- **Symptoms Reader:** Reads the user input and creates the test path for input image.
- **Symptoms Analyzer:** According to symptoms image displays the subjective result.
- **Calculate Values:** Calculates the probabilistic model of the diseases.

3.3.2 State Diagram

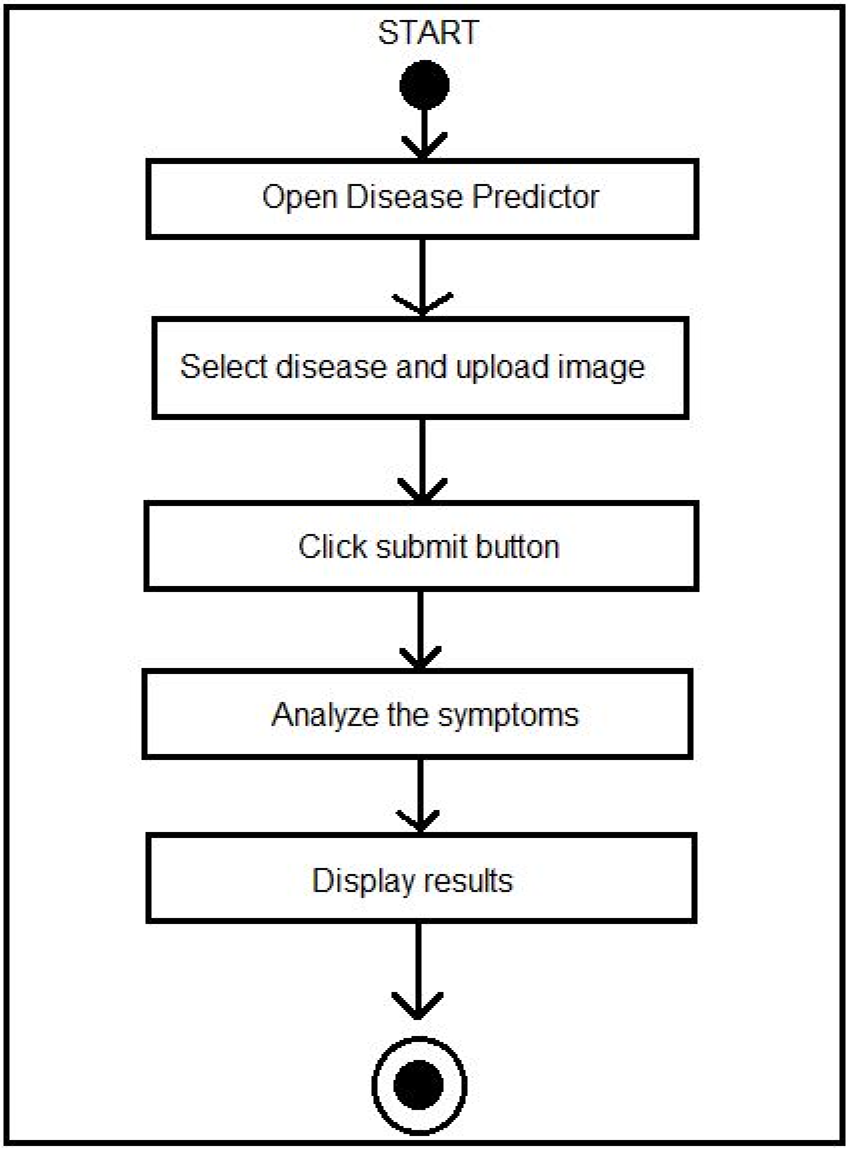


Fig 3 : State Diagram

It explains different state of the system. First the user opens Disease Predictor. The user selects the symptoms and upload the image. When finished selecting symptoms the user submits the image. Disease Predictor analyzes the symptoms and displays the result.

3.3.3 Sequence Diagram

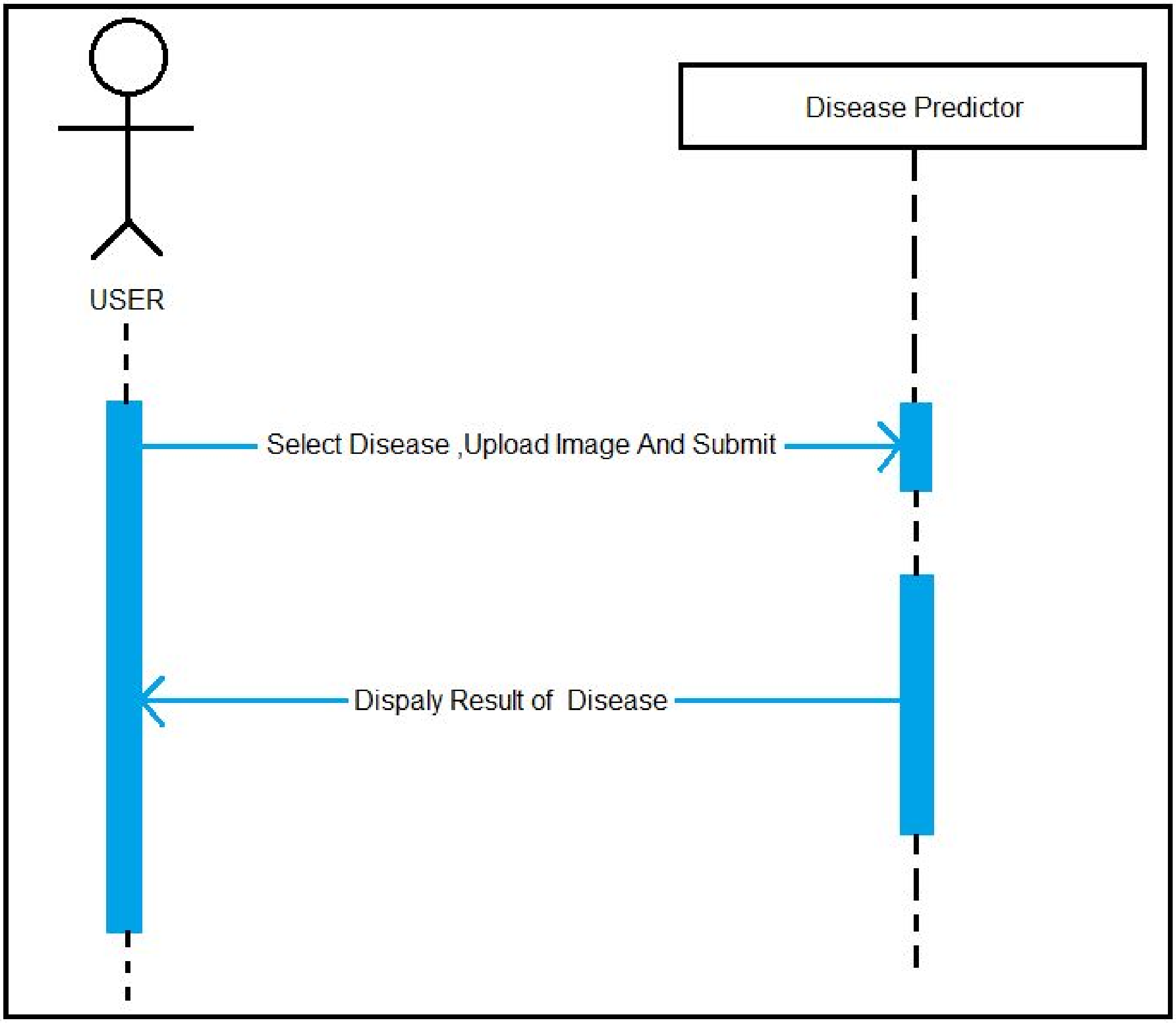

USER
Disease Predictor
Select Disease ,Upload Image And Submit
Dispaly Result of Disease

Fig 4 : Sequence Diagram

It explains the sequence of the Disease Predictor. Initially system shows the disease or symptoms to be selected. The user selects the symptoms and submits to the system .The Disease Predictor predicts and display the result.

CHAPTER 4 : INTRODUCTION TO NEURAL NETWORK

4.1 What Is NEURAL NETWORK?

In our project, we will use neural networks to predict and analyze the disease.. Neural networks are inspired by the brain. In 1943, McCulloch and Pitts [6] laid the formal foundation for the field of artificial neural networks. Because neural networks were computationally expensive and there was no good learning algorithm for training the models, they became unpopular in the 1960’s. Marvin Minsky and Seymour Papert made this clear in their paper [9]. Another key advance that came later was the backpropagation algorithm, which effectively solved the neural network problem, given by Werbos [10]. In the early 1980's, researchers showed renewed interest in neural networks. From 2005 onwards, they have again become popular, as computers have now become fast enough to do large computations. People have also achieved success in training neural networks with the SGD (stochastic gradient descent) algorithm, which fortunately works, but does not have clear theoretical justification. In addition, neural network models are big, and thus require a lot of training data. With the recent advances in Big Data, it has become very easy to collect training data. They are the hottest area in the field of machine learning [7].

4.2 Difference Between Machine Learning And Neural Network

Neural networks work very well with different machine learning problems. Neural networks are a type of machine learning algorithm. The basic difference between machine learning and conventional programming languages is that in conventional programming, a computer has to be explicitly programmed. We ourselves have to write and maintain the code. But in case of neural networks, the network adapts itself to the problem during training. In conventional programming style, people have to understand the problem well and research different approaches. Since a clear solution is often elusive in practical problems,people tend to use heuristics, which work for some use cases but do not generalize well. But with machine learning, we have the model itself learn from the data. Machines tend to learn faster (vis a vis core research), so they take less time in solving problems. Machine learning algorithms like neural networks also do a good job with generalization. Some of the applications of neural networks include facial recognition as used by Facebook to tag photos [11], image captioning [8] by Google and etc.

Neural networks are essentially a bunch of interconnected elements called neurons. They are an information processing paradigm which is inspired by biological nervous systems. The nervous system contains around 1010 neurons. Figure shows a single neuron.

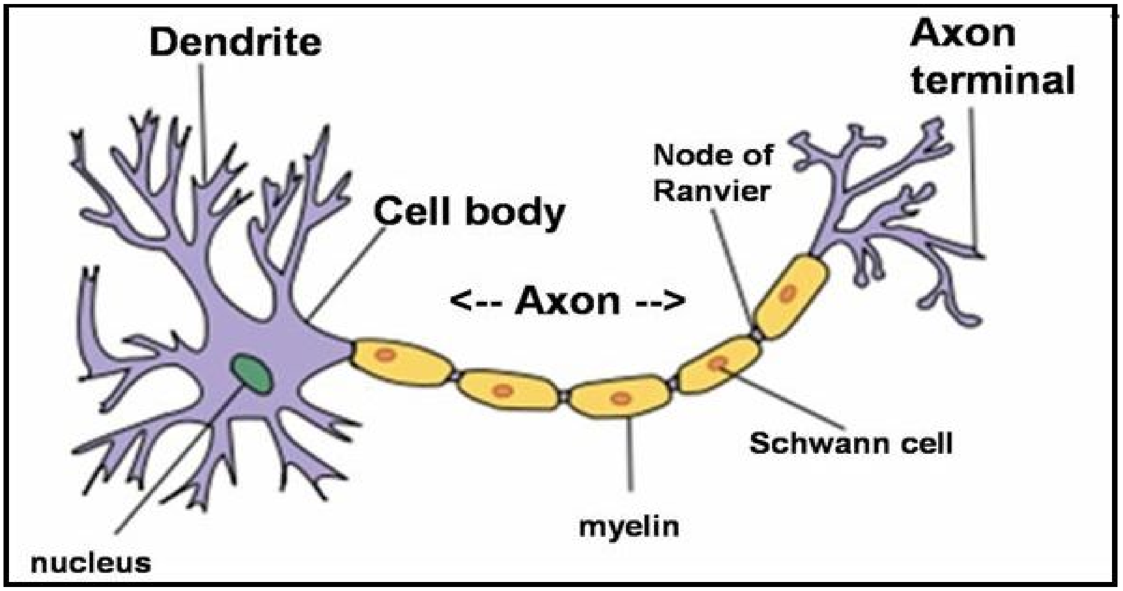


Fig 5 : Biological Neuron

Each biological neuron consists of a cell body. It contains lots of dendrites, which bring Electrical signals into the cell, and an axon, which transmits these signals out of the cell. A neuron fires when the collective effect of its inputs reaches a certain threshold. The axons of neurons are dependent on each other and can influence the dendrites of another neuron. Similarly, a neural network starts with a model, as in Figure . It consists of several inputs and a single output. Every input is modified by a weight and multiplied with the input value. The neuron combines these weighted inputs and, with the help of the threshold value and activation function,determines its output. **The objective of the neural network is to transform the inputs into a meaningful output.**

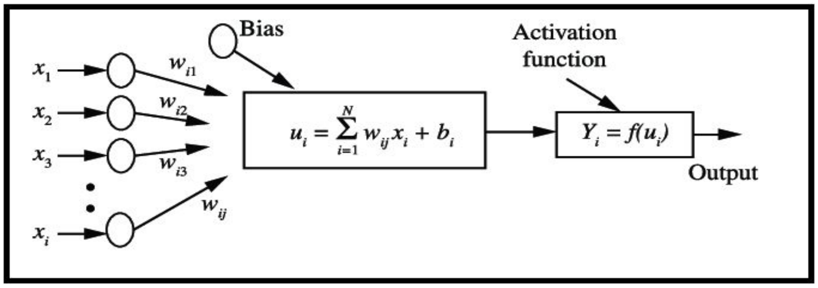


Fig 6 : A Simple Neural Network Model

4.3 What is ResNet ?

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers---8x deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers. The depth of representations is of central importance for many visual recognition tasks.

According to the universal approximation theorem, given enough capacity, we know that a feedforward network with a single layer is sufficient to represent any function. However, the layer might be massive and the network is prone to overfitting the data. Therefore, there is a common trend in the research community that our network architecture needs to go deeper.

Since AlexNet, the state-of-the-art CNN architecture is going deeper and deeper. While AlexNet had only 5 convolutional layers, the VGG network [13] and GoogleNet (also codenamed Inception_v1) [14] had 19 and 22 layers respectively.

However, increasing network depth does not work by simply stacking layers together. Deep networks are hard to train because of the notorious vanishing gradient problem — as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitely small. As a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly.

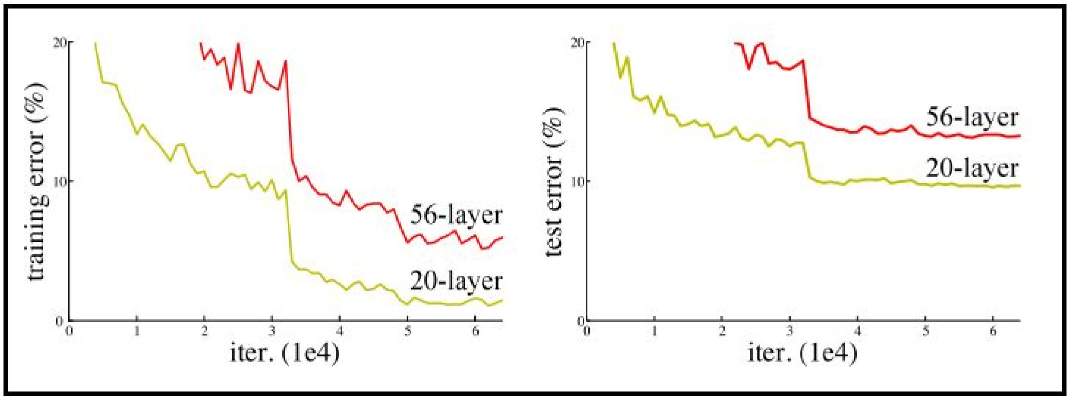


Fig 7 : Increasing network depth leads to worse performance

Before ResNet, there had been several ways to deal the vanishing gradient issue, for instance, [14] adds an auxiliary loss in a middle layer as extra supervision, but none seemed to really tackle the problem once and for all.

The core idea of ResNet is introducing a so-called “identity shortcut connection” that skips one or more layers, as shown in the following figure:

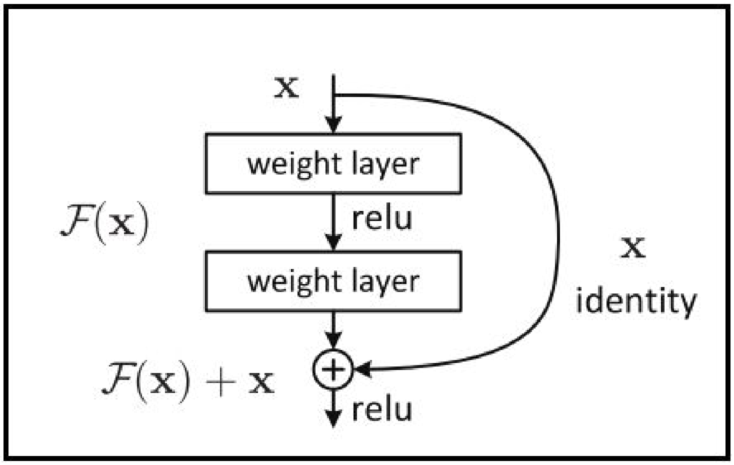

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig 8 : A residual block

The authors of [12] argue that stacking layers shouldn’t degrade the network performance, because we could simply stack identity mappings (layer that doesn’t do anything) upon the current network, and the resulting architecture would perform the same. This indicates that the deeper model should not produce a training error higher than its shallower counterparts. They hypothesize that letting the stacked layers fit a residual mapping is easier than letting them directly fit the desired underlying mapping. And the residual block above explicitly allows it to do precisely that.

As a matter of fact, ResNet was not the first to make use of shortcut connections, Highway Network [15] introduced gated shortcut connections. These parameterized gates control how much information is allowed to flow across the shortcut. Similar idea can be found in the Long Term Short Memory (LSTM) [16] cell, in which there is a parameterized forget gate that controls how much information will flow to the next time step. Therefore, ResNet can be thought of as a special case of Highway Network.

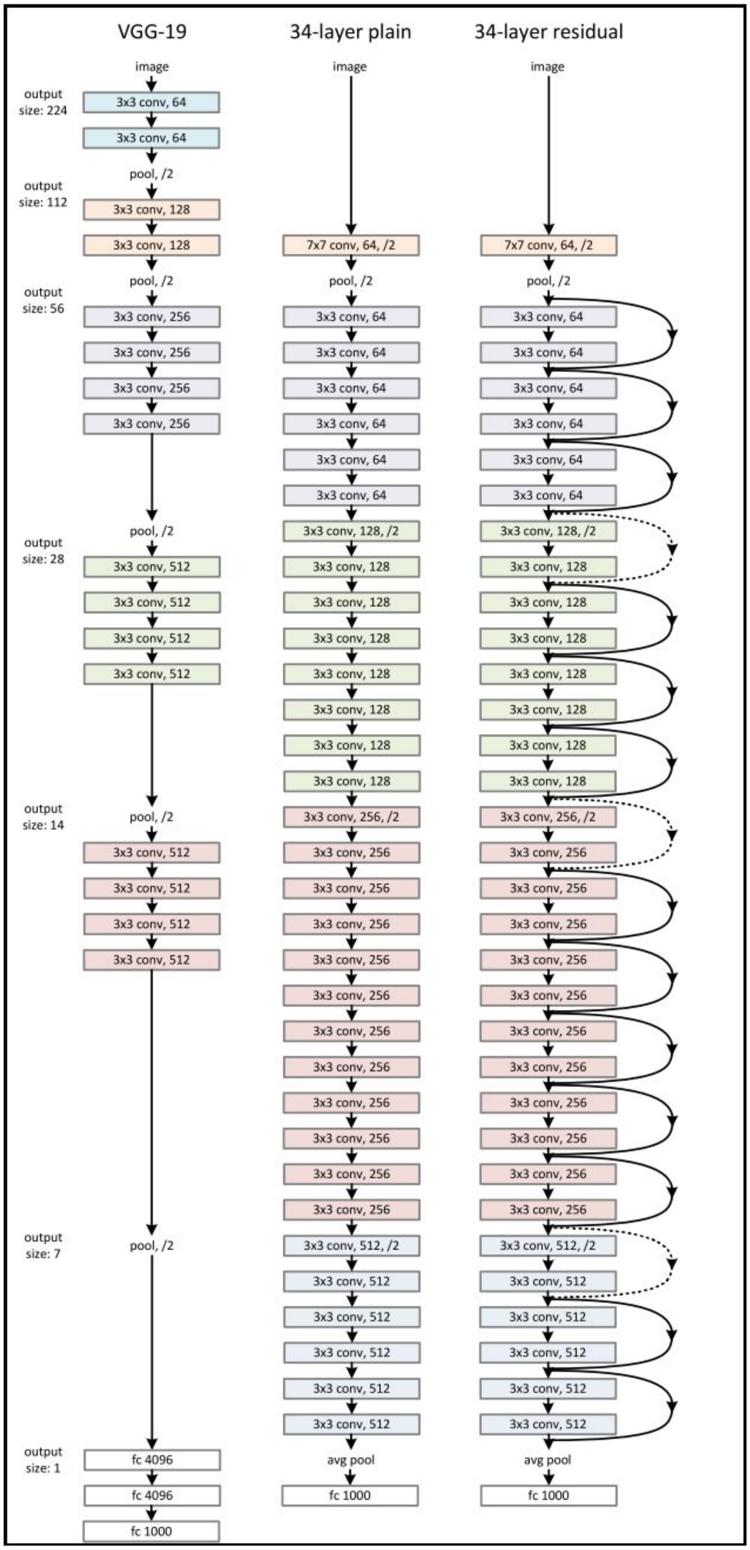
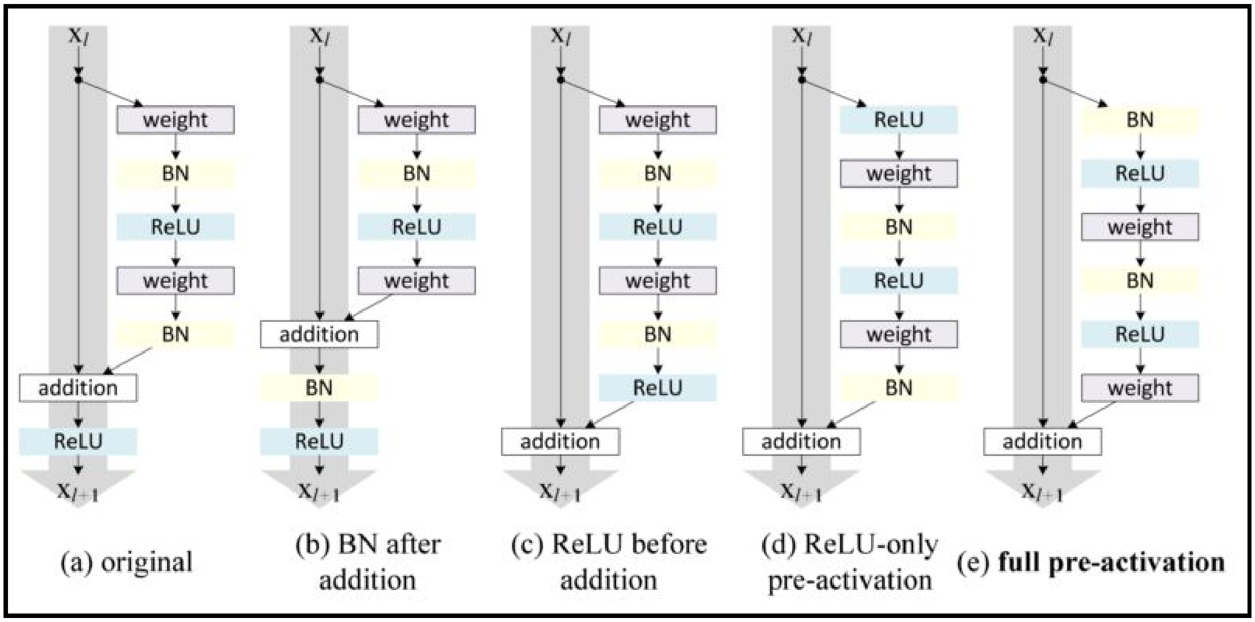


Fig 9 : The ResNet architecture

However, experiments show that Highway Network performs no better than ResNet, which is kind of strange because the solution space of Highway Network contains ResNet, therefore it should perform at least as good as ResNet. This suggests that it is more important to keep these “gradient highways” clear than to go for larger solution space.

Following this intuition, the authors of [12] refined the residual block and proposed a pre-activation variant of residual block [17], in which the gradients can flow through the shortcut connections to any other earlier layer unimpededly. In fact, using the original residual block in [12], training a 1202-layer ResNet resulted in worse performance than its 110-layer counterpart.



(a) original (b) BN after addition (c) ReLU before addition (d) ReLU-only pre-activation (e) **full pre-activation**

Fig 10 : Variants of residual blocks

CHAPTER 5 : DATA SET AND METHODS

5.1 Data Set

Data collection has been done from the internet to identify the disease here the real symptoms of the disease are collected i.e. no dummy values are entered. The symptoms of the disease are collected from different health related websites.

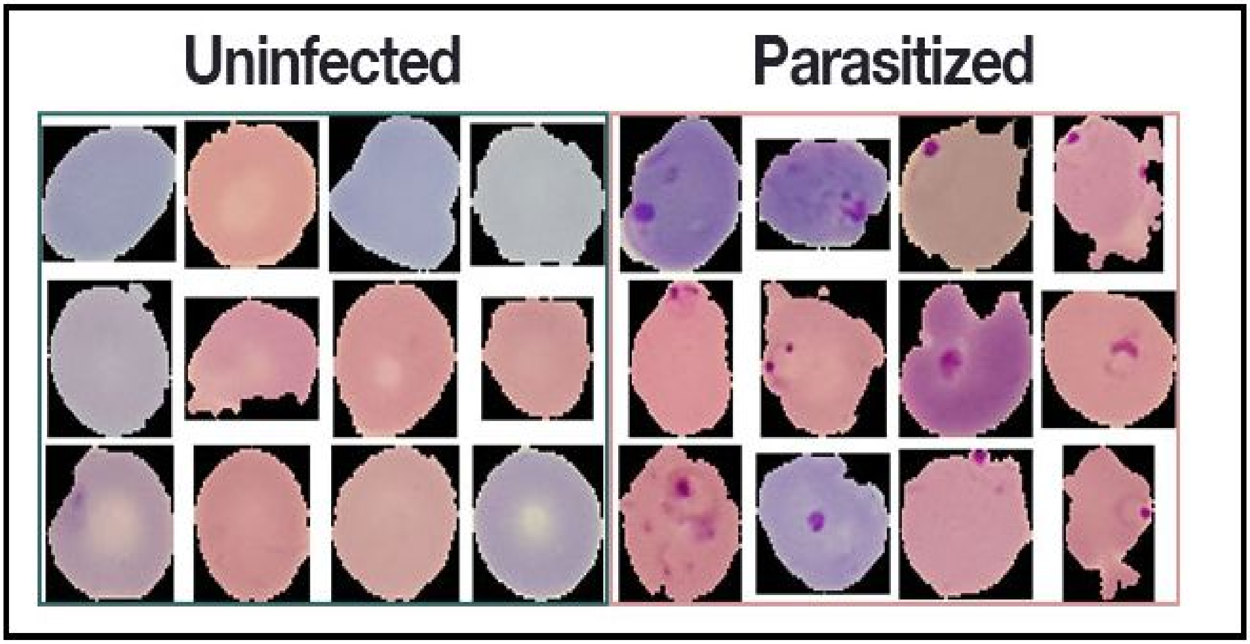


Fig 11 : A subset of Dataset of Malaria

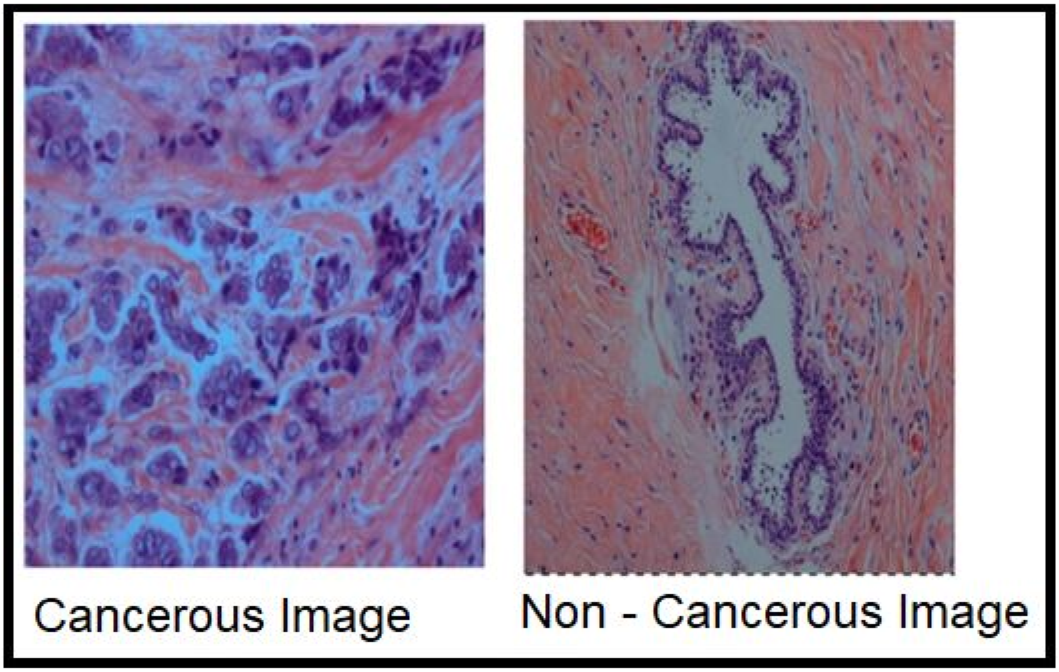


Fig 12 : A Subset of Dataset of Cancer

A subset of the Malaria Dataset provided by the National Institute of Health (NIH). We will use this dataset to develop a deep learning medical imaging classification model with Python, OpenCV, and Keras.

5.2 Methodology

Disease Prediction has been already implemented using different techniques like Neural Network, decision tree and Naïve Byes algorithm. Particularly heart related disease is mostly analyzed. From the analysis it was found that Neural Network and Deep Neural Network is more accurate than other techniques. So, Disease Predictor also uses Deep Neural Network for the prediction of different diseases.

CHAPTER 6 : EXPERIMENTAL MODEL

6.1 Implementation

Disease Predictor is the ability to predict the disease that has been provided to the system. For disease prediction, we need to implement the ResNet.

6.1.1 Building our deep learning + medical image dataset

Our malaria dataset does not have pre-split data for training, validation, and testing so we'll need to perform the splitting ourselves.

6.1.2 Training a deep learning model for medical image analysis

Now that we've created our data splits, let's go ahead and train our deep learning model for medical image analysis.

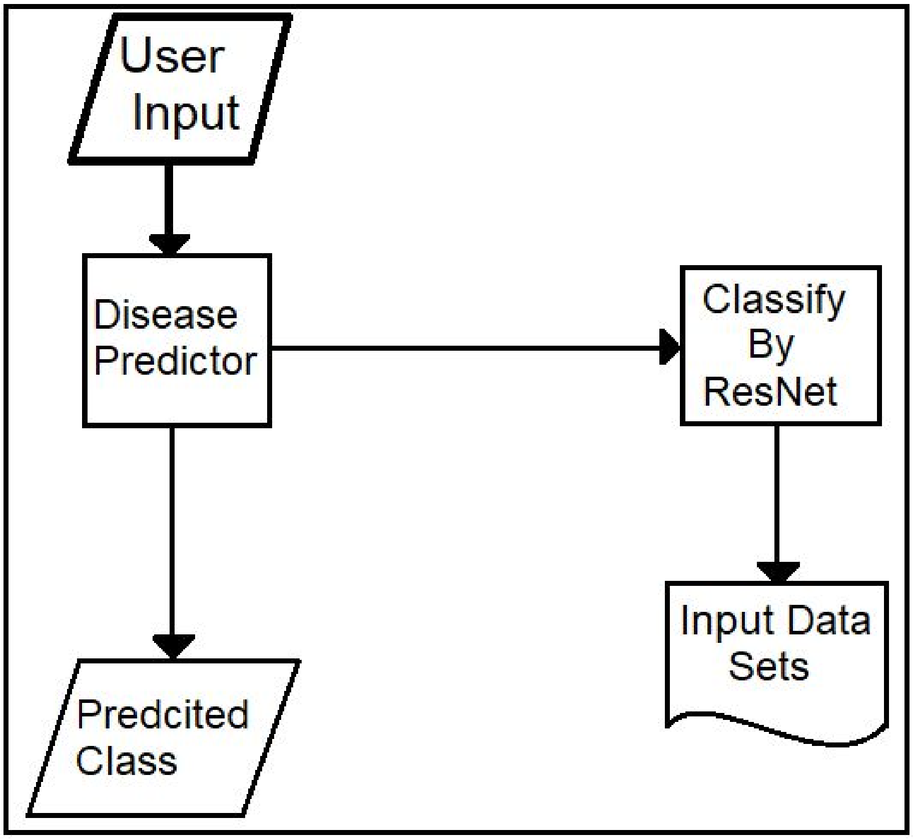
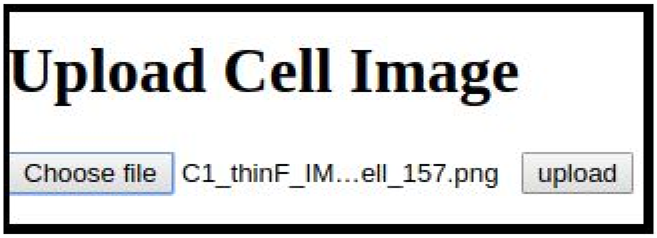


Fig 13 : Workflow

6.2 How model works?



Upload Cell Image

C1_thinF_IM…ell_157.png

Fig 14 : User Interface

Here the user interface is shown in figure.In this user will upload the image of the blood cell.And the images can be uploaded as many as possible.But for testing only one clear image is sufficient.

After uploading the image user has to just wait for the result.

CHAPTER 7 : EXPERIMENTAL RESULTS

7.1 Result of Model

Overall, the entire training process took *only 6 hours* (*significantly* faster than the 24-hour training process of NIH's method). At the end of the 50th epoch we are obtaining:

- **96.50% accurac**y on the *training* data
- **96.78% accuracy** on the *validation* data
- **97% accuracy** on the *testing* data

There are a number of benefits to using the ResNet-based model we trained here for medical image analysis.

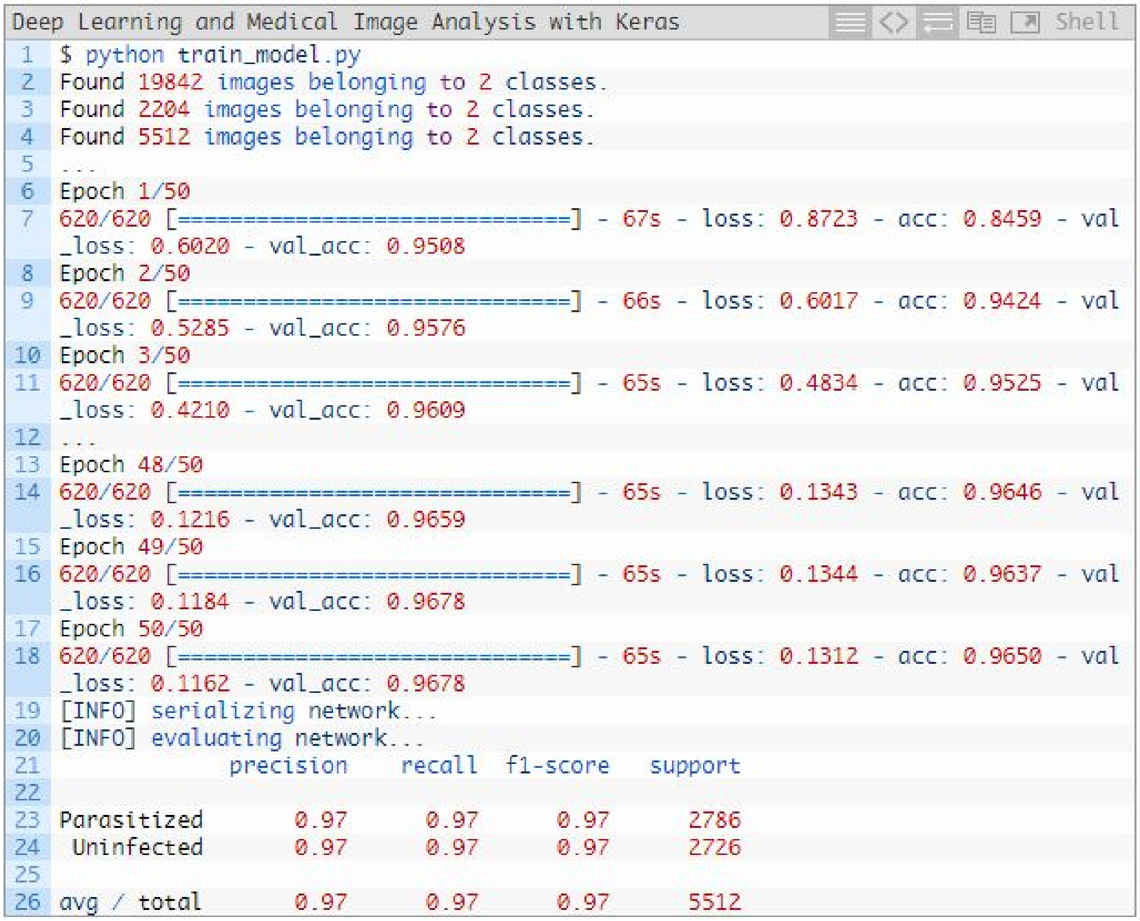
To start, **our model is a complete end-to-end malaria classification system.**
Unlike NIH's approach which leverages a multiple step process of (1) feature extraction from multiple models and (2) classification, we instead can utilize only a single, compact model and obtain comparable results.

Speaking of compactness, our serialized model file is only **17.7MB**. Quantizing the weights in the model themselves would allow us to obtain a model **< 10MB** (or even smaller, depending on the quantization method) with only slight, if any, decreases in accuracy.

Our approach is also faster in two manners.

First, it takes less time to train our model than NIH's approach.Our model took only **6 hour to train** while NIH's model took ~24 hours.
Secondly, our model is faster in terms of both (1) forward-pass inference time and (2) significantly fewer parameters and memory/hardware requirements.

Consider the fact that NIH’s method requires pre-trained networks for feature extraction. Each of these models accepts input images that have input image spatial dimensions in the range of 224×244, 227×227, and 299×299 pixels.



```
$ python train_model.py
Found 19842 images belonging to 2 classes.
Found 2204 images belonging to 2 classes.
Found 5512 images belonging to 2 classes.
...
Epoch 1/50
620/620 [==============================] - 67s - loss: 0.8723 - acc: 0.8459 - val
_loss: 0.6020 - val_acc: 0.9508
Epoch 2/50
620/620 [==============================] - 66s - loss: 0.6017 - acc: 0.9424 - val
_loss: 0.5285 - val_acc: 0.9576
Epoch 3/50
620/620 [==============================] - 65s - loss: 0.4834 - acc: 0.9525 - val
_loss: 0.4210 - val_acc: 0.9609
...
Epoch 48/50
620/620 [==============================] - 65s - loss: 0.1343 - acc: 0.9646 - val
_loss: 0.1216 - val_acc: 0.9659
Epoch 49/50
620/620 [==============================] - 65s - loss: 0.1344 - acc: 0.9637 - val
_loss: 0.1184 - val_acc: 0.9678
Epoch 50/50
620/620 [==============================] - 65s - loss: 0.1312 - acc: 0.9650 - val
_loss: 0.1162 - val_acc: 0.9678
[INFO] serializing network...
[INFO] evaluating network...
             precision    recall  f1-score   support

Parasitized       0.97      0.97      0.97      2786
 Uninfected       0.97      0.97      0.97      2726

avg / total       0.97      0.97      0.97      5512
```

Fig 15 : Output of the Model

Our model requires only 64×64 input images and obtains near identical accuracy.

All that said, we have not performed a full-blown accuracy, sensitivity, and specificity test, but based on our results we can see that we are on the right track to creating an automatic malaria classifier that is not only more accurate but significantly smaller, requiring less processing power as well.

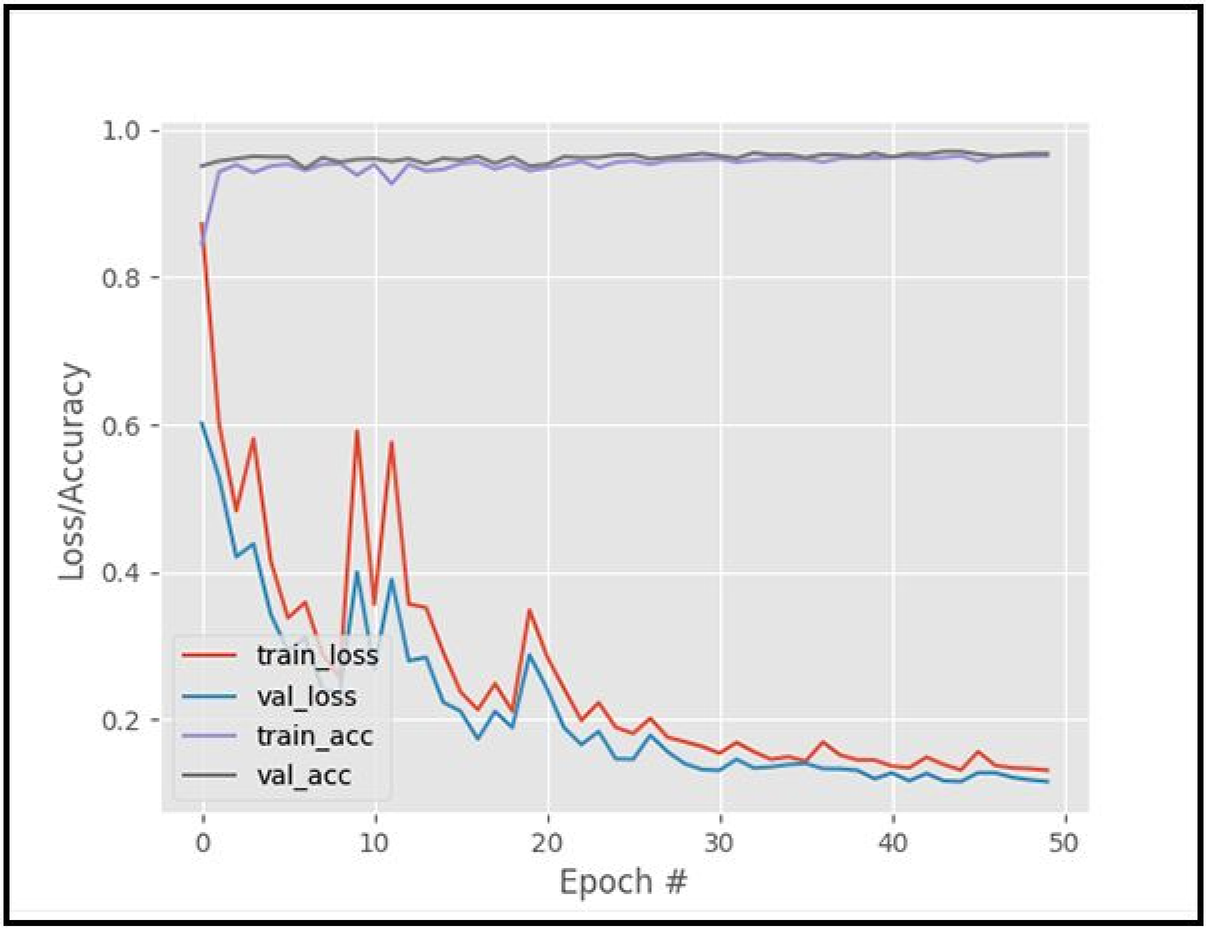


Fig 16 : Plot of the training model for loss and accuracy

Our malaria classifier model training/testing accuracy and loss plot shows that we've achieved high accuracy and low loss.

The output of the image is shown in figure below.As you can see that if the blood cell if affected then it will show the parasitized and if the blood cell is not affected then it will show unaffected.Here we have shown only 4 frames of the images but it can be added if we want more number of images.

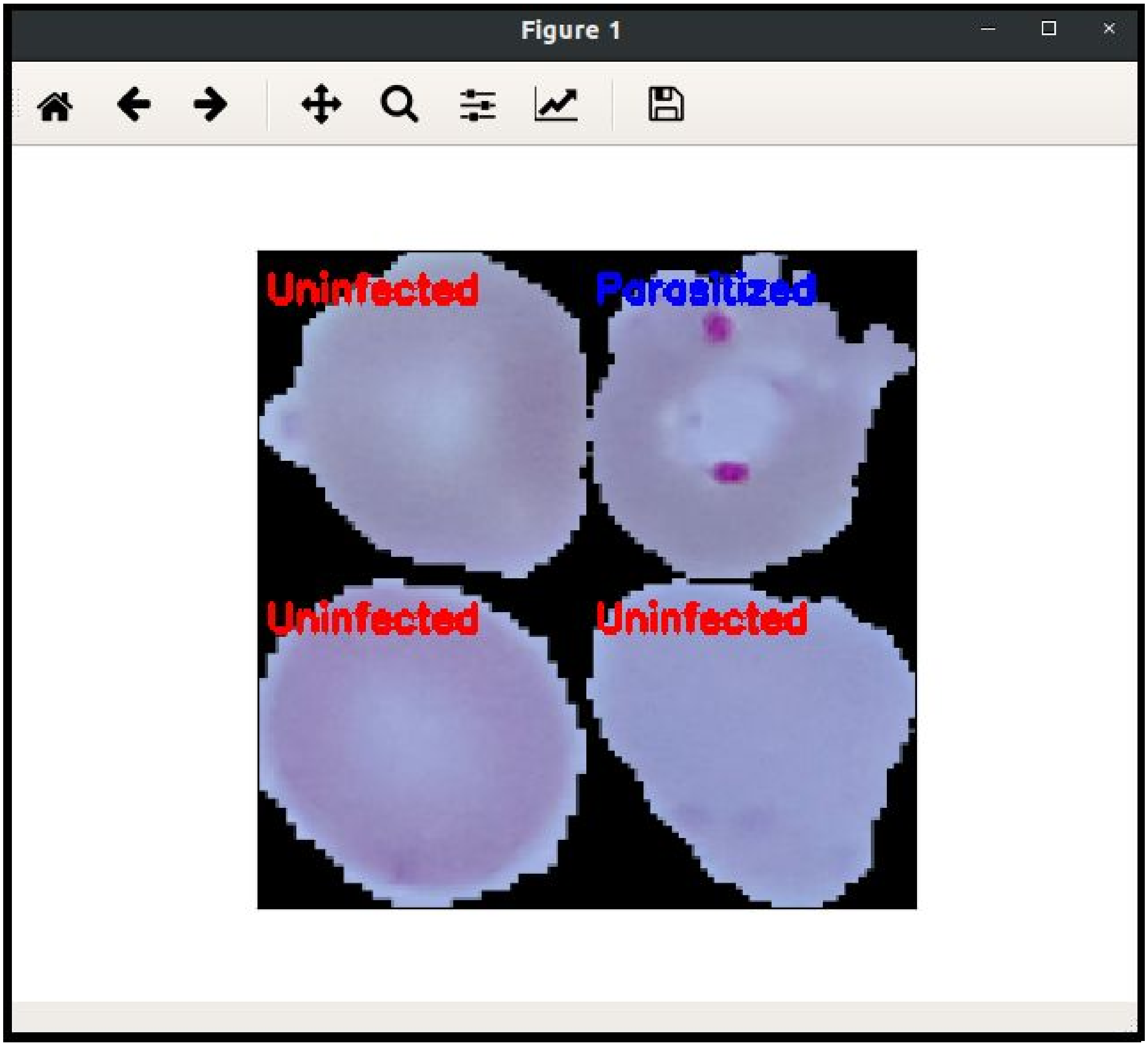


Fig 17 : Output of the Image Uploaded By User

CHAPTER 8 : MAINTENANCE AND SUPPORT

8.1 Corrective Maintenance

- In case of any bugs left in the system, the bugs and issues will be fixed for smooth running of the application.
- The accuracy of the system can be further improved with other algorithms if needed.
- Multiple diseases images can be predicted if sample data set can be available.
- Hybrid system can be implemented.

8.2 Adaptive Maintenance

- The features in the application can be added such as history of the disease can be kept in the log.
- The available images of diseases can also be added for covering more number of diseases.

CHAPTER 9 : CONCLUSION

9.1 Conclusion

This project aims to predict the disease on the basis of the symptoms. The project is designed in such a way that the system takes symptoms from the user as input and produces output i.e. predict disease. Average prediction accuracy probability of 85% to 90% is obtained. Disease Predictor was successfully implemented using flask framework.

9.2 Future Work

- In this system the website is static and on local host.So we will try to make website dynamic and host it.
- Here only Malaria is implemented and can be predicted.In future we will try to implement more diseases that can be predicted.
- We will enhance our system by adding multiple diseases recognization.
- Hybrid algorithm will be implemented so that we can get more accuracy.
- This project has not implemented recommendation of medications to the user. So, medication recommendation can be implemented in the project.
- History about the disease for a user can be kept as a log and recommendation can be implemented for medications.

REFERENCES

[1] Al-Aidaroos, K., Bakar, A., & Othman, Z. (2012). Medical Data Classification With Naive Bayes Approach. Information Technology Journal .

[2] A.Davis, D., V.Chawla, N., Blumm, N., Christakis, N., & Barbasi, A. L. (2008). Predicting Individual Disease Risk Based On Medical History.

[3] Jyoti Soni, Ansari, U., Sharma, D., & Soni, S. (2011). Predictive Data Mining for Medical Diagnosis: An Overview Of Heart Disease Prediction.

[4] Adam, S., & Parveen, A. (2012). Prediction System For Heart Disease Using Naive Bayes.

[5] NisharBanu, MA; Gomathy, B;. (2013). Disease Prediction System Using Data Mining Techniques.

[6] F. Azam and H. F. VanLandingham. An efficient dynamic system identification technique using modular neural networks. Artificial Neural Networks for Intelligent Engineering, 7:225- 230,1997.

[7] Artificial neural networks are changing the world. What are they? https://www.extremetech.com/extreme/215170-artificial-neural-networks-are-changing-the-world-what-are-they . Accessed in Oct 2015.

[8] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, Show and Tell: A Neural Image Caption Generator, 20 Apr 2015

[9] Minsky, M. S. Papert. An Introduction to Computational Geometry. MIT Press, 1969.

[10] Werbos, P.J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, 1975

[11] Yaniv Taigman,Ming Yang, Marc'Aurelio Ranzato, Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification, June 24, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385,2015.

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556,2014.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,pages 1–9,2015.

[15] R. Srivastava, K. Greff and J. Schmidhuber. Training Very Deep Networks. arXiv preprint arXiv:1507.06228v2,2015.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735–1780, Nov. 1997.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Identity Mappings in Deep Residual Networks. arXiv preprint arXiv:1603.05027v3,2016.