

DISEASE PREDICTION AND MEDICAL CHECK UP USING LOGISTIC REGRESSION AND CNN

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Electronics and Communication Engineering

by

B. PUSHPAK VINAY (39130074)

B.SAI (39130071)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

APRIL 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
Jeppiar Nagar, Rajiv Gandhi Salai, Chennai-600119
www.sathyabama.ac.in

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of B. PUSHPAK VINAY (39130074) and B.SAI (39130071) who carried out the project entitled "DISEASE PREDICTION AND MEDICAL CHECK UP USING LOGISTIC REGRESSION AND CNN" under our supervision from November 2022 to April 2023.


Internal Guide

Dr. P. CHITRA M.E., Ph.D.


Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce Examination held on 20/4/23

Internal Examiner


20/4/23


External Examiner

DECLARATION

We B. PUSHPAK VINAY (39130074) and B.SAI (39130071) here by declare that the Project Report entitled “**DISEASE PREDICTION ANDMEDICAL CHECK UP USING LOGISTIC REGRESSION AND CNN**” done by us under the guidance of **Dr. P. CHITRA, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE: 19/04/23

PLACE: CHENNAI

1.



2.



SIGNATURE OF THE CANDIDATE(S)

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D. Dean, School of Electrical and Electronics Engineering** and **Dr. T.RAVI, M.E., Ph.D. Head of the Department, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.P. CHITRA, M.E., Ph.D., ASSOCIATE PROFESSOR, of Department of Electronics and Communication Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the project work.

We wish to express our thanks to all teaching and non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Early diseases prediction plays an important role in improving healthcare quality and can help individuals avoid dangerous health situations before it is too late. The deep learning field has shown greater potential in applications such as disease prediction and drug response prediction. Deep learning techniques have continuously improved the accuracy of medical disease prediction as well as helped shape overall healthcare. Input is obtained in the form of numeric values for diseases such as cancer, diabetes, and cardiovascular conditions. To predict diseases such as Pneumonia, malaria, or kidney diseases, an image is used as an input. Output results are acquired instantly in real-time in the form of a probable disease, its causes, and the aftermath. We will be using Deep Learning algorithms, mainly CNN and DNN, and Image Processing.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	viii
1	INTRODUCTION	1
	1.1 IDEOLOGY	2
	1.2 CNN ON MEDICAL IMAGE CLASSIFICATION	2
2	LITERATURE SURVEY	4
	2.1 EXISTING MODEL	8
	2.2 PROPOSED MODEL	8
3	AIM AND SCOPE OF THE PROJECT	10
	3.1 Aim of the Project	10
	3.2 Software Requirement Specifications	10
	3.3 Software working	10
4	SYSTEM DESIGN	14
	4.1 System Architecture	14
	4.2 Use-Case Diagram	14
	4.3 Activity Diagram	15
	4.4 Sequence Diagram	16
	4.5 Data flow Diagram Level 1	17
	4.6 Data flow Diagram Level 2	18
	4.7 ER Diagram	19

5	IMPLEMENTATION DETAILS	20
5.1	Getting the Input	20
5.2	Cancer, Diabetes Heart disease prediction	21
5.3	CNN for image classification	21
5.4	Logistic Regression for numeric input	22
5.5	End to end diagnosis	24
5.6	GUI and preprocessing module	24
5.7	Implementation of CNN	24
5.8	Unit Testing	28
6	RESULTS AND CONCLUSION	29
	REFERENCES	34
	APPENDIX	
	A. RESEARCH PAPER	

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	SYSTEM ARCHITECTURE	14
4.2	USE-CASE DIAGRAM	15
4.3	ACTIVITY DIAGRAM	16
4.4	SEQUENCE DIAGRAM	17
4.5	DATA FLOW DIAGRAM 1	17
4.6	DATA FLOW DIAGRAM 2	18
4.7	ER DIAGRAM	19
5.1	INPUT VALUES FOR ALL DISEASES	20
5.2	CNN ARCHITECTURE	22
5.3	CONVOLUTION METHODOLOGY	26
5.4	MAXPOOLING	27
5.5	CONVERTING MATRIX INTO LINEAR ARRAY FORM	27
6.1	PARASITIC MALARIA PREDICTED	29
6.2	NORMAL LUNGS PREDICTED	30
6.3	DIABETES - INPUT	30
6.4	DIABETIC NOT PREDICTED	31
6.5	HEART DISEASE INPUT	31
6.6	HEART DISEASE PREDICTED	32
6.7	MALARIA IMAGE INPUT	32
6.8	PNEUMONIA IMAGE INPUT	33

LIST OF ABBREVIATIONS

OCT - Optical Coherence Tomography

CNN - Convolutional Neural Network

SVM - Support Vector Machine

KNN - K Nearest Neighbour

CHAPTER 1

INTRODUCTION

Accurately predicting diseases plays a significant role in public health, especially at the early stage which allows patients to take prevention treatments in time. With the growing volume and availability of electronic health records (EHRs), predictive modeling tasks for disease progression and analysis have obtained increasing interest from researchers. The EHR data are temporally sequenced by patient visits with each visit represented as a set of high dimensional clinical events. Mining EHRs is especially challenging compared to standard data mining tasks, due to their noisy, irregular, and heterogeneous nature. A conventional approach of disease prediction is the one-size-fits-all model. That is, using all available training data to build a global model, and then with this model, predicting the risk of diseases for each patient. The benefit of applying a one-size-fits-all model is that it captures the overall information of the entire training population. However, patients may have different phenotypes, different medical conditions, etc. Using a global model may miss some specific information that is important for individual patients. Thus, building a targeted, patient-specific model for each individual patient is urgent and important for personalized medicine. Recent studies show that personalized models can improve predictive performance over global models. A general framework for personalized prediction contains two stages: (1) measuring the similarity among patients, and (2) building a separate model for each patient using his/her similar cohorts. This framework is motivated by the working process of human doctors, i.e., after reviewing or recalling the diagnosed patients with similar diseases or symptoms, the doctors then carefully make decisions. If doctors can find similar patients, the probability of successfully curing this patient may improve a lot. Many similar learning methods have been proposed on healthcare datasets.

However, these models are developed for handcrafted vector representations such as demographics or average numerical values, without considering the temporal information from different visits. For the longitudinal EHR data, the number of patient

visits varies largely, due to patients' irregular visits and incomplete recordings. The aforementioned learning metrics cannot be directly applied to the longitudinal data, since the historical records of each patient do not naturally form a comparable vector. Therefore, one of the key challenges in measuring patient similarity is to derive an effective representation for each patient without the loss of his/her historical information. Recently, deep learning approaches have been widely adopted and rapidly developed in inpatient representation learning such as autoencoder, recurrent neural networks (RNNs), and convolution neural networks (CNNs). CNN has shown its superior ability in the task of measuring patient similarity. However, one drawback of the traditional CNN architecture is that it could not fully utilize the temporal and contextual information of EHRs for disease prediction. Consequently, simultaneously modeling the temporality and content of EHR data is more challenging.

1.1 IDEOLOGY

To tackle the aforementioned challenges and issues, in this paper, we aim to solve the following key problems in personalized prediction: how to build a model to accurately measure patients' similarities from their historical records, and how to build an accurate personalized prediction model with the learned similarities. To achieve these goals, we first design a novel time-fusion CNN-based framework to account for the temporality across different time intervals. With the proposed framework, we can generate the vector representation for each patient. Based on the learned patient representations, a matching metric is then introduced to obtain a similar representation. Considering the practical meaning, we add a firm symmetric constraint to the framework structure. This similarity learning framework is end-to-end, which learns patient representations and pairwise similarity simultaneously. Since the similarity probability between a pair of patients indicates the risk level of the two patients developing the same disease, we use it as the score to rank the similarity among patients.

1.2 CNN ON MEDICAL IMAGE CLASSIFICATION

With the different CNN-based deep neural networks developed and achieved a significant result on ImageNet Challenger, which is the most significant image classification and segmentation challenge in the image analyzing field. The CNN-

based deep neural system is widely used in the medical classification task. CNN is an excellent feature extractor, therefore utilizing it to classify medical images can avoid complicated and expensive feature engineering. Presented a customized CNN with shallow ConvLayer to classify image patches of lung disease. The authors also found that the system can be generalized to other medical image datasets. Moreover, in other research, it also found that CNN based system can be trained from a big chest X-ray (CXR) film dataset and state-of-art with high accuracy and sensitivity results on their dataset, like Stanford Normal Radiology Diagnostic Dataset containing more than 400,000 CXR and a new CXR database (ChestX-ray8), which consist of 108,948 frontal-view CXR. Moreover, using limited data makes it hard to train an adequate model.

Therefore, the transfer learning of CNN is widely used in medical image classification tasks. Germany et al. use InceptionV3 with ImageNet trained weight and transfer learning on a medical image dataset containing 108,312 optical coherence tomography (OCT) images. They got an average accuracy of 96.6%, with a sensitivity of 97.8% and a specificity of 97.4%. The authors also compared the results with six human experts. Most of the experts got high sensitivity but low specificity, while the CNN-based system got high values on both sensitivity and specificity. Moreover, on the average weight error measure, the CNN-based system exceeds two human experts. The authors also verified their system on a small pneumonia dataset, including about five thousand images, and achieved an average accuracy of 92.8%, with a sensitivity of 93.2% and a specificity of 90.1%. This system finally may help in accelerating diagnosis and referral of patients and therefore introduce early treatment, resulting in an increased cure rate. Moreover, Vianna also studied how to utilize transfer learning to build an X-ray image classification system that is the critical component of a computer-aided diagnosis system. The authors found a fine-tuned transfer learning system with data augmentation effectively alleviates the overfitting problem and yields a better result than two other models: training from scratch and a transfer learning model with only a retrained last classification layer.

CHAPTER 2

LITERATURE REVIEWS

TITLE 1:

A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis, The Lancet Digital Health Volume 1, Issue 6, October 2019

AUTHORS:

Liva Faes Xiaoxuan Liu Pearse A Kean

DESCRIPTION:

Diagnosis of disease using deep learning algorithms holds enormous potential. The accuracy of deep learning algorithms is equivalent to health-care professionals, while acknowledging that more studies considering the integration of such algorithms in real- world settings are needed. The more important finding around methodology and reporting means the credibility and path to impact of such diagnostic algorithms might be undermined by an excessive claim from a poorly designed or inadequately reported study. In this review, we have highlighted key issues of design and reporting that investigators should consider.

TITLE 2 :

Disease Prediction by Machine Learning Over Big Data from Healthcare Communities.

AUTHORS:

Min Chen, Yixue Hao, Kai Hwang, Lu Wang, Lin Wang

DESCRIPTION:

The authors propose a new convolutional neural network (CNN)-based multimodal disease risk prediction algorithm using structured and unstructured data from the hospital. To the best of our knowledge, none of the existing work focused on both data types in the area of medical big data analytics. Compared with several typical prediction algorithms, the prediction accuracy of our proposed algorithm reaches 94.8% with a convergence speed, which is faster than that of the CNN-based unimodal disease risk prediction algorithm.

TITLE 3 :

Diagnosing of disease using machine learning,2021, Machine Learning and the Internet of Medical Things in Healthcare Pages 89-111

AUTHORS:

Pushpa Singh,Narendra Singh,Krishna Kant Singh,Akansha Singh⁴

DESCRIPTION:

Disease diagnosis using machine-learning techniques like algorithms like SVM, KNN, Naive Bayes, and Decision tree can enhance the quickness of decision-making, and it can reduce the rate of false positives. The practical implementation of these algorithms is discussed with Python. These algorithms are used to diagnose various diseases like cancer, diabetes, epilepsy, heart attack, and other prominent diseases.

TITLE 4 :

An overview of deep learning in medical imaging focusing on MRI, May 2019, Zeitschrift für Medizinische Physik Volume 29, Issue 2, Pages 102-127

AUTHORS:

Alexander Selvikvåg lundervold Arvid Lundervold

DESCRIPTION:

Deep neural networks are now the state-of-the-art machine learning models across a variety of areas, from image analysis to natural language processing, and widely deployed in academia and industry. These developments have a huge potential for medical imaging technology, medical data analysis, medical diagnostics and healthcare in general, slowly being realized.

TITLE 5 :

Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things Platform, May 2020, IEEE Access (Volume: 8)

AUTHORS:

Yuanyuan Pan; Minghuan Fu; Biao Cheng; Xuefei Tao; Jing Guo

DESCRIPTION:

The Enhanced Deep learning assisted Convolutional Neural Network (CNN) has been proposed to assist and improve patient prognostics of heart disease. The EDCNN model is focused on a deeper architecture which covers the multilayer perceptron's model with regularization learning approaches. The EDCNN system has been implemented on the Internet of Medical Things Platform (IoMT) for decision support systems which helps doctors to effectively diagnose heart patient's information in cloud platforms anywhere in the world.

TITLE 6 :

Designing Disease Prediction Model Using Machine Learning Approach, 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) **AUTHORS:**

hiraj Dahiwade; Gajanan Patle; Ektaa Meshram

DESCRIPTION:

For disease prediction, we use K-Nearest Neighbor (KNN) and Convolutional neural network (CNN) machine learning algorithms for accurate prediction of disease. For disease prediction required disease symptoms dataset. In this general disease prediction the living habits of a person and checkup information are considered for the accurate prediction. The accuracy of general disease prediction by using CNN is 84.5% which is more than KNN algorithm. And the time and the memory requirement are also more in KNN than CNN.

TITLE 7 :

Healthcare Techniques Through Deep Learning:Issues, Challenges and Opportunities

AUTHORS:

Durr-e-maknoon Nisar Meshrif Alruily Sultan H. Almotir Rashid Amin

DESCRIPTION:

Tools and algorithms are the major parts of any system. In deep learning, tools are used to convert data into actionable info. The deep learning models are categorized according to supervised and unsupervised learning techniques. Due to these algorithms' significant importance, medical devices should be built over time based on

these models, such as the microscope, phonendoscope, and electrocardiogram, accommodating physicians' perceptive capability limitations. In processing and forecasting from large datasets, deep learning is playing a critical role

TITLE 8 :

A Survey on Computer Vision for Assistive Medical Diagnosis From Faces, SEPTEMBER 2018, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, VOL. 22, NO. 5,

AUTHORS:

Jerôme Thevenot^{*}, Miguel Bordallo Lopez^{*}, and Abdenour Hadid

DESCRIPTION:

a range of distant and non-invasive imaging solutions has been described, providing insights into the suitability of each one of the techniques in the assessment of different types of symptoms related to medical conditions. The findings show that utilizing computer vision methods, over 30 conditions can be preliminary diagnosed from the automatic detection of some of their symptoms

TITLE 9 :

Early-Stage Risk Prediction of Non-Communicable Disease Using Machine Learning in Health CPS July 2021, IEEE Special Section On AI And IoT Convergence For Smart Health

AUTHORS:

Rahatara Ferdousi, M. Anwar Hossain, Abdulmotaleb El Saddik

DESCRIPTION:

we propose a novel machine learning based health CPS framework that addresses the challenge of effectively processing the wearable IoT sensor data for early risk prediction of diabetes as an example of NCDs. In the experiment, a verified diabetic dataset has been used for training, while the testing has been performed on an artificially generated data collection from sensors

TITLE 10:

Medical image analysis based on deep learning approach, 2021, Multimedia Tools and Applications volume 80, pages24365–24398 (2021)

AUTHORS:

Muralikrishna Puttagunta & S. Ravi

DESCRIPTION:

Discussed are new algorithms and strategies in the area of deep learning. DLA can support clinical decisions for next-generation radiologists. DLA can automate radiologist workflow and facilitate decision-making for inexperienced radiologists. DLA is intended to aid physicians by automatically identifying and classifying lesions to provide a more precise diagnosis. DLA can help physicians to minimize medical errors and increase medical efficiency in the processing of medical image analysis.

2.1 EXISTING MODEL

Disease prediction systems have been playing a significant role in the life of people, and it has been considered an important topic by many academics. Although prediction results achieved are promising, these traditional approaches are still far from being highly accurate and efficient. The existing systems are simple and effective but are extremely vulnerable to impact. Moreover, state-of-the-art methods can detect diseases pertaining to certain organs only while some severe conditions may go completely undetected. This could lead practitioners to false assumptions and improper diagnosis and treatments provided to patients.

2.2 PROPOSED SYSTEM

We propose an end-to-end application that predicts Cancer, Diabetes, Heart disease using distinct techniques. The aim of developing this application with the help of Deep Learning algorithms is to immensely help to solve health-related issues by assisting the physicians to predict and diagnose diseases at an early stage. The accurate analysis by our proposed application benefits in early disease prediction, patient care, and community services. The proposed application is simple and it shows the good efficient performance. The overall accuracy of the proposed scheme has been evaluated with the traditional state-of-the-art models and the results from our proposed application show a higher accuracy rate. The proposed prediction model caters to this objective following a stepwise approach through cleaning,

feature extraction, and classification. Initially disease dataset is taken as an input for the system for Diabetes, heart disease, and cancer datasets. Relevant elements are fed to the neural network by eliminating the irrelevant features using feature selection. As the algorithms are selected, the symptoms are processed, and the most likely disease is given as an output. The performance of the network is dependent on the number of guidelines used which decide the behavior of the network. The input is given to the neurons which perform some operation to generate the output.

CHAPTER 3

AIM AND SCOPE OF THE PROJECT

3.1 AIM OF THE PROJECT

The proposed prediction model caters to this objective following a stepwise system for diabetes, heart disease, and cancer datasets. Relevant elements are fed to the neural network by eliminating the irrelevant features using feature selection as the algorithms is selected, the symptoms are processed.

3.2 SOFTWARE REQUIREMENT SPECIFICATION

Operating system	: Windows7 (with service pack 1), 8, 8.1,10
Language	: Python

3.3 SOFTWARE WORKING

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field. Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open- source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate

on projects without worrying about any compatibility issues. It gives you a seamless environment that simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let's discuss some fundamental commands you can use to start using this package manager.

Listing All Environments

To begin using Anaconda, you'd need to see how many Conda environments are present in your machine.

```
conda env list
```

It will list all the available Conda environments in your machine. **Creating a New Environment**

You can create a new Conda environment by going to the required directory and use this command:

```
conda create -n <your_environment_name>
```

You can replace <your_environment_name> with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y:

```
proceed ([y])/n)?
```

On the other hand, if you want to create an environment with a particular version of Python, you should use the following command:

```
conda create -n <your_environment_name> python=3.6
```

Similarly, if you want to create an environment with a particular package, you can use the following command:

```
conda create -n <your_environment_name> pack_name
```

Here, you can replace pack_name with the name of the package you want to use.

If you have a .yml file, you can use the following command to create a new Conda environment based on that file:

```
conda env create -n <your_environment_name> -f <file_name>.yml
```

We have also discussed how you can export an existing Conda environment to a .yml file later in this article.

Activating an Environment

You can activate a Conda environment by using the following command:

```
conda activate <environment_name>
```

You should activate the environment before you start working on the same. Also, replace the term <environment_name> with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command:

```
conda deactivate
```

Installing Packages in an Environment

Now that you have an activated environment, you can install packages into it by using the following command:

```
conda install <pack_name>
```

Replace the term <pack_name> with the name of the package you want to install in your Conda environment while using this command.

Updating Packages in an Environment

If you want to update the packages present in a particular Conda environment, you should use the following command:

```
conda update
```

The above command will update all the packages present in the environment. However, if you want to update a package to a certain version, you will need to use the following command:

```
conda install <package_name>=<version>
```

Exporting an Environment Configuration

Suppose you want to share your project with someone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can create an environment configuration .yml file and share it with that person. Now, they can create an environment like your one by using the .yml file.

For exporting the environment to the .yml file, you'll first have to activate the same and run the following command:

```
conda env export ><file_name>.yml
```

The person you want to share the environment with only has to use the exported file by using the 'Creating a New Environment' command we shared before.

Removing a Package from an Environment

If you want to uninstall a package from a specific Conda environment, use the following command:

```
conda remove -n <env_name><package_name>
```

On the other hand, if you want to uninstall a package from an activated environment, you'd have to use the following command:

```
conda remove <package_name>
```

Deleting an Environment

Sometimes, you don't need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment, which you can do so by using the following command:

```
conda env remove --name <env_name>
```

The above command would delete the Conda environment right away.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

This diagram is nothing but a simple description of all the entities that have been incorporated into the system. The diagram represents the relations between each of them and involves a sequence of decision-making processes and steps. You can simply call it a visual or the whole process and its implementation. All functional correspondences are explained in this diagram.

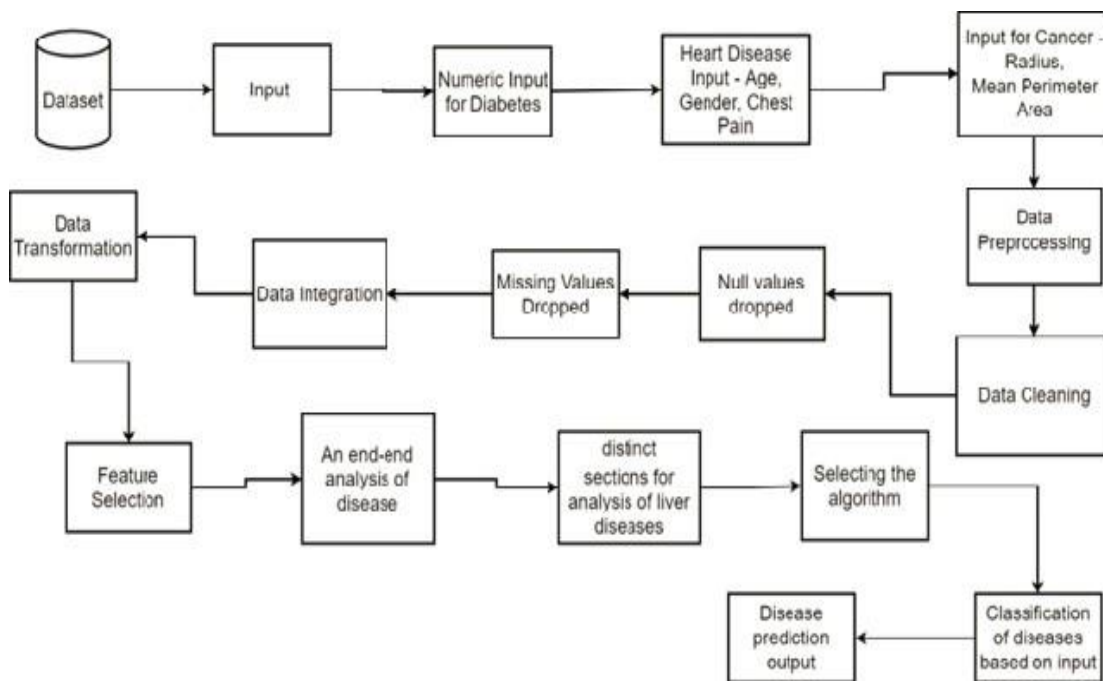


Fig: 4.1: Architecture Diagram

4.2 USE-CASE DIAGRAM

A use case diagram is a standard diagram that shows all interactions between the user, dataset, and algorithm used. It is developed in the early stages of the process.

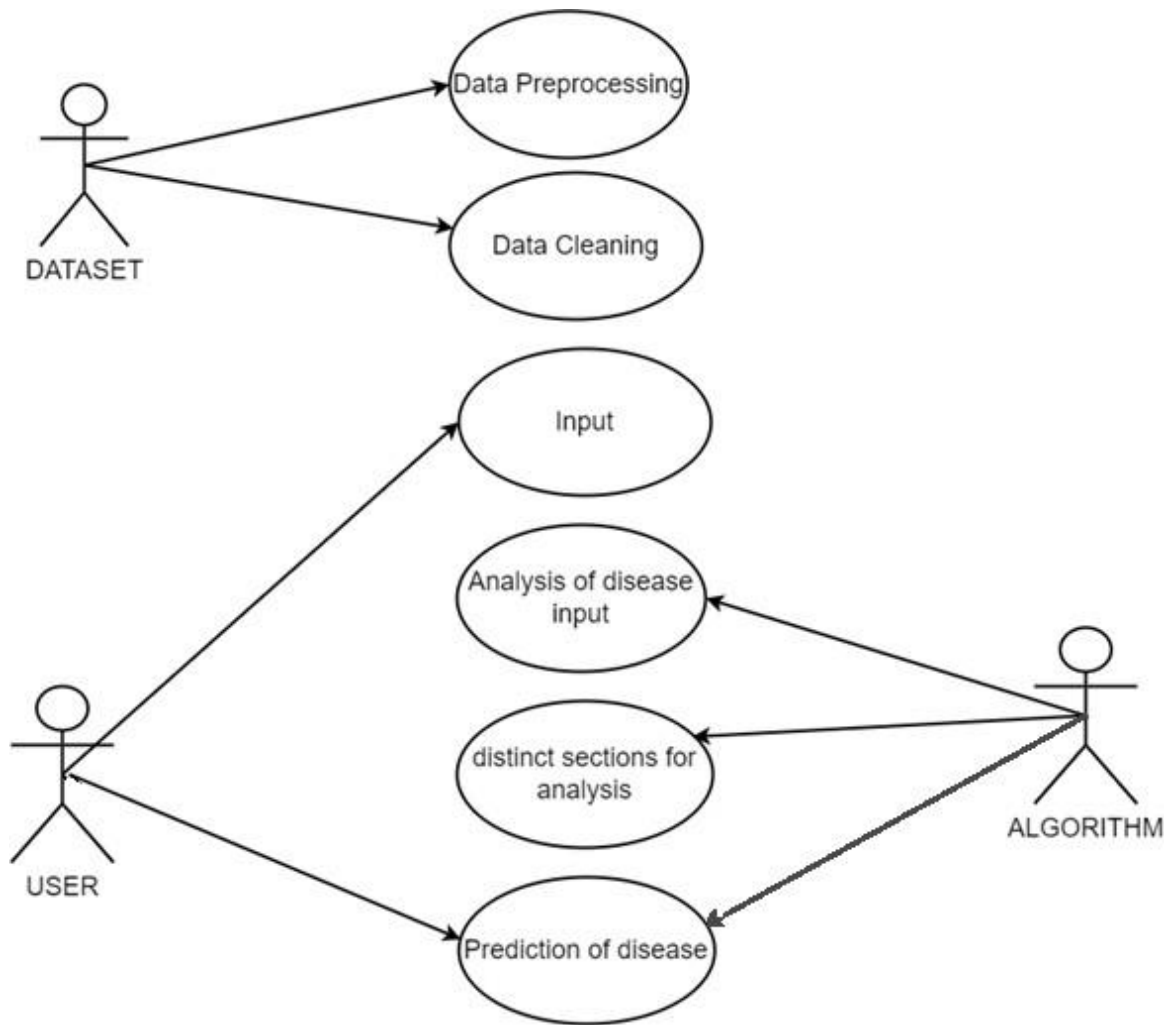


Fig: 4.2: Use-Case Diagram

4.3 ACTIVITY DIAGRAM

In simple terms, a diagram that represents the order of all activities is called the activity diagram. It shows the workflow between different activities that take place in the whole process. However, these are not exactly flowcharts but are similar.

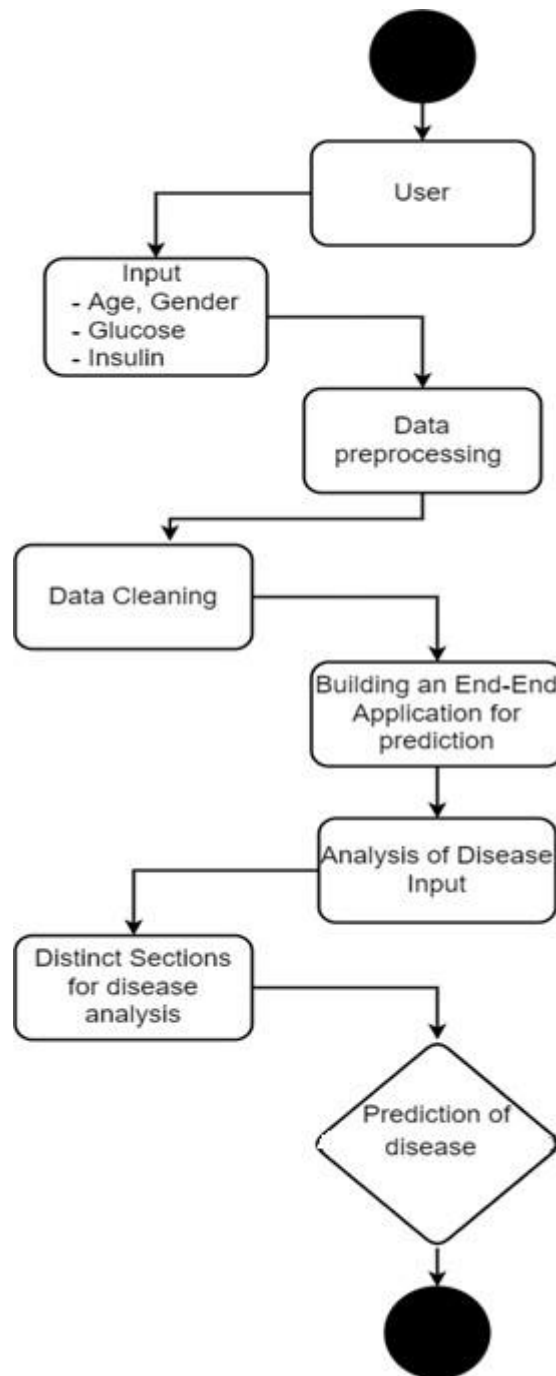


Fig: 4.3: Activity Diagram

4.4 SEQUENCE DIAGRAM

These are other kinds of interaction-based diagrams that show how all the operations are carried out. They capture the context of collaborations between objects and processes.

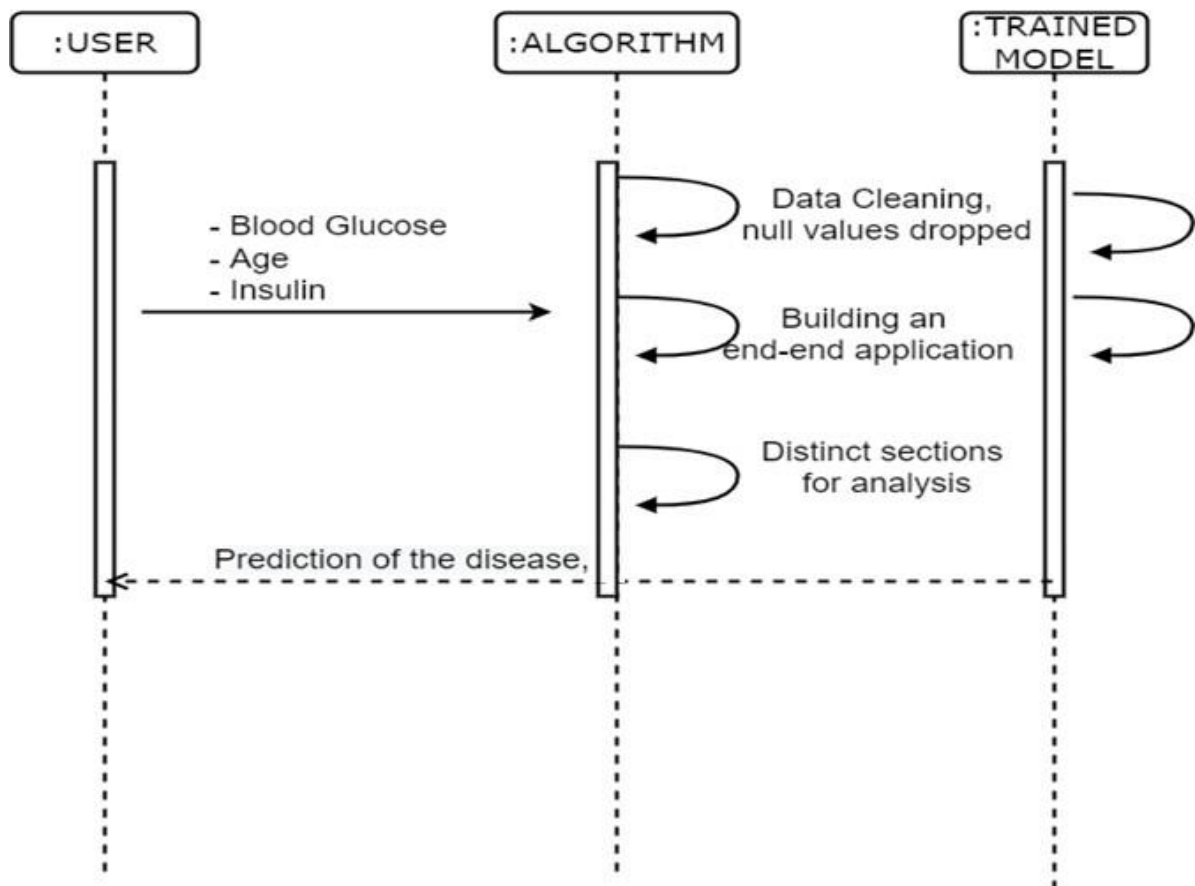


Fig: 4.4: Sequence Diagram

4.5 DATA FLOW DIAGRAM LEVEL 1

This is basically a contextual diagram, also referred to as a “context diagram”. It only represents the top level or the 0 Level in the whole process. It gives an abstraction kind of view and shows the whole system as a single process and its relationship to externalities.

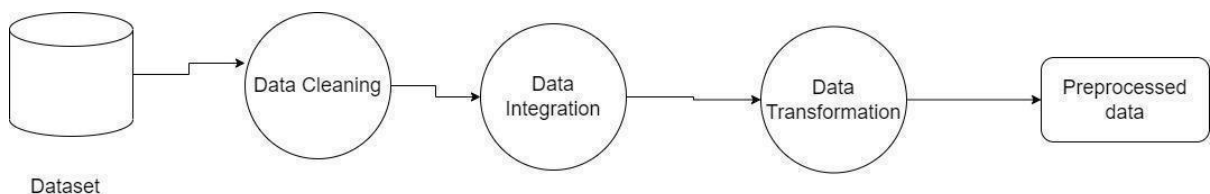


Fig: 4.5: Data Flow Diagram 1

4.6 DATA FLOW DIAGRAM LEVEL 2

Level DFDs represent the complete system as a single process. it rotates every process and subprocess that comes together in a sequence to form the complete system. This along with and 2-level data flow diagrams comprise the “fundamental system model”.

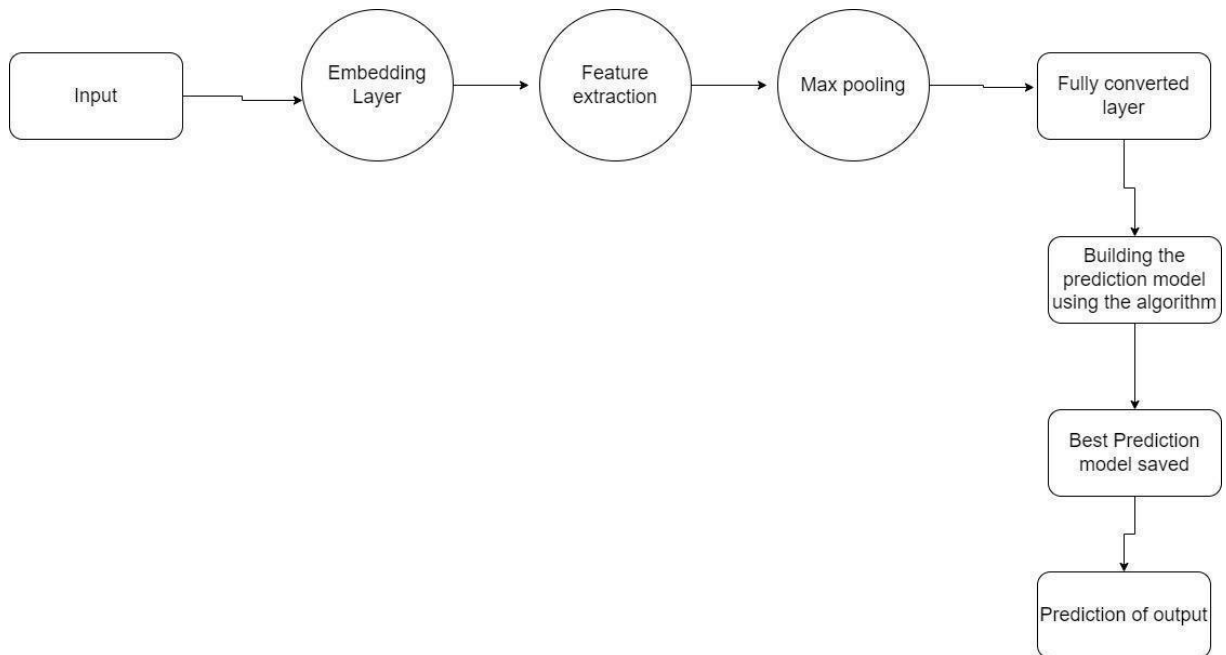


Fig: 4.6: Data Flow Diagram 2

4.7 ER DIAGRAM

ER stands for Entity Relationship. These diagrams display the relationship of entities that are used and stored in the database. They explain the structure of the whole process. These diagrams can be made using three basic concepts, attributes, relationships, and entities.

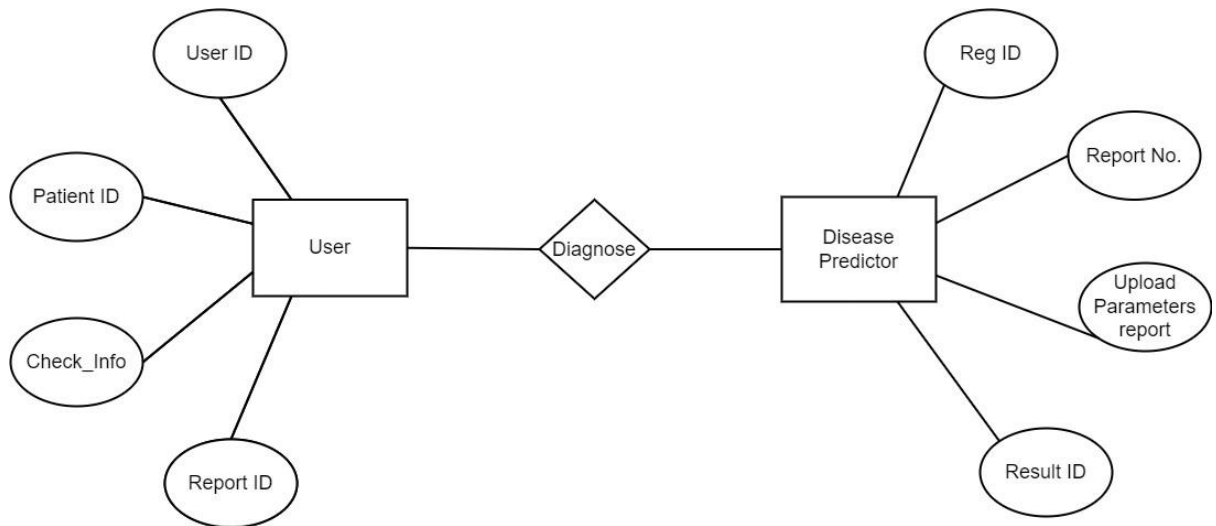


Fig: 4.7: ER Diagram

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 GETTING THE INPUT

Prediction of cancer disease via perimeters such as radius mean perimeter area
prediction of Diabetes via blood glucose, thickness (numeric values)
Prediction of Heart disease via - chest pain type, age, gender (numeric)
To predict Malaria and Pneumonia we get the Image as input.

Prediction of liver diseases - Input Age

Gender Alkalinity

Phosphate balance

Prediction of Kidney Diseases - Input

Input is obtained by fetching machine-oriented data from the dataset that is used to analyze the condition of the kidneys.

Prediction of Pneumonia – Input

Image is used as an input; image processing is leveraged. Prediction of Malaria – Input

Image is used as an input; image processing is leveraged.

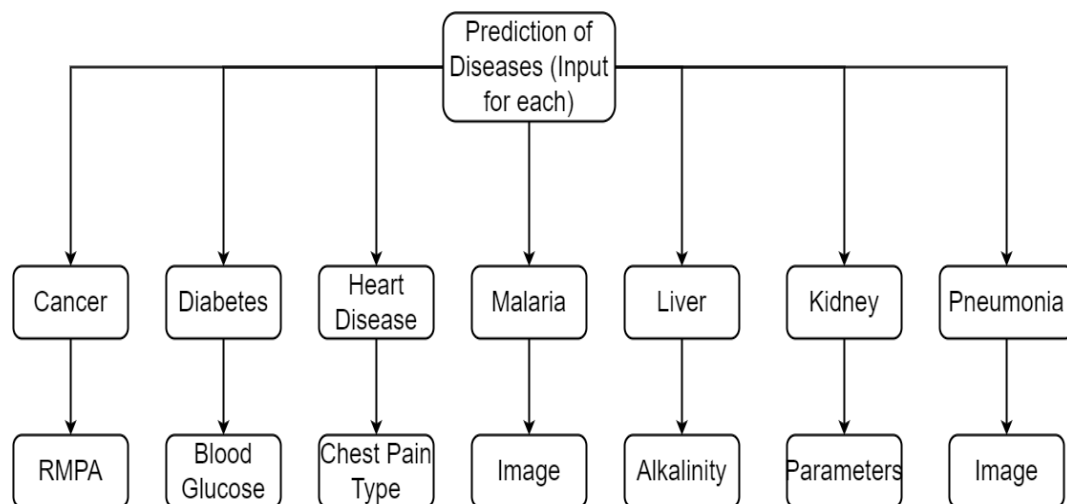


Fig: 5.1: Input values for all diseases

Initially, the disease dataset is taken as an input for the system for Diabetes, heart disease, and cancer datasets. Relevant elements are fed to the neural network by eliminating the irrelevant features using feature selection. The input is given to

the network which performs some operations to generate the output. Inputs for various disease.

5.2 CANCER, DIABETES HEART DISEASE PREDICTION

The diagnosis of heart disease depends on the detailed and precise analysis of the patient's clinical test data and an individual's health history. Here, we're using the Deep Neural Network, DNN algorithm to diagnose heart diseases, diabetes, and cancer-related ailments. The DNN with multiple hidden layers is used hence the proposed model has high performance than ANN.

Input for Diabetes (Numeric Values) Glucose: the concentration test in glucose
Skin thickness: the thickness of the skin in the triceps fold (mm) Insulin: insulin serum for 2-h (μ U/ml)

Input for Heart Diseases

Age: current age of the patient Gender

Chest Pain Type

Input For Cancer-related Ailments Radius

Mean Perimeter Area

5.3 CNN FOR IMAGE CLASSIFICATION

In deep learning, a convolutional neural network (CNN) is a class of deep neural networks, most commonly applied to analyze visual imagery. It is a type of artificial neural network used primarily for image recognition and processing, due to its ability to recognize patterns in images. While CNNs are designed to solve problems with visual imagery, they also have many applications outside of image recognition and analysis, including image classification, natural language processing, drug discovery, and health risk assessments. CNNs also help provide depth estimation for self-driving cars. A CNN is a powerful tool but requires millions of labeled data points for training. Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and output an activation value. The features are converted to a new representation called word embedding by the layer called Embedding Layer. The Embedding layer's output is given to the CNN layer for feature extraction. In

this model, max-pooling has been used. In addition, the batch normalization layer has been used as it involved the training of a very deep neural network. So, the technique adjusts the scaling and activation to normalize the input layer and speed up the learning procedure between hidden units. Towards the last stage of the CNN used in the study, there is a flattening layer to convert the output of convolutional layers into a single-dimensional feature vector. In a fully connected layer, each neuron of the previous layer is directly connected to each of the neurons in its next layer. The fully connected network with dense layers finally drives the classification decision

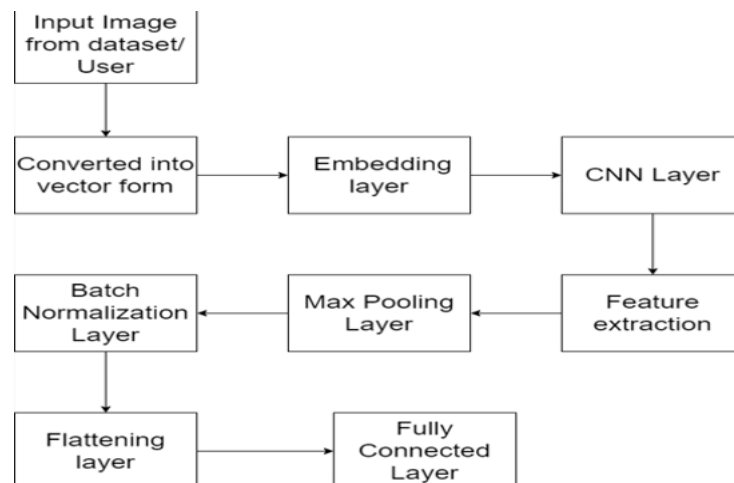


Fig: 5.2: CNN Architecture

5.4 LOGISTIC REGRESSION FOR NUMERIC INPUT

Logistic regression is an algorithm used to predict the probability of categorical dependent variables. The dataset imported will be divided into two parts, namely training data and testing data. Training data is used as a basis for building models. This type of analysis is used to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation. In this way, you can predict how likely it is that an event will happen or a choice will be made. In order to predict the output of a categorical dependent variable, logistic regression must first determine the category or class of the dependent variable. This can be accomplished through a simple calculation, which returns an estimate of how likely it is that any given value within that category(s) will occur. The probabilistic values returned by this equation are then used to generate a prediction for how different combinations of independent variables would produce this particular output. Meanwhile, testing data is used as a

basis for testing or validating the model. The data should contain different variables like age, gender, sex, chest pain, glucose levels, insulin levels, etc. In logistic regression, the dependent variable is always binary. Prediction using the logistic regression method will produce several data that can be used as a basis for concluding to make predictions. We have used a sigmoid function which helps in the graphical representation of the classified data. By using logistic regression, the accuracy is increased.

5.5 END TO END DIAGNOSIS

For the analysis of all these diseases, we will create an end-to-end application that is able to analyze all of the input data for the diseases, process the data using distinct techniques, and then accurately predict the most likely disease which is given as output. The application will contain distinct sections for analysis of liver diseases, cardiovascular diseases, kidney-related ailments, cancer, pneumonia, and malaria. Not only just the disease, but the application would also predict the most likely disease based on the given input data. The application will be designed to predict the disease mention the aftermath of the same, how can it be treated and also the factors or changes that caused the disease. The application will act as a virtual assistant in major clinical laboratories, healthcare centers and medical clinics. The proposed model is a dynamic web application for an end-to-end disease prediction and diagnosis.

The application has been designed using the Flask interface that would effectively display the output and brief out the symptoms. The complete check-up would be performed using all the parameters and images as a data set. In future, with a little improvement, it would fetch data from the cloud and can be retrieved whenever needed. Our proposed methodology helps to improve the accuracy of diagnosis and is greatly helpful for further treatment. This work will be useful for identifying the patients who suffer from heart disease. When a patient is predicted with a positive result their reports and data can be closely analyzed.

5.6 GUI & PREPROCESSING MODULE

This is the interface where the user or patient interacts with the application. This UI is built in a simple way so that it will be easy for the users to interact with.. In this module, these input parameters are trained well and a model is created. These input parameters have to be trained well in order to obtain a high accuracy. The features in the training data and the quality of labeled training data will determine how accurately the machine learns to identify the outcome. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset, and so the data preprocessing module is required to perform tasks such as cleaning the data and making it suitable for a machine learning model which helps in increasing the accuracy and efficiency of a machine learning model. To accurately predict the case using ML we need our data to be cleaned and in a formatted way. In data processing step we also divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Data preprocessing comprises of 4 steps.

1. The first step is the data cleaning in which the duplicate, incorrectly formatted, corrupted data will be fixed or removed.
2. The second step is the data integration in which it combines multiple sources data into single view.
3. The third step is the data reduction step in which the data are encoded, scaled sorted if needed.
4. The final step is the data transformation in which the data is transformed into a required format

5.7 IMPLEMENTATION of CNN

CNN Classification involves 4 steps namely Convolution layer, Activation function, max Pooling, fully connected network.

Relu layer: Rectified linear unit is used to scale the parameters to non-negative values. We get pixel values as negative values too. In this layer we make them as 0's. The purpose of applying the rectifier function is to increase the non-linearity in our images. The reason we want to do that is that images are naturally non-linear.

The rectifier serves to break up the linearity even further in order to make up for the linearity that we might impose on an image when we put it through the convolution operation. What the rectifier function does to an image like this is remove all the black elements from it, keeping only those carrying a positive value (the grey and white colors). The essential difference between the non-rectified version of the image and the rectified one is the progression of colors. After we rectify the image, you will find the colors changing more abruptly. The gradual change is no longer there. That indicates that the linearity has been disposed of.

Pooling layer: The pooling (POOL) layer reduces the height and width of the input. It helps reduce computation, as well as helps make feature detectors more invariant to its position in the input. This process is what provides the convolutional neural network with the “spatial variance” capability. In addition to that, pooling serves to minimize the size of the images as well as the number of parameters which, in turn, prevents an issue of “overfitting” from coming up. Overfitting in a nutshell is when you create an excessively complex model in order to account for the idiosyncracies we just mentioned. The result of using a pooling layer and creating down sampled or pooled feature maps is a summarized version of the features detected in the input. They are useful as small changes in the location of the feature in the input detected by the convolutional layer will result in a pooled feature map with the feature in the same location. This capability added by pooling is called the model’s invariance to local translation.

Fully Connected layer: The role of the artificial neural network is to take this data and combine the features into a wider variety of attributes that make the convolutional network more capable of classifying images, which is the whole purpose from creating a convolutional neural network. It has neurons linked to each other, and activates if it identifies patterns and sends signals to output layer. The output layer gives output class based on weight values.

In Convolution layer a 3×3 kernel is applied on the input matrix with stride 1, so that the size of the image is reduced. The obtained matrix is sent to the activation function

layer, here ReLU (Rectified Linear Unit) activation function is used to eliminate negative values. The next step is max pooling, here the rectified map goes through a pooling layer, pooling is down sampling operation that reduces the dimensionality of feature map. The output of this layer is a matrix with reduced size containing all the features. These 3 steps are repeated for two times, in which 32 filters are used each time. The next step is Flattering in which we convert the 2D array from pooling into a long continuous linear vector to which a filter of size 64 is applied followed by activation function. Later a Drop Out of size 0.5 is applied to prevent over-fitting. In the next step ANN model is applied, in which sigmoid activation is used function to detect the presence of stone.

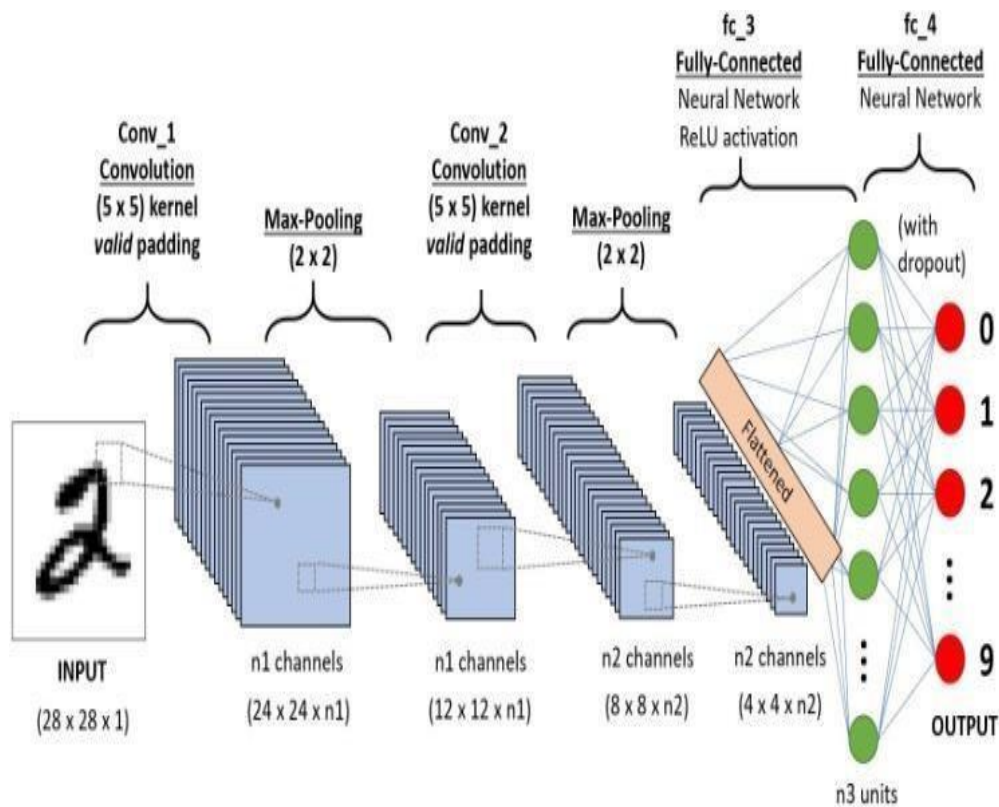


Fig: 5.3: Convolution methodology

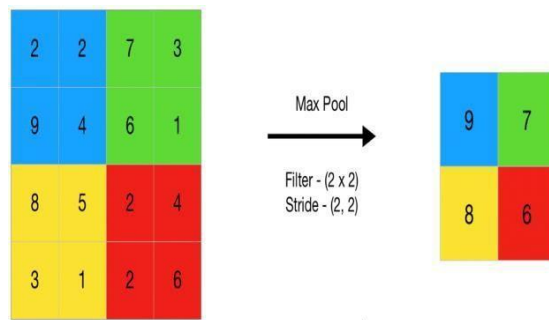


Fig: 5.4: Maxpooling

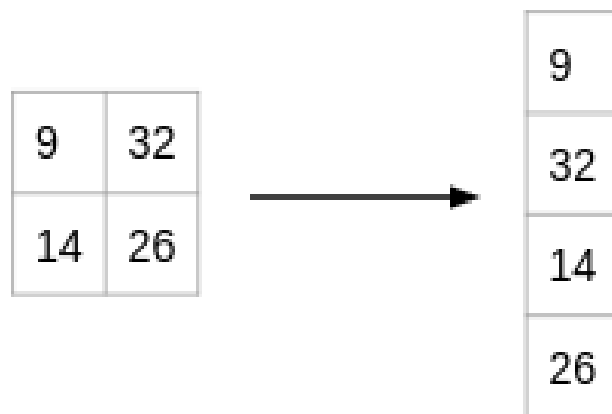


Fig: 5.5: Converting matrix into linear array form

Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

5.8 Unit Testing

A unit is the smallest possible testable software component. A unit can be function or procedure implemented in a procedural programming languages. A unit may also be a small-sized COTS component purchased from an outside vendor that is undergoing evaluation by the purchaser, or a simple module retrieved from an in-house reuse library. Unit test results are recorded for future testing process. This result document used for integration and system tests.

Some of the phases for unit test planning are;

- ❖ Describe Unit Test Approach and Risks.
- ❖ Identify Unit features to be tested.
- ❖ Add levels of detail to the test plan.

TEST NO.	INPUT	EXPECTED BEHAVIOUR	TESTING	STATUS
1	User uploading proper input values	Data updated to the trained model	Unit Testing	Pass
2	User uploading invalid input data	Data not updated to the trained model	Unit Testing	Fail
3	User uploading proper image	Disease predicted	Output Testing	Pass
4	User uploading invalid image	Disease prediction model not loaded	Output Testing	Fail

CHAPTER 6

RESULTS AND CONCLUSIONS

The results thus justify the effectiveness of the proposed method in disease prediction. Although the results are promising, the datasets being used are relatively smaller in comparison to the big data culture predominant in the present day and age. Thus, the research's future path will be to test the proposed model on large-high-quality datasets to check the effectiveness and reliability of the proposed model. Our proposed methodology helps to improve the accuracy of diagnosis and is greatly helpful for further treatment. This work will be useful for identifying the patients who suffer from heart disease. When a patient is predicted with a positive result their reports and data can be closely analyzed. In future enhancements, the accuracy has to be tested with different datasets and to apply other AI algorithms to check the accuracy estimation

This is the image upload UI for malaria disease prediction system. The given image is predicted as Parasitic and the accuracy is found to be 100%.

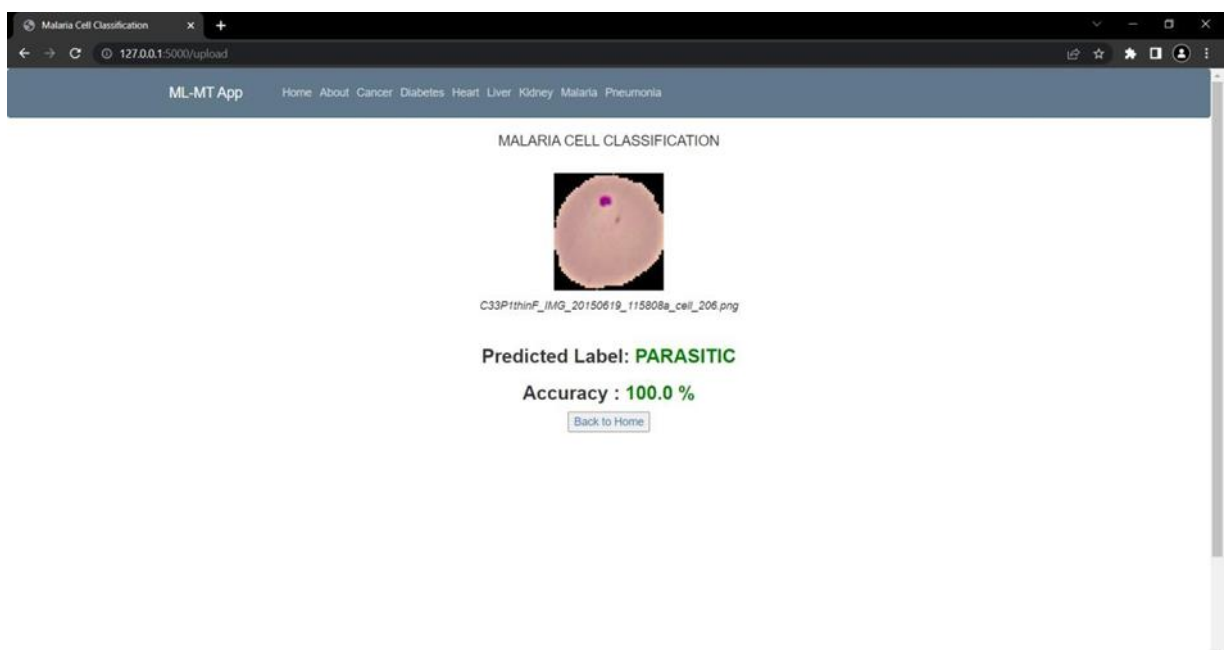


Fig: 6.1: Parasitic image

This is the image upload UI for Lung disease system. The given image is predicted as normal so the lungs is not affected with any disease and the accuracy is found to be 90%.

This is the image upload UI for Lung disease system. The given image is predicted as normal so the lungs is not affected with any disease and the accuracy is found to be 90%.

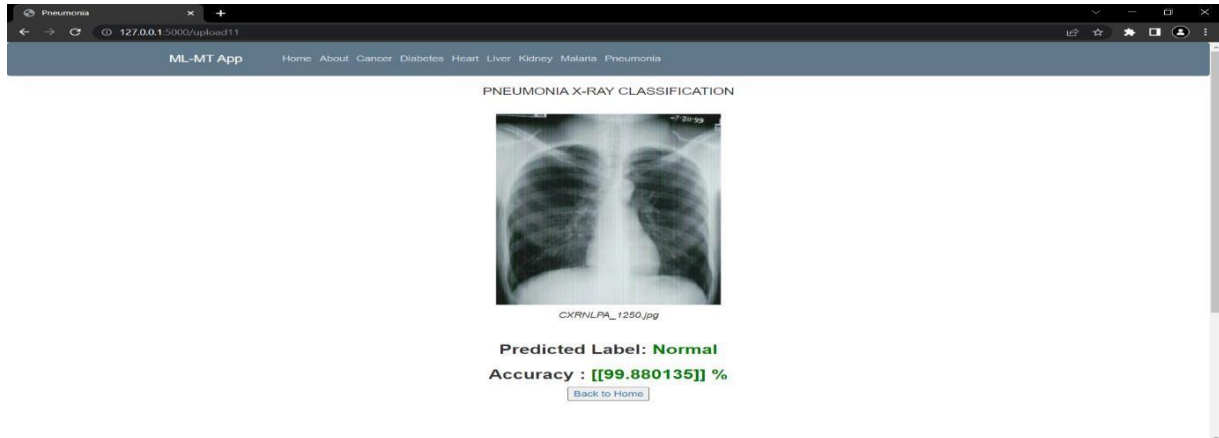


Fig: 6.2: Normal lungs Predicted

This is the input UI for diabetes prediction system. Each field has certain value and these values are passed to the trained ML model.

The screenshot shows a web browser window with the URL 127.0.0.1:5000/diabetes. The page title is "Details Diabetes: Welcome". It prompts the user to "Enter the Following parameters :". The form includes input fields for "Pregnancies" (0), "Glucose" (95), "BloodPressure" (80), "SkinThickness" (60), "Insulin" (100), "BMI" (25), "DiabetesPedigreeFunction" (0.5), and "Age" (35). A "Submit" button is located at the bottom of the form. The browser's address bar and navigation tabs are visible at the top.

Fig: 6.3: Diabetes – input

This is result page for Diabetes disease system. Upon submitting the data, the Trained ML model predicts the output. Here there is no disease predicted for the given input values

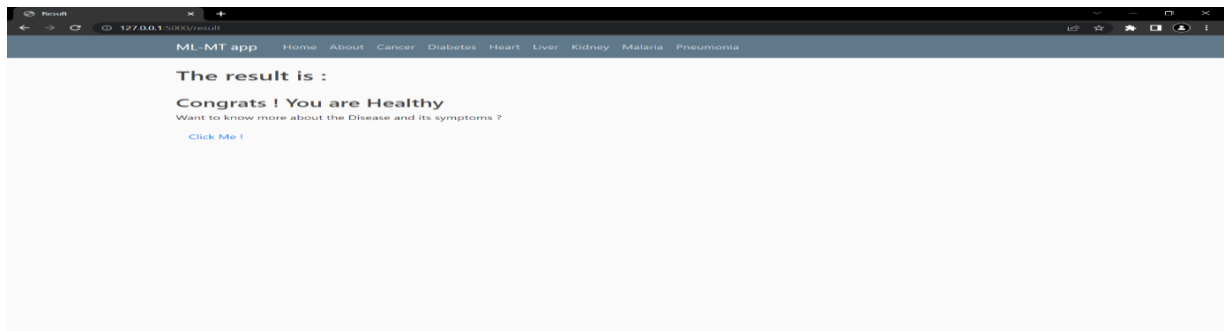


Fig: 6.4: Diabetes predicted

This is the input UI for Heart Disease prediction system. Each field has certain value and these values are passed to the trained ML model.

A screenshot of a web browser displaying the 'ML-MT App' interface for heart disease prediction. The browser's address bar shows '127.0.0.1:5000/heart'. The app's navigation bar includes links for Home, About, Cancer, Diabetes, Heart, Liver, Kidney, Malaria, and Pneumonia. The main content area displays the text 'Details Heart: Welcome' followed by 'Enter the Following parameters :'. Below this, there are ten input fields for the following parameters: Age, Sex(1:Male,0:Female), Chest pain type, Trestbps, Serum cholestoral in mg/dl, Restecg, Thalach, Exang, Oldpeak, Slope, and Thal. A 'Submit' button is located at the bottom of the form.

Fig: 6.5: Heart Disease Input

This is result page for Heart disease prediction system. Upon submitting the data, the Trained ML model predicts the output. Here the heart disease is predicted for the given input values.

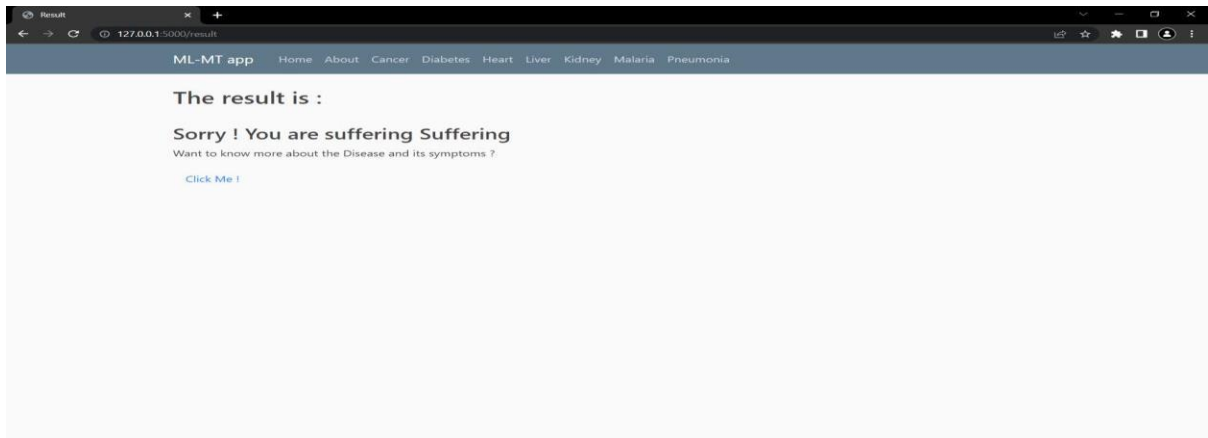


Fig: 6.6: Heart Disease Predict

This is the image upload UI for Malaria disease prediction system. First, the input image has to be selected and then the upload button has to be clicked to upload the image.

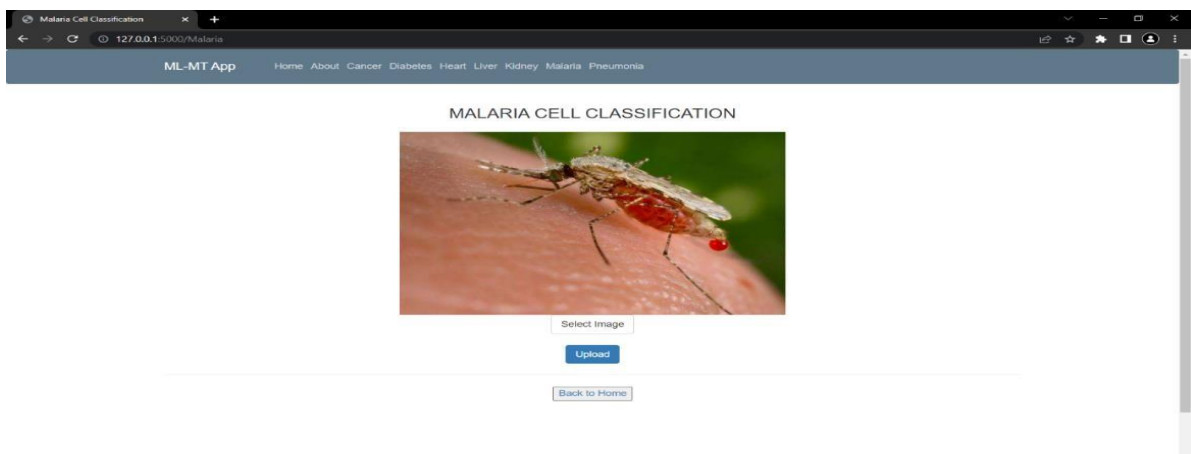


Fig: 6.7: Malaria image

This is the image upload UI for Pneumonia disease prediction system. First, the input image has to be selected and then the upload button has to be clicked to upload the image.

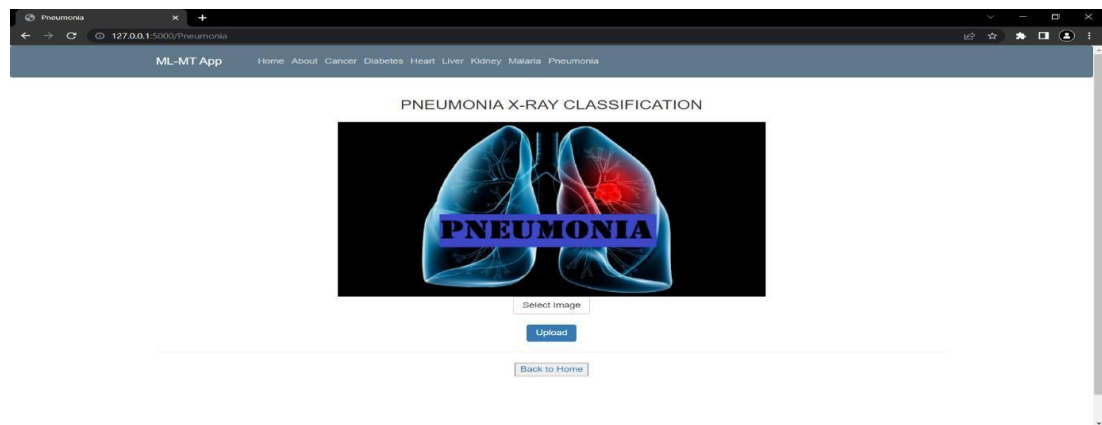


Fig: 6.8: Pneumonia image input.

REFERENCES

- [1] Alexander Selvikvag; Lundervold; Arvid Lundervold; Katarzyna Węgrzyn-Wolska, "An overview of deep learning in medical imaging focusing on MRI", *Zeitschrift für Medizinische Physik*, Volume 29, Issue 2, Pages 102-127, May 2019.
- [2] Alice Othmania; Abdul Rahman Taleb; Hazem Abdelkawy; Abdenour Hadid, "Age estimation from faces using deep learning: a comparative analysis", *Computer Vision and Image Understanding*, Volume 196, 102961, July 2020.
- [3] Durr-e-maknoon Nisar; Meshrif Alruily; Sultan H. Al Motor; Rashid Amin, "Healthcare Techniques Through Deep Learning: Issues, Challenges and Opportunities", 2019.
- [4] Hiraj Dahiwade; Gajanan Patle; Ektaa Meshram, "Designing Disease Prediction Model Using Machine Learning Approach", 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019.
- [5] Jerome Thevenot; Miguel Bordallo Lopez; Abdenour Hadid, "A Survey On Computer Vision For Assistive Medical Diagnosis From Faces", *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS*, VOL. 22, No. 5, 2018.
- [6] Liva Faes; Xiaoxuan Liu; Pearse A Kean, "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis", *The Lancet Digital Health*, Volume 1, Issue 6, October 2019
- [7] M. Chen, Y. Hao, K. Hwang, L. Wang and L. Wang, "Disease prediction by machine learning over big data from healthcare communities", *IEEE Access*, vol. 5, no. 1, pp. 8869-8879, 2017
- [8] Pushpa Singh; Narendra Singh; Krishna Kant Singh; Akansha Singh, "Diagnosing of disease using machine learning", 2021, *Machine Learning and the Internet of Medical Things in Healthcare* Pages 89- 111, 2021.
- [9] Rahatara Ferdousi; M. Anwar Hossain; Abdulmotaleb El Saddik, "Early-Stage Risk Prediction of Non-Communicable Disease Using Machine Learning in Health CPS", *IEEE Special Section on AI And IoT Convergence for Smart Health*, July 2021.

- [10] Yuanyuan Pan; Minghuan Fu; Biao Cheng; Xuefei Tao; Jing Guo, "Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things Platform, IEEE Access (Volume: 8), May 2020.

APPENDIX

A. RESEARCH PAPER

DISEASE PREDICTION AND MEDICAL CHECK USING LOGISTIC REGRESSION AND CNN

Chitra P, Pushpak Vinay B, Sai B

Department of Electronics and Communication , Sathyabama institute of science and technology
Chennai, India

Abstract— The ability to detect diseases early is crucial to enhancing the standard of care and can assist people in averting potentially dangerous health conditions before it is too late. Applications like disease prediction and treatment response prediction have demonstrated more potential for deep learning. In addition to influencing healthcare as a whole, deep learning techniques have steadily increased the Input is obtained in the form of numeric values for diseases such as cancer, diabetes, and cardiovascular conditions. To predict diseases such as Pneumonia, malaria, or kidney diseases, an image is used as an input. Output results are acquired instantly in real-time in the form of a probable disease, its causes, and the aftermath. We will be using Deep Learning algorithms, mainly CNN and DNN, and Image Processing.

Keywords— CNN, Diseases, Logistic regression

I. INTRODUCTION

Accurately predicting diseases plays a significant role in public health, especially at the early stage which allows patients to take prevention treatments in time. With the growing volume and availability of electronic health records (EHRs), predictive modeling tasks for disease progression and analysis have obtained increasing interest from researchers. The EHR data are temporally sequenced by patient visits with each visit represented as a set of high dimensional clinical events. Mining EHRs is especially challenging compared to standard data mining tasks, due to their noisy, irregular, and heterogeneous nature. A conventional approach of disease prediction is the one-size-fits-all model. That is, using all available training data to build a global model, and then with this model, predicting the risk of diseases for each patient. The benefit of applying a one-size-fits-all model is that it captures the overall information of the entire training population. However, patients may have different phenotypes, different medical conditions, etc. Using a global model may miss some specific information that is important for individual patients. Thus, building a targeted, patient-specific model for each individual patient is urgent and important for personalized medicine. Recent studies show that personalized models can improve predictive performance over global models. A general framework for personalized prediction contains two stages: (1) measuring

the similarity among patients, and (2) building a separate model for each patient using his/her similar cohorts. This framework is motivated by the working process of human doctors, i.e., after reviewing or recalling the diagnosed patients with similar diseases or symptoms, the doctors then carefully make decisions. If doctors can find similar patients, the probability of successfully curing this patient may improve a lot. Many similar learning methods have been proposed on healthcare datasets.

II. LITERATURE SURVEY

[1] The potential for disease diagnosis using deep learning algorithms is huge. Although additional research on the application of such algorithms in algorithms have accuracy levels comparable to those of healthcare professionals. The most significant discovery relates to methodology and reporting, which indicates that an exaggerated claim from a poorly planned or insufficiently documented study could damage the credibility an [2] The authors propose a novel multimodal illness risk prediction system that utilizes convolutional neural networks (CNNs) and incorporates both structured and unstructured data from the hospital. This approach is unique in the field of medical big data analytics as no previous work has specifically addressed both data types. Compared to many conventional prediction algorithms, the suggested approach achieves a prediction accuracy of 94.8% and converges faster than the CNN-based unimodal disease risk prediction algorithm.

[3] Using machine learning techniques for disease detection, such as SVM, KNN, Naive Bayes, and Decision trees, can improve decision-making speed and lower the incidence of false positives. Python is used to discuss the practical implementation of these methods. These algorithms are used to make diagnoses for a number of conditions, including cancer, diabetes, epilepsy, heart attacks, and other serious illnesses.

[4] Deep neural networks are now the most advanced machine learning models used in both academia and industry, with applications ranging from image analysis to natural language processing. These advancements slowly but surely

are beginning to realize the enormous promise for medical imaging technologies, medical data analysis, medical diagnostics, and healthcare in general.

[5] The Enhanced Deep learning aided Convolutional Neural Network (EDCNN) has been proposed as a tool to improve patient prognostics for heart disease. The EDCNN model features a multi-layer perceptron with regularization learning methods and a deeper architecture than traditional models. It has been integrated into the Internet of Medical Things Platform (IoMT) to support decision-making units.

[6] The K-Nearest Neighbor (KNN) and Convolutional Neural Network (CNN) machine learning algorithms for disease prediction in order to provide accurate results. Dataset of disease symptoms is needed for disease prediction. The person's lifestyle choices

[7] Any system's main components are its tools and algorithms. Tools are used in deep learning to transform data into useful information. According to supervised and unsupervised learning methodologies, deep learning models are divided into categories. Because of how crucial these algorithms are, medical instruments like the ECG, phonendoscope, and microscope should be created throughout time using these models to account for physicians' limited perceptual abilities. Deep learning is playing a crucial part in the processing and forecasting of huge information.

[8] Insights into the suitability of each approach in the evaluation of various types of symptoms connected to medical disorders have been provided by the description of a variety of remote and non-invasive imaging solutions. According to the research, over 30 illnesses can be tentatively identified using computer vision techniques by automatically detecting some of their symptoms.

[9] In the field of deep learning, new algorithms and tactics are discussed. For radiologists of the future, DLA can support clinical choices. DLA can automate radiologists' workflow and aid inexperienced radiologists' decision-making. By automatically finding and classifying lesions to make a more accurate diagnosis, DLA is designed to help doctors. Medical image analysis processing using DLA can help doctors reduce medical errors and work more efficiently.

III. IMPLEMENTATION

Module 1: Data Acquisition And Pre-Processing

a. Data Gathering

We have gathered the data for our project for the following diseases in the CSV and Image format:

Alzheimer : CSV dataset

Cancer : CSV dataset

Diabetes: CSV dataset

Heart: CSV dataset

Kidney: CSV dataset

Liver: CSV dataset

Pneumonia: Image dataset

b. Data Preprocessing

The process of transforming raw data into a useful and efficient format, known as data preprocessing, is a data mining approach. This process modifies the data's structure to enable faster and more effective completion of data science tasks such as machine learning. These techniques are typically applied at the beginning of the machine learning and AI development pipeline to ensure accurate results. While there are many tools and strategies for data preparation, we have chosen to focus on the following measures for our dataset.

For dataset in CSV file Format

should appear outside of the closing quotation mark. For example, "The term 'artificial intelligence' is often used in the tech industry," where the period appears outside of the closing quotation mark because it applies to the sentence as a whole, not just the highlighted phrase. However, if the highlighted phrase is part of a larger sentence, the punctuation should appear within the quotation marks. For example, "She asked, 'Are you coming to the party?'" Here, the question mark applies specifically to the quoted question, so it appears within the quotation marks. should appear outside of the quotation marks. When a parenthetical phrase or statement is at the end of a sentence, the punctuation should be placed outside of the closing parenthesis. For example, "She completed her assignment on time (despite the difficulties she faced)." In this case, the period is placed outside of the closing parenthesis. However, if the entire sentence is parenthetical, then the punctuation should be placed within the parentheses. For example, "His favorite color (as he often stated) was green." Here, the entire sentence is parenthetical, so the comma is placed within the parentheses.

Removing Unwanted Columns:

Our dataset contains redundant data. We consider it a good idea to eliminate these features from the dataset because their presence has no impact on the target. To ensure that we only work with relevant data, this method of deleting unnecessary features and maintaining only the required features in the dataset is quite helpful.

Therefore, we drop a few columns from our dataset using the drop command. Python's "df.drop" function can be used to remove a single column or a number of columns from a pandas data frame. We further define the title of the column, index, axis and labels.

Identifying Null Values

The Kaggle dataset we are working with has some missing elements, which may be due to inefficient data collection or simply because the data does not exist. These missing elements are denoted by NaN (not a number) or None, and they can make our computations difficult and inaccurate. To address this issue, we need to identify the missing data and replace them with usable elements.

Pandas treats None and NaN similarly when indicating missing or null values. To facilitate this convention, Pandas provides several useful utilities for detecting, removing, and replacing null values in a DataFrame. The `isnull()` and `notnull()` functions are used in Pandas DataFrame to check for missing values, and they assist in determining whether a value is NaN or not. These functions can also be used with Pandas Series to identify null values in a series.

Removing Null Values

After identifying the null values in a dataset, we can remove them by dropping the rows and columns that contain them. This can be accomplished using the `dropna()` function, which removes any rows or columns that contain null values. If we do not want to delete the entire column of null values, we can use the `fillna()` function, which loops through the database and fills in the null values with the desired value.

The dataset obtained from Kaggle may have missing elements represented by NaN or None, which can make data processing inaccurate and complicated. Therefore, we need to identify and replace these missing values with suitable elements to make our data analysis more reliable.

Resizing The Images

The Python interpreter has the ability to manipulate images using the PIL (Python Imaging Library). The PIL image is represented by a class of the same name that is available in the Image module. The module also provides additional factory functions that can be used for creating new images or loading images from files.

An image's new size is returned by the function `image.resize()`. The following parameters are taken into account:

Size: The intended size in pixels is represented as a 2-tuple. (Length, Width)

Grayscale is the process of transforming a picture from a different colour space to different shades of grey, including RGB, CMYK, HSV, and others. Either all white or all black is possible. Import OpenCV, use `imread()` to read the original image, and then use `cv2.cvtColor()` to convert the colour image to grayscale. The `destroyAllWindows()` function enables users to instantly close or destroy all windows when the script has finished running.

Removing Noisy Images

We can remove the noisy parts of any image in the dataset using the command `cv2.fastNlMeansDenoisingColored()`. The following parameters have been defined in our model for the same:

- **h:** the filter strength determining parameter. A higher h value removes noise more effectively, but it also removes image details.
- **ForColorComponents:** the same as h, but only for colour images. (usually the same as h)
- **templateWindowSize:** This should be unusual. (Suggested 7)

- **searchWindowSize:** this should be unusual. (recommended number 21)

Module 2: Model Training and Validation

a. Model Creation Using Logistic Regression

The algorithms used for our model are binary classification with logistic regression. Logistic regression is used in the classifier construction process. Compared to linear regression, logistic regression is more sophisticated. This is because linear regression is unable to categorize data that are widely dispersed in a given location. The data can be classified using linear regression, where the line can separate the input data into two major categories (or classes). The line cannot clearly separate the data into two classes when there is overlap in the data. Logistic regression gets around this restriction.

In order to predict binary values in a set of independent variables (1/0, Yes/No, True/False), logistic regression, one of the classification techniques, is utilized. Dummy variables are used to represent binary or categorical values. When the outcome variable is categorical variable.

IV. RESULT

We create multiple forms for all the diseases in our model- Alzheimer, cancer, heart disease, lung disease, kidney disease, cancer, and pneumonia. Then we combine all these into a single HTML file using the flask module and python notebook.

Diabetes

Diabetes Predictor

Number of Pregnancies eg. 0

Glucose (mg/dL) eg. 80

Blood Pressure (mmHg) eg. 80

Skin Thickness (mm) eg. 20

Insulin Level (IU/mL) eg. 80

Body Mass Index (kg/m²) eg. 23.1

Diabetes Pedigree Function eg. 0.52

Age (years) eg. 34

Figure 1: diabetes inputs

ere is a sample of combining multiple pages in app to create a single web page where the user can enter his/her information as input and then the app predicts whether the

person has the disease or not.

specified training data. It essentially finds the best values for the model's parameters based on the input data, enabling the model to make more accurate predictions on unseen or new data. Once the fit method has been run, the trained model can be used to predict new data using the 'predict' method. The 'fit' method is a fundamental part of the machine learning workflow and is used in a wide range of applications, including classification, regression, clustering, and more.

Heart

Heart Disease Predictor

age	sex(Male:1, female:0)	chest pain type
resting blood pressure in mm	serum cholestoral in mg/dl	fasting blood sugar 120 mg/c
resting electrocardiographic r	maximum heart rate achieved	exercise induced angina (1 =
ST depression induced by exe	the slope of the peak exercise	number of major vessels (0-3
3 = normal; 6 = fixed defect;		

Predict

Figure 2: heart inputs

Kidney

Kidney Disease Predictor

age	bp	al
su	rbc	pc
pcc	ba	bgr
bu	sc	pot
wc	htn	dm
cad	pe	ane

Predict

Figure 3: kidney inputs

Liver

Liver Disease Predictor

Age	Total Bilirubin
Direct Bilirubin	Alkaline Phosphatase
brown	
Alamine Aminotransferase	Aspartate Aminotransferase
Total Protiens	Albumin
Albumin and Globulin Ratio	Gender(Male: 1, Female: 0)

Predict

Figure 3: liver inputs

Outputs

If he was healthy:

The output will look like result will get congrats you are healthy.

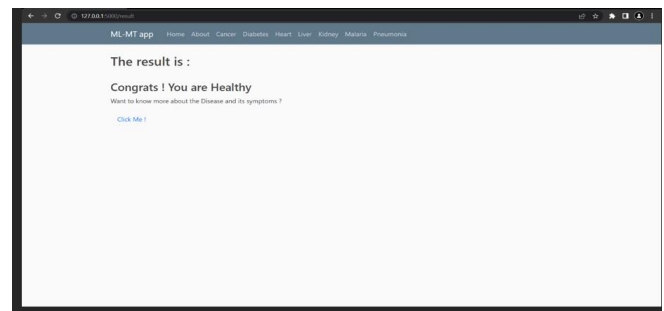


Figure 4: output for not infected

If he was effected :

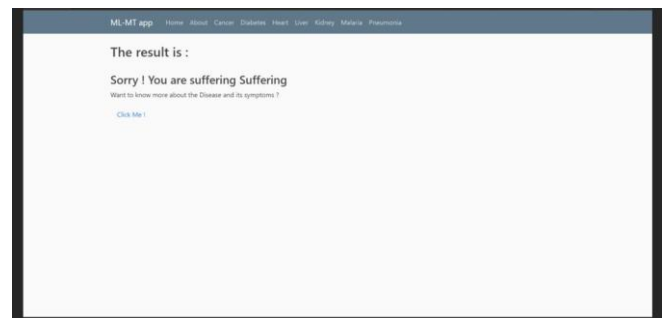


Figure 5 : output for infected

Module 3: Web App Deployment

The server side processing will be handled by code that we write. Requests will be sent to our code. It will determine the subject matter and nature of the requests. Additionally, it will decide what kind of reply to send to the user. We'll use Flask for all of this. It simplifies the web application design

Inside the directory that will house the project files, we will construct a virtual environment. first, use pip to install virtualenv. Use the command pip install Flask to install Flask after installing virtualenv, creating, and activating a virtual environment. Now that the project's framework has been established, functionality addition may start.

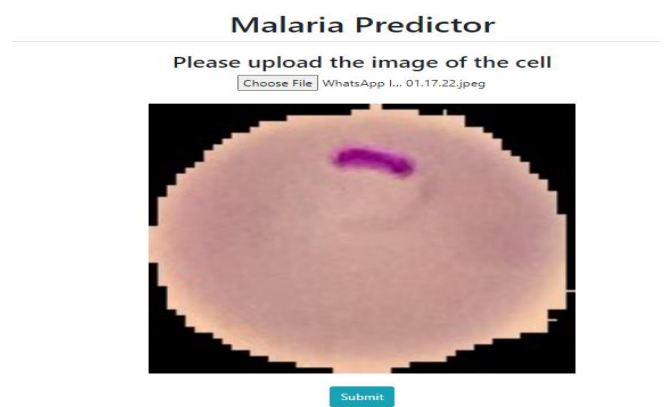
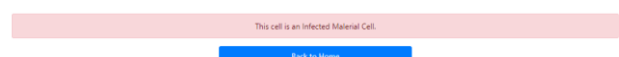


Figure 6: input for malaria



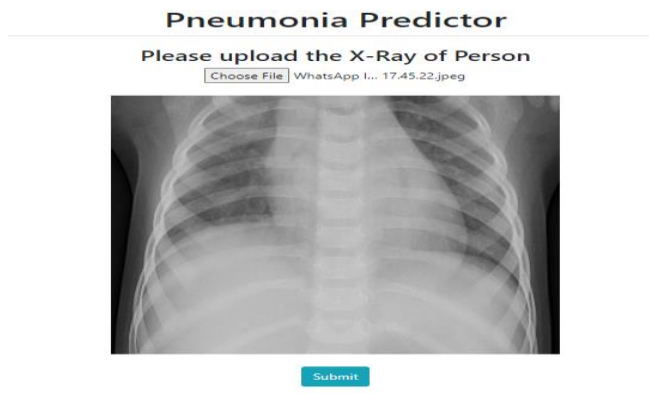


Figure 7: input for pneumonia

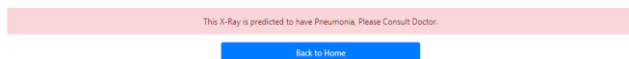


Figure 8 : output for pneumonia

After saving it in a file and running the script, we will have the opportunity to view the output. Then navigate to the URL that was provided there, and you will see your first webpage showing the message on your local server. The `route()` decorator in Flask is what is used to connect a URL to a

After that, we create the form by utilizing HTTP methods within Flask. The Hypertext Transfer Protocol, also known as HTTP, serves as the basis for data transfer on the World Wide Web. This protocol defines a variety of different ways that can be utilized to retrieve data from a given URL. The GET requests are handled by the Flask route as their default response. However, you can alter this decision by giving an alternative set of methods as the `route()` decorator's option., to show how the POST method is used in URL routing. We can then use this to show how URL routing uses the POST method. Using this method, we are able to generate the HTML login page with information such as the user's name, age, gender, and any other disease-related input.

function, as we discover as we go deeper into the configuration. Now, in order to increase the functionality of our straightforward web application, we have also added a different technique called `add url rule ()`. This method, which, much like `route()` was in the preceding example. To get things rolling, a new file containing an introduction to our website

is generated in the same directory. We give this file the name `main.py`. We can now run the program by using the commands on the terminal after instructing it which application to run. `process`. Flask enables us to concentrate on what the consumers are asking for and the appropriate kind of response to provide.

field with the necessary data, and then click the submit button. If the sickness was discovered or not, the outcome will indicate that. In order to build the server and save this HTML file, we finally try this Python script. Once the development server has started, open your HTML file in the browser, fill out the text

V. REFERENCES:

- [1] Liva Faes, Xiaoxuan Liu, and Pearse A. Kean, "Deep learning performance vs health-care professionals in diagnosing disorders from medical imaging: a systematic review and meta-analysis", *The Lancet Digital Health*, Volume 1, Issue 6, October 2019,
- [2] Disease prediction using machine learning over big data from healthcare communities is described in by M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang. *IEEE Access*, vol. 5, no. 1, pp. 8869-8879, 2017.
- [3] Machine Learning and the Internet of Medical Things in Healthcare Pages 89–111, Pushpa Singh, Narendra Singh, Krishna Kant Singh, and Akansha Singh, "Diagnosing of Disease Using Machine Learning", 2021.
- [4] Katarzyna Wgrzyn-Wolska, Lundervold, Arvid, and Alexander Selvikvag, "An overview of deep learning in medical imaging concentrating on
- [5] Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things
- [6] "Designing Disease Prediction Model Using Machine Learning Approach," 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019.
- [7] "Healthcare Techniques Through Deep Learning: Issues, Challenges and Opportunities," 2019 Dur-e-maknoon Nisar, Meshrif Alruily, Sultan H. Almotir, and Rashid Amin
- [8] Jérôme Thevenot, Miguel Bordallo Lopez, and Abdenour Hadid, "A Survey On Computer Vision For Assistive Medical Diagnosis From Faces," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS*, VOL. 22, No. 5, 2018,