PARKINSON'S DISEASE DETECTION

A Project Report

Submitted to the Kerala Technological University

in partial fulfillment of requirements for the award of degree

in

Master of Computer Applications

by

SULFIKAR ALI.A

KVE20MCA-2014

DEPARTMENT OF MCA

KVM COLLEGE OF ENGINEERING and IT,Cherthala

ALAPPUZHA,KERALA

JULY 2022

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

KVM COLLEGE OF ENGINEERING AND IT

CHERTHALA, KERALA

2020 - 22

CERTIFICATE

This is to certify that the report entitled “**PARKINSON’S DISEASE DETECTION**“ submitted by **SULFIKAR ALI.A** (KVE20MCA-2014) to Department of MCA in partial fulfillment of the Master of computer applications degree is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Ms.Darsana Ramachandran
(Guide)
Head of the Department
Department of MCA
KVM College Of Engineering and IT
Cherthala, Kerala

Ms. Testa K.U
(Co-Guide)
Assistant Professor
Department of MCA
KVM College Of Engineering and IT
Cherthala, Kerala

Ms.Darsana Ramachandran
Associate Professor
Head of the Department
Department of MCA
KVM College Of Engineering and IT
Cherthala, Kerala

College Seal

DECLARATION

I, SULFIKAR ALI.A hereby declare that the project report **PARKINSON'S DISEASE DETECTION**, submitted for partial fulfillment of the requirements for the degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Ms.Darsana Ramachandran , Assistant Professor, Dept of CSE, KVM CE & IT.

, hereby declare that the project report entitled **" PARKINSON'S DISEASE DETECTION"**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the Kerala Technological University, Kerala is a bonafide work done by me under supervision of Ms.Darsana Ramachandran.

This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources.

I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Cherthala SULFIKAR ALI.A

/07/2022

ACKNOWLEDGEMENT

I take this opportunity to express my deepest sense of gratitude and sincere thanks to everyone who helped me to complete this work successfully. I express my sincere thanks to **Dr.Manjula Devananda**, Principal and HOD , CSE , KVM College of Engineering and IT , Cherthala for providing me with all the necessary facilities and support.

I would like to express my sincere gratitude to **Ms. Testa K.U**, Assistant Professor, KVM College of Engineering and IT, Cherthala for her support and co-operation. I would like to place on record my sincere gratitude to my project guide **Ms.Darsana Ramachandran**, Associate Professor, Head of the Department, Department of MCA, KVM College of Engineering and IT for the guidance and mentorship throughout the course.

I thank my Almighty God for giving me the strength to venture for such an enigmatic logical creation in a jovial way. Finally I thank my family, and friends who contributed their support for the successful fulfillment of this seminar work.

SULFIKAR ALI.A

ABSTRACT

Parkinson's disease (PD) is a neuro degenerative movement disease where the symptoms gradually develop start with a slight tremor in one hand and a feeling of stiffness in the body and it became worse over time. It affects over 6 million people worldwide. However there is no recognized test for PD for patients, particularly in the early stages. This results in increased mortality rate. Thus detection system of Parkinson's disease with easy steps and feasible one to detect parkinson's disease at the early stage is essential

This project aims to employ Machine Learning (ML) classifying algorithms to predict whether the patient has Parkinson's Disease (PD) or not. We can detect parkinson's disease in many ways according to there symptoms.Here we predict disease by voice features and spiral drawing(tremor detection). our experimental results will show early detection of disease which facilitates clinical monitoring of elderly people and increase their life span by improving their lifestyle which leads to a peaceful life.

Contents

Chapter 1

INTRODUCTION

Parkinson disease (PD) is a neurological disorder based on dopamine receptors. Parkinson disease mostly causes problems in moving around. It can cause a person to move very slowly. Parkinson is a progressive neurological condition, which is characterized by both motor (movement) and non-motor symptoms. Apart from many common symptoms each person will experience and demonstrate an individual presentation of the condition. A person with Parkinson disease appears stiff or rigid. At times, a person with Parkinson disease may appear to suddenly "freeze up" or be unable to move for a short period of time. Parkinson disease is a progressive neuro degenerative condition resulting from the death of the dopamine containing cells of the substantia nigra. There is no consistently reliable test that can distinguish Parkinson disease from other conditions that have similar clinical presentations. The diagnosis is primarily a clinical one based on the history and examination. People with Parkinson disease classically present with the symptoms and signs associated with Parkinsonism, namely hypokinesia (i.e. lack of movement), bradykinesia (i.e. slowness of movement), rigidity (wrist, shoulder and neck.) and rest tremor (imbalance of neurotransmitters, dopamine and acetylcholine). Parkinsonism can also be caused by drugs and less common conditions such as: multiple cerebral infarction, and degenerative conditions such as progressive supra nuclear palsy (PSP) and multiple system atrophy (MSA). Although Parkinson disease is predominantly a movement disorder, other impairments frequently develop, including psychiatric problems such as depression and dementia. Autonomic disturbances and pain may later ensue, and the condition progresses to cause significant disability and handicap with impaired quality of life for the affected person. Family and carers might get affected indirectly.

1.1 BACKGROUND OF STUDY

The clinical syndrome, paralysis agitans, was first studied by James Parkinson and mentioned in his 1817 essay . To honor his contributions, the disease was later named after him. PD was first known as lessened muscular power and hence followed by movement disorders . These movement disorders mostly include tremors, rigidity, and postural instability . Consequently, many studies have been contributed to these motor symptoms, both clinically and computationally. Moreover, there are studies based on brain imaging and other signs the patient might experience.

1.1.1 Motor symptoms

As mentioned, PD is a central nervous system disorder resulting in a loss of motor function, increased slowness, and rigidity. The most visible symptoms are related to motor functions. AI-based techniques can be useful to detect signs like tremor or bradykinesia (refers to slowness of movement). Unfortunately, these symptoms do not help us diagnose the disease early as they become apparent later.

Publications have covered a high number of techniques for automated detection of PD motor symptoms using various methods like Neural Networks, Hidden Markov Models, and Support Vector Machines. Some have used IMU (inertial measurement unit) sensor data for automated assessment of movement disorders .

1.1.2 Non-motor symptoms

In 1872, neurologist Jean-Martin Charcot studied tremors, and his essential contribution to the study of Parkinson's disease was the differentiation of this disorder from other tremorous disorders. Examining large numbers of patients, he developed a method to identify patients suffering from both action and rest tremors.

He observed patients with active tremor had symptoms like weakness, spasticity, and visual disturbance. In contrast, those with rest tremors differed in rigidity, slowed movements, and a very soft speech . This was the very first time that speech symptoms took a severe role in determining PD occurrence.

About 90 percentage of people with PD experience changes in speech and voice at the same time during the disease . Yet the exact relation between the disease variability and voice disability is unknown. Speech disorders in patients with PD are characterized by monotonous, soft, and breathy speech with variable rate and frequent word-finding difficulties

1.2 OBJECTIVES

Diagnosis of Parkinson's disease (PD) is commonly based on medical observations and assessment of clinical signs, including the characterization of a variety of motor symptoms. However, traditional diagnostic approaches may suffer from subjectivity as they rely on the evaluation of movements that are sometimes subtle to human eyes and therefore difficult to classify, leading to possible misclassification. In the meantime, early non-motor symptoms of PD may be mild and can be caused by many other conditions. Therefore, these symptoms are often overlooked, making diagnosis of PD at an early stage challenging. To address these difficulties and to refine the diagnosis and assessment procedures of PD, machine learning methods have been implemented for the classification of PD and healthy controls or patients with similar clinical presentations (e.g., movement disorders or other Parkinsonian syndromes).

The main ojective of this project is to build a model to accurately detect the presence of Parkinson's disease in an individual.We will use here CNN Algorithm, Support Vector Machines (SVMs) and utilize the data-set available on UCL Parkinson Data-set.

Chapter 2

LITERATURE SURVEY

A literature review is a comprehensive summary of previous research on a topic. The literature review surveys scholarly articles, books, and other sources relevant to a particular area of research. The review should enumerate, describe, summarize, objectively evaluate and clarify this previous research.

From time to time, several notable attempts were done by various researchers for detecting Parkinson's disease. The following is a brief review of some work done for detecting Parkinson disease from voice samples of subjects

2.1 A Detection Support System for Parkinson's Disease Diagnosis Using Classification and Regression Tree

Authors :- Hadjahamadi, A.H. and Askari, T.J. (2012).

Findings ;-

In this paper, The detection of PD is very important at the early stage. The detection can be performed using data mining technique. This paper theoretically explains the algorithms to detect PD such as Naive Bayes, support vector machine (SVM), multi layer perceptron neural network (MLP) and decision tree. This paper has taken 8 patients voice input dataset and checked their performance with four types of classifiers such as Naïve Bayes, SVM, MLP neural network, and decision tree. [].

2.2 A step towards the automated diagnosis of Parkinson's disease: analyzing handwriting movements.

Authors :- Pereira, C.R

Findings :-

Pereira et al. applied machine learning and computer vision techniques on their own designed handwriting dataset called "Hand PD" to deal with PD recognition. Here they have used some supervised techniques such as OPF, SVM and NaiveBayes classifier (NB) to evaluate the dataset among which NaiveBayes classifier (NB) have resulted the best accuracy which is 78.9 percentage. This accuracy is quite low and needs to be improved. [].

2.3 Detecting of Parkinson Disease through Voice Signal Features

Authors :- Alemami, Y. and Almazaydeh, L. (2014)

Findings :-

This paper predicts parkinson's disease from voice input with acoustic devices. In this paper, people from different locations and voice parameters are analyzed to predict PD among the patients. Multi layer Perceptron (MLP) and Logistic Regression (LR) frameworks were used to recognize parkinson's disease (PD) from the voice dataset. Different kernel values were used to get the best possible accuracy . The linear kernel SVM (Support Vector Machine) produced the best accuracy.SVM gives high accuracy of 70 percentage for training and 30 percentage for the test when implemented on a discretized PD dataset and a splitting dataset. The MLP neural network gives the highest accuracy when used to classify actual PD dataset without discretization, attribute selection, or changing test mode. []

2.4 A comparison of multiple classification methods for diagnosis of Parkinson disease. Expert Systems with Applications

Authors :- Das, R. (2010)

Findings :-

Resul Das compares four different types of classification algorithms in the machine learning domain for enabling the diagnosis of Parkinson's disease. In the study a SAS based software has been used to model various classifiers which can recognise the presence of PD. The classifiers employed are – DM Neural, Neural Network, Regression and Decision Tree. Various evaluation methods were run to evaluate the efficiency of the classifiers and the accuracy of NN yielded 92.9 percentage correct classification rate. [].

2.5 Classification of Parkinson disease datawith artificial neural networks. In IOP Conference Series

Authors :-Yasar, Saritas, Sahman Cinar.

Findings :-

. A.Yasar et al used artificial neural networksfor the detection of Parkinson's disease. The dataset was taken from UCI machine learning repository. Using the MATLAB tool, 45 properties were chosen as input values and one output for the classification. Their proposed model was able to distinguish the healthy subjects from the PD subjects with an accuracy of 94.93 percentage .From the review above, it may be observed that various ML techniques have been applied in recent research works over voice based PD detection. But it may be observed that in none of these works, the ensemble based ML approaches like the XGBoost were used for model construction [].

Table 2.1: Summary of related works

SL.NO	TITLE	AUTHOR	YEAR	FINDINGS
1	A Detection Support System for Parkinson's Dis- ease Diagnosis Using Classification and Regression Tree	AHadjahamadi, A.H. and Askari, T.J	2012	the algorithms to detect PD such as Naive Bayes, support vector machine (SVM), multilayer perceptron neural network (MLP) and decision tree .
2	A step towards the automated diagnosis of Parkin- son's disease: analyzing handwriting movements	Pereira, C.R	2012	applied machine learning and computer vision techniques on their own designed handwriting dataset called "Hand PD" to deal with PD recognition..
3	Detecting of Parkinson Disease through Voice Sig- nal Features	Alemami, Y. and Almazaydeh, L	2014	people from different locations and voice parameters are analyzed to predict PD among the patient.
4	A comparison of multiple classification methods for diagnosis of Parkinson disease. Expert Systems with Applications	Das, R	2010	compares four different types of classification algorithms in the machine learning domain for enabling the diagnosis of Parkinson's disease.
5	Classification of Parkinson disease datawith artificial neural networks	Yasar, Saritas, Sahman , Cinar.	2012	. Using the MATLAB tool, 45 properties were chosen as input values and one output for the classification.

Chapter 3

SYSTEM ANALYSIS

3.1 ANALYSIS OF DATASET

A dataset in machine learning is, quite simply, a collection of data pieces that can be treated by a computer as a single unit for analytic and prediction purposes. This means that the data collected should be made uniform and understandable for a machine that doesn't see data the same way as humans do. For this, after collecting the data, it's important to preprocess it by cleaning and completing it, as well as annotate the data by adding meaningful tags readable by a computer.

3.1.1 VOICE DATASET ANALYSIS

For the voice analysis, the data is collected which contains the voice data for both PD and healthy subjects . This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals .The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.The data is in ASCII CSV format. The rows of the CSV file contain an instance corresponding to one voice recording. There are around six recordings per patient, the name of the patient is identified in the first column.

Sl. No	Attributes	Information
1	name	ASCII subject name and recording number
2	MDVP:Fo(Hz)	Average vocal fundamental frequency
3	MDVP:Fhi(Hz)	Maximum vocal fundamental frequency
4	MDVP:Flo(Hz)	Minimum vocal fundamental frequency
5	MDVP:Jitter(%)	Several measures of variation in fundamental frequency
6	MDVP:Jitter(Abs)	
7	MDVP:RAP	
8	MDVP:PPQ	
9	Jitter:DDP	
10	MDVP:Shimmer	Several measures of variation in amplitude
11	MDVP:Shimmer(dB)	
12	Shimmer:APQ3	
13	Shimmer:APQ5	
14	MDVP:APQ	
15	Shimmer:DDA	
16	NHR	Two measures of ratio of noise to tonal components in the v
17	HNR	
18	RPDE	Two nonlinear dynamical complexity measures
19	Spread1	
20	Spread2	
21	PPE	Pitch Period Entropy
22	D2	Recurrence Period Density Analysis
23	DFA	Signal fractal scaling exponent
	status	Health status of the subject (one) - Parkinson's, (zero) - heal

Figure 3.1: voice Analysis

3.1.2 SPIRAL DRAWING ANALYSIS

Spiral drawing dataset of PD affected and unaffected patients collected by neurologists are obtained from Machine Learning repository. These are stored into the python environment as Testing and Training dataset and imported using necessary packages. Python is an open source dynamic, high level, free and interpreted programming language. This supports object oriented programming and procedural programming.

Images of healthy and patients with Parkinson's drawing spirals. The images are further divided into training and testing groups for comparing (or reproducing) the results of the original publication.

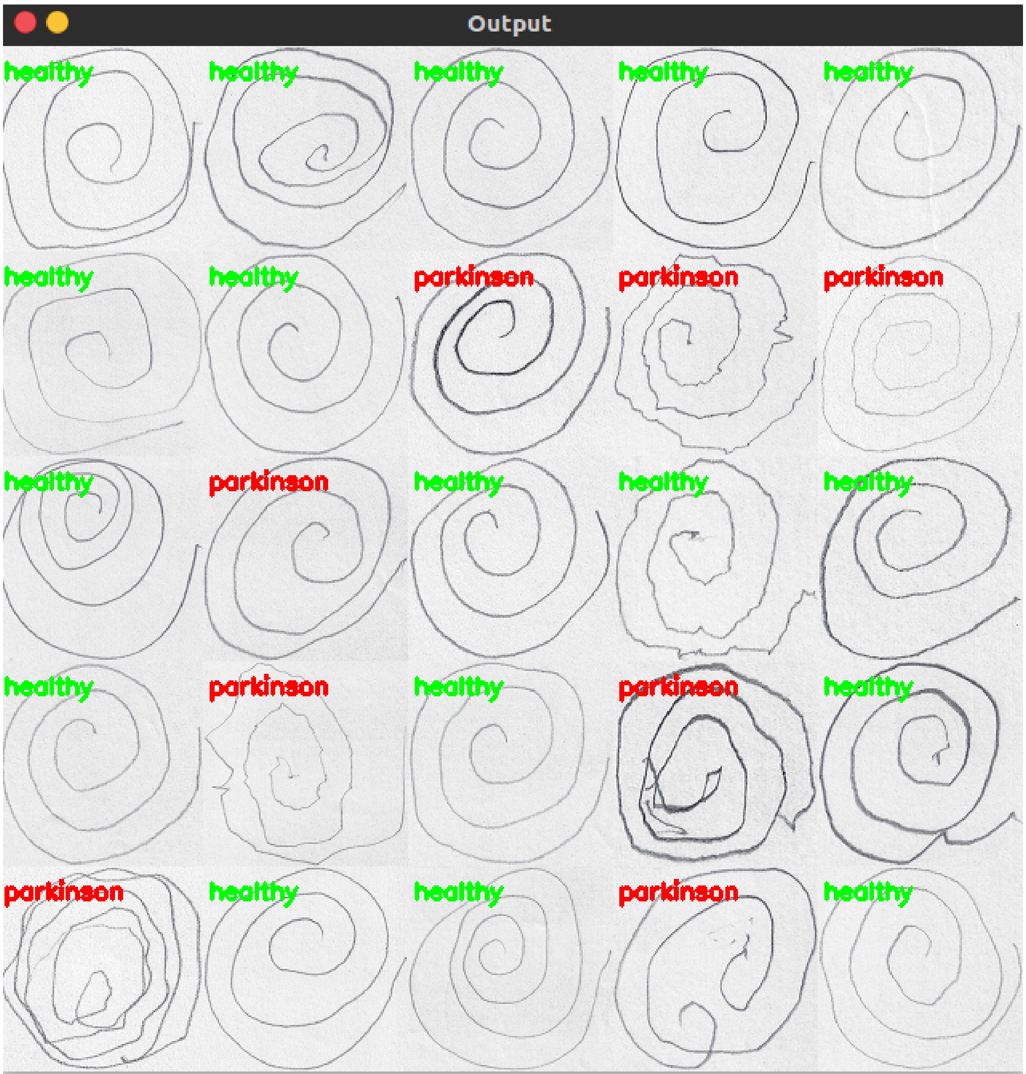

Figure 3.2: Spiral Analysis

3.2 DATA PREPROCESSING

Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the data mining process. More recently, data preprocessing techniques have been adapted for training machine learning models and AI models and for running inferences against them.Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks.

The data obtained will be cleansed with data null or not applicability and the unwanted colums from the dataset will be discarded. As part of the data mining process, data preparation is a crucial stage. Projects involving data mining and machine learning are particularly susceptible to the adage ”garbage in, garbage out.” Sometimes, data collection techniques are weakly regulated, resulting in unreliable results such as out-of-range or non-existent numbers.

Pre-processing of image generally removes low frequency background noise, normalizes the intensification of the individual practical image, removes reflection of light to get rid of the image noise, and prepares the face image to better feature extraction. In our system, we have first resized the images . Then We have converted the image to an array of pixel value. Each pixel value of the array is converted to float and divided by 255.0 so that all the pixel values comes to a range between 0 to 1.

3.3 ANALYSIS OF ALGORITHMS

The term ”analysis of algorithms” was coined by Donald Knuth.Algorithm analysis is an important part of computational complexity theory, which provides theoretical estimation for the required resources of an algorithm to solve a specific computational problem. Most algorithms are designed to work with inputs of arbitrary length. Analysis of algorithms is the determination of the amount of time and space resources required to execute it.

Usually, the efficiency or running time of an algorithm is stated as a function relating the input length to the number of steps, known as time complexity, or volume of memory, known as space complexity.

3.3.1 Convolutional Neural Network (ConvNet/CNN)

A Convolutional Neural Network (ConvNet/CNN) is algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

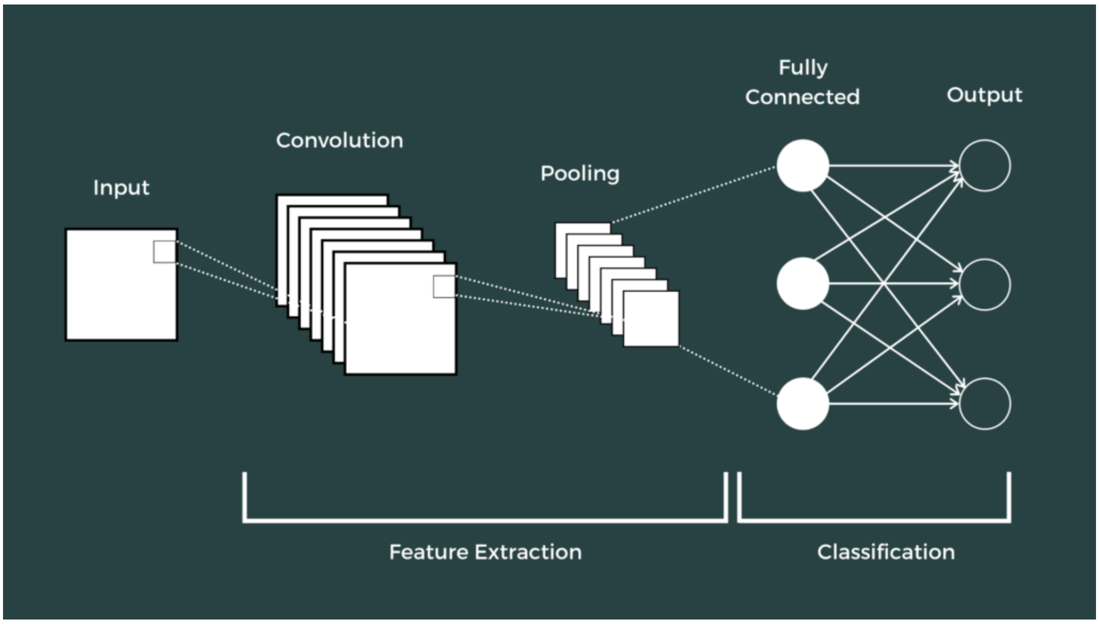

Figure 3.3: Convolutional Neural Network

The implementation uses a CNN model architecture with the following characteristics —

- The model contains four Convolutional Layers with 128, 64, 32, and 32 filters, respectively.
- The convolutional layers contain filters with varying filter sizes.
- A MaxPool2D layer follows each convolutional layer.
- Two Fully Connected layers follow the convolutional block.

3.3.2 Support Vector Machine (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as

Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

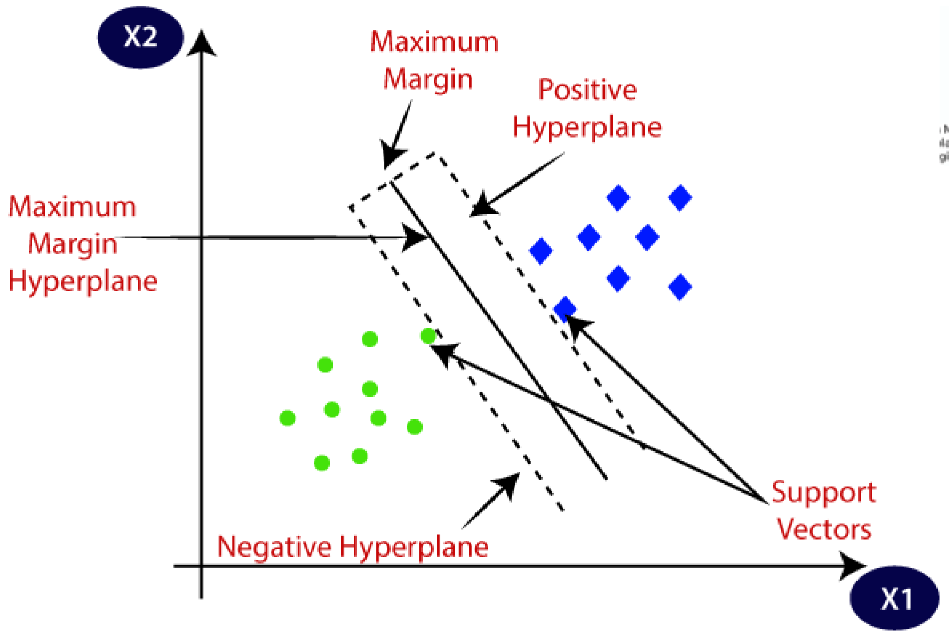

Figure 3.4

SVM has found it's applications in a wide range of classification tasks. In particular, SVM has demonstrated excellent performance on many medical diagnosis tasks. However, there is still much room for improvement of the SVM classifier. Because it has been proved that proper model parameters setting can improve the SVM classification accuracy substantially . Values of parameters such as penalty parameter and the kernel parameter of the kernel function should be carefully chosen in advance when SVM is applied to the practical problems.

3.4 DATA VISULAIZATION

Data visualization is the representation of data or information in a graph, chart, or other visual formats. It communicates the relationships of the data with images. Data visualization is used in a large number of areas in statistics and machine learning. The process of transforming large data sets into a statistical and graphical representation. It is an essential task of data science and knowledge discovery techniques to make data less confusing and more accessible. Visualization takes a huge complex amount of data to represent charts or graphs for quick information to absorb and better understand ability. It avoids hesitation on large data sets table to hold audience interest longer.

Below visual(fig:3.5) shows the data from the voice dataset.The X-axis contains (0,1) .that is "0" for Healthy and "1" for Parkinson's. The Y-axis contains the number of people

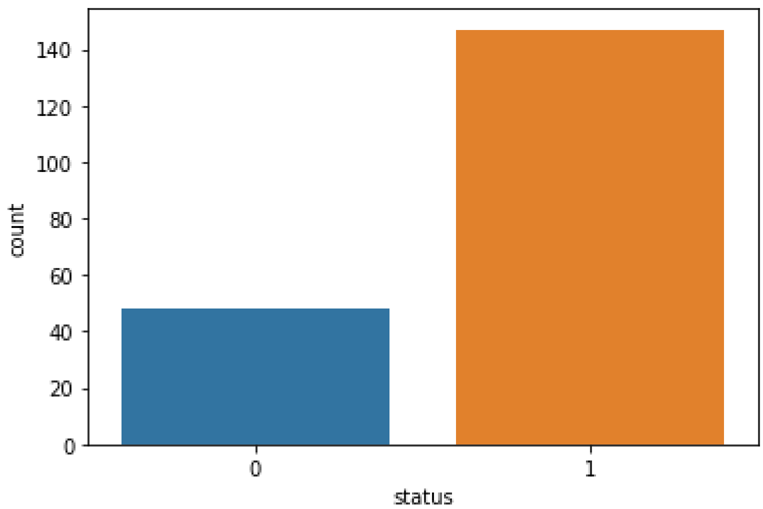

Figure 3.5: Data visualization of voice data set

The below Figure(fig:3.6) is the visualization of number of training images per category.It contains 35 patients and 35 healthy people data.The x-axis contains the status (parkinson's and healthy).The y-axis contains count(number of images).

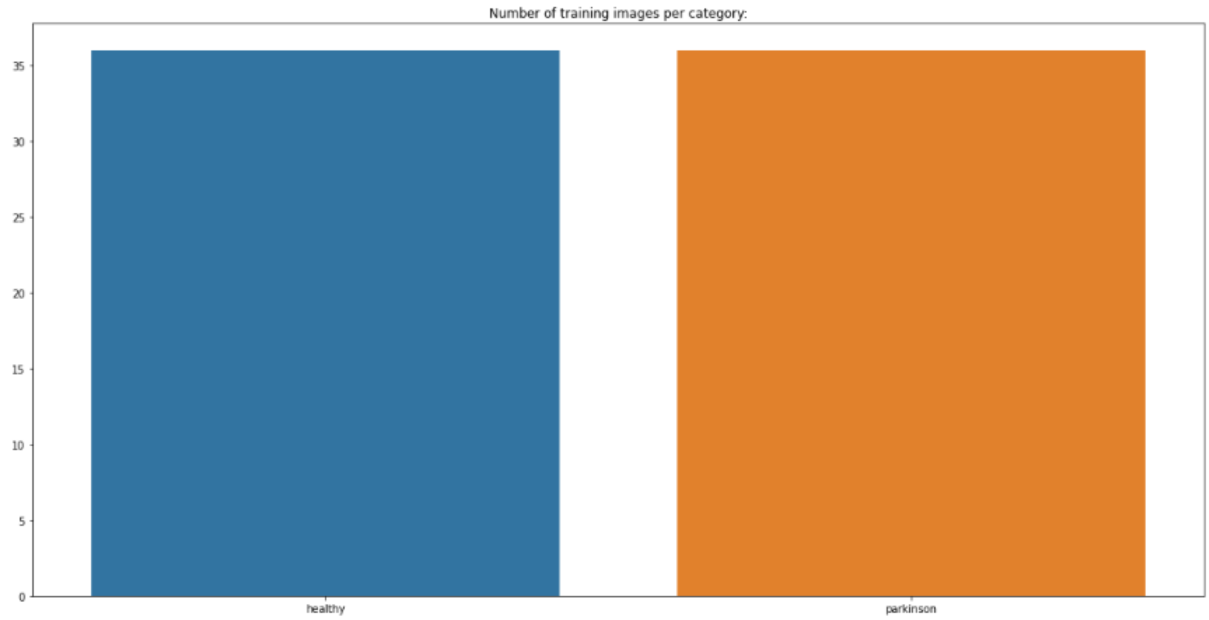

Figure 3.6: Data visualization of Number of trainig images per catagery set in Spiral Analysis

The below Figure(fig:3.7) is the visualization of number of testing images per category.It contains 3h patients and 35 healthy people data.The x-axis contains parkinson’s and healthy .The y-axis contains number of images.

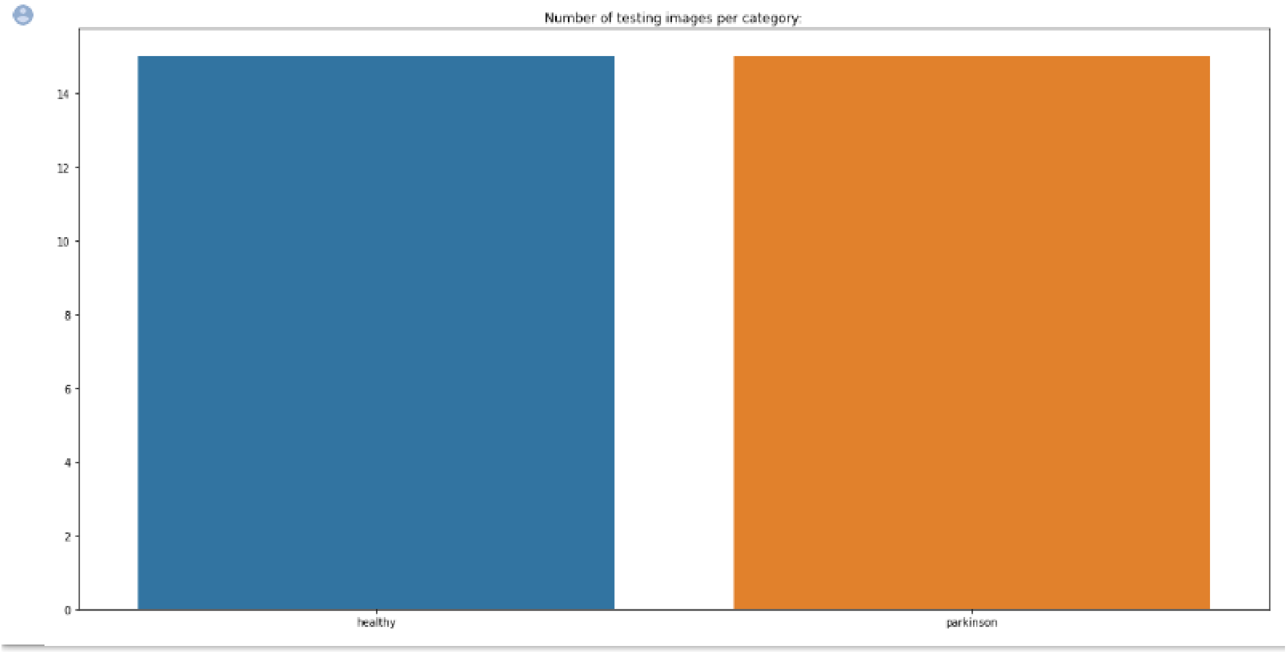

Figure 3.7: Data visualization of Number of Testing images per catagery set in Spiral Analysis

Model Accuracy

Model accuracy is defined as the number of classifications a model correctly predicts divided by the total number of predictions made. It's a way of assessing the performance of a model, but certainly not the only way.

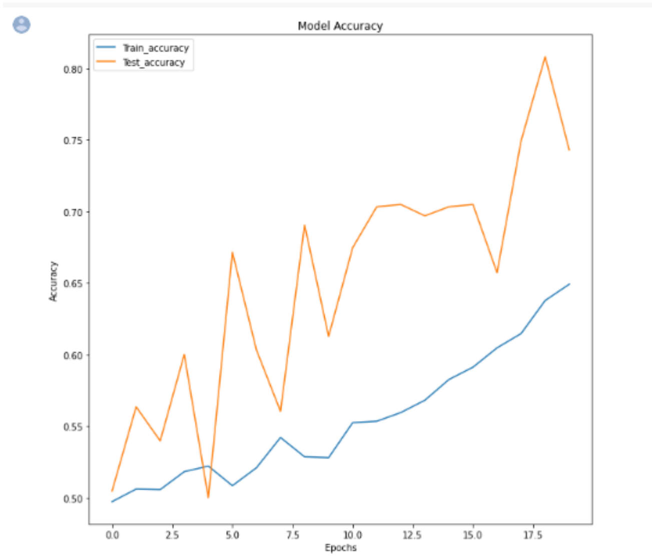

Figure 3.8: Model accuracy

Model Loss

That is, loss is a number indicating how bad the model's prediction was on a single example. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples.

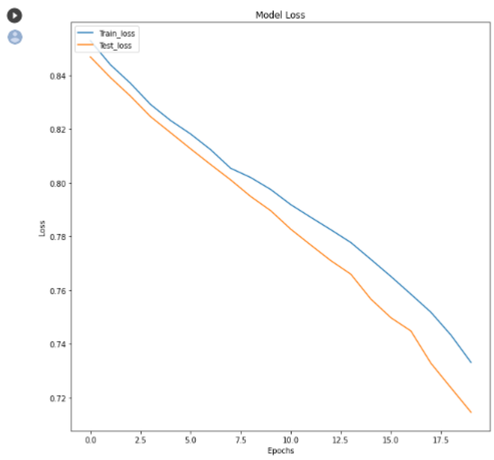

Figure 3.9: Model Loss

3.5 ACTIVITY DIAGRAM

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

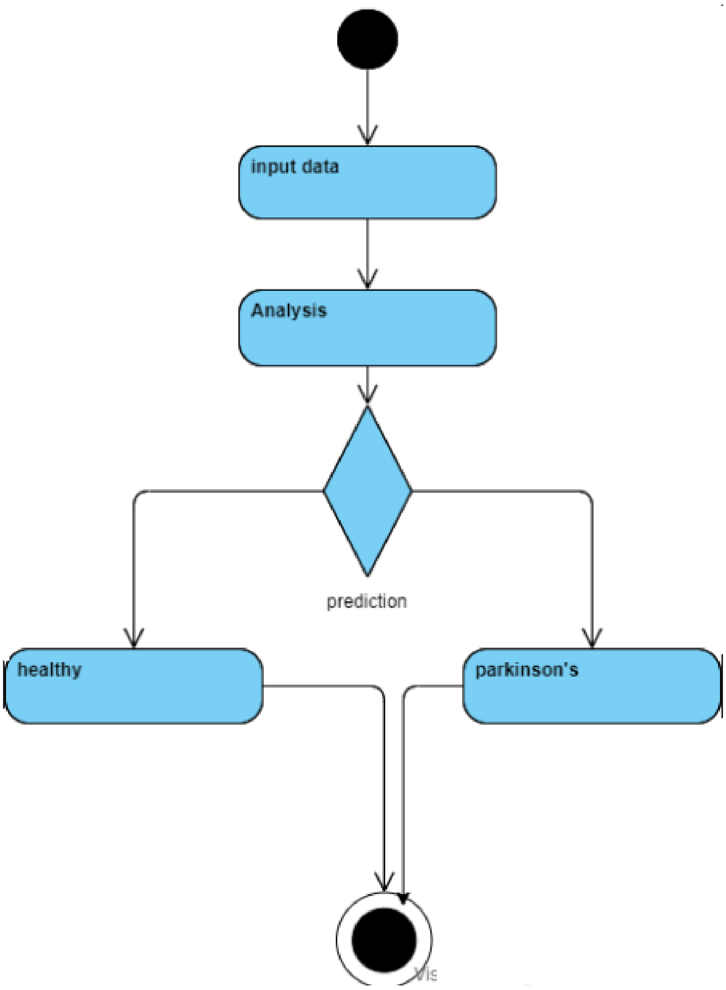

Figure 3.10: activity diagram

3.6 USE CASE DIAGRAM

A use case chart is a diagram for condensing information about a system and the users inside it.It is often displayed as a visual representation of how various system components interact with one another.Use case diagrams will detail the system's events and the order in which they occur. The scope and high-level functions of a system are described in use-case diagrams.The interactions between the system and its actors are also depicted in these diagrams.Utilize-case diagrams show what the system does and how the actors use it, but they do not show how the system works within.

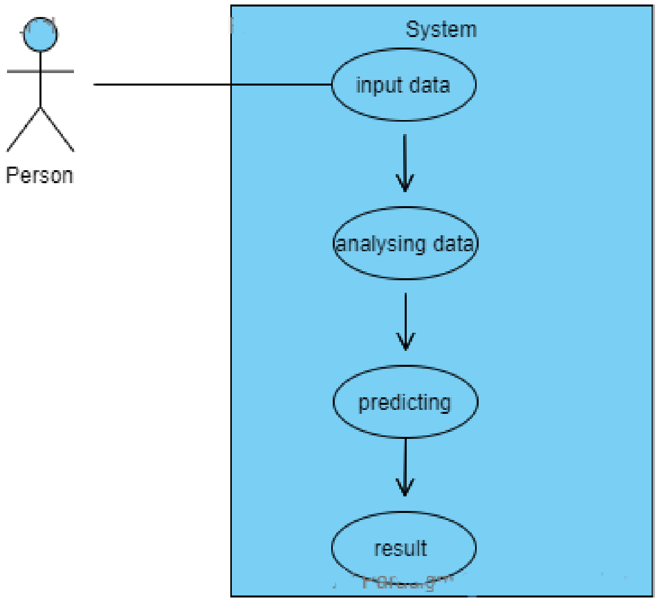

Figure 3.11: Use case diagram

3.7 BLOCK DIAGRAM

Block diagrams derive their name from the rectangular elements found in this type of diagram. They are used to describe hardware and software systems as well as to represent processes. Block diagrams are described and defined according to their function and structure as well as their relationship with other blocks.

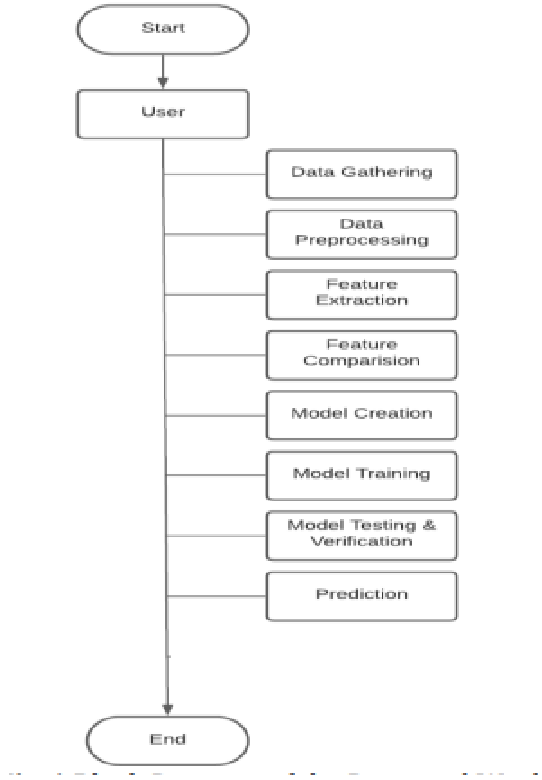

Figure 3.12: Block diagram

3.8 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

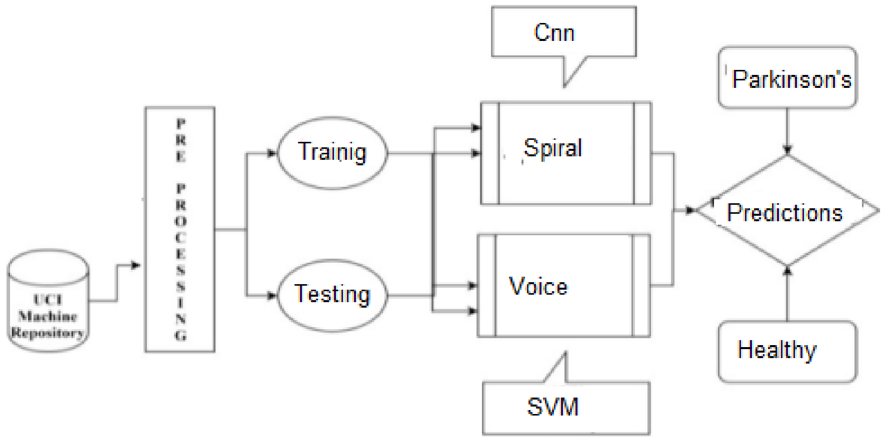

Figure 3.13: Class diagram

3.9 DATA FLOW DIAGRAM

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.

Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

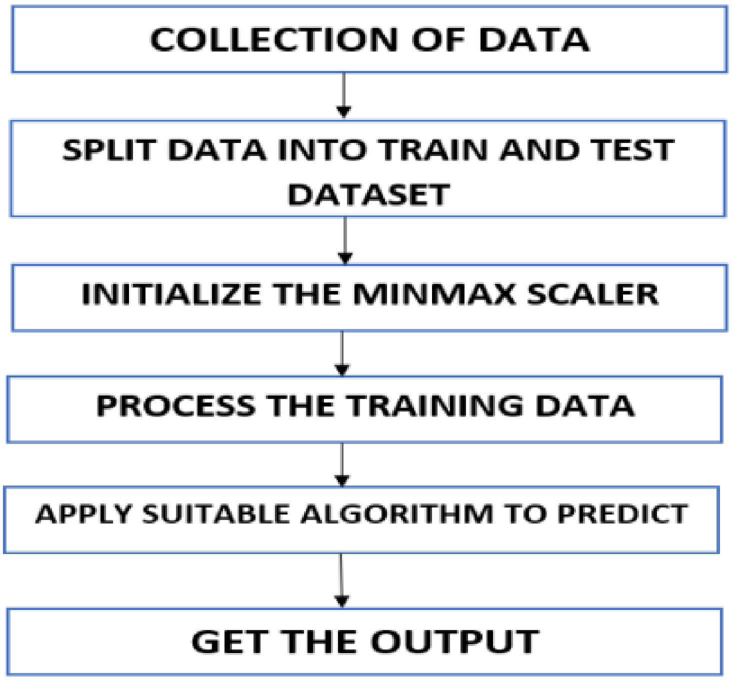

Figure 3.14: Data Flow diagram

3.10 PROPOSED WORK

the proposed Work used to predict parkinson's disease using patients voice and spiral drawings. For experimental results, we used Jupiter based python script for analysis. In the proposed Work, we proposed a hybrid architecture with voice and spiral drawing dataset information analysis to predict Parkinson disease in an easy manner and feasible one. The proposed predictive analytics framework is a combination of SVM and CNN to predict PD from the patients

The voice dataset is been downloaded from the website.By using SVM,the problem can be solved with minimal error rate. Parkinson's disease voice dataset from Machine learning repository is used as input. Thus our experimental results will show early detection of disease will facilitate clinical monitoring of elderly people and increase the chances of their life span and improved lifestyle to lead peaceful life

For processing the spiral images, we used Jupyter based python language for data analysis. Our proposed system provides accurate results by integrating spiral drawing inputs of normal and parkinson's affected patients. From these drawings we use CNN algorithm for feature extraction from the spiral drawings. The extracted values are been matched with the trained data and results are produced.

3.11 PROJECT PIPELINE

A machine learning pipeline is the end-to-end construct that orchestrates the flow of data into, and output from, a machine learning model (or set of multiple models). It includes raw data input, features, outputs, the machine learning model and model parameters, and prediction outputs.

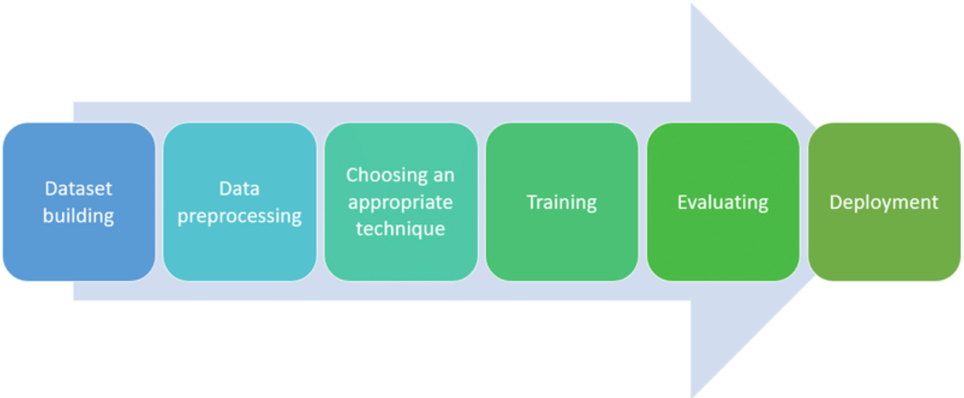

Figure 3.15: Pipeline

The project workflow basically flows in six major steps. They are listed below

- DataSet Building
- Data Pre-processing
- Techniques or Algorithms
- Training
- Testing or Evaluation
- Deployment or Predict

3.12 FEASIBILITY STUDY

Feasibility study is conducted to determine whether the proposed system is feasible. Now it is running manual system, in which all processing is done manually. Lots of stress and manpower is needed here. Implementing the proposed system can eliminate failures of the current system, which saves money and time. Thus the proposed system is economically feasible. The new system can support in the later versions of windows operating system. So the proposed system is technically feasible.

A feasibility report is a paper that examines a proposed solution and evaluates whether it is possible, given certain constraints. It includes six sections: introduction, background information, requirements, evaluation, conclusions, and finally, the recommendation or final opinion section.

The main purpose of a feasibility study is to assess the financial viability of developed land and whether it will be a success or failure. The study aims to highlight problems and risks that a potential plot will face such as gaining planning, local authority, assessing current development supply and demand and the suitability of a site, neighbours and so forth.

3.12.1 ECONOMIC FEASIBILITY

Economic feasibility attempts to measure the cost of developing and implementing a new system, against the benefits that accrue from having the new system in place, this feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in services quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

3.12.2 OPERATIONAL FEASIBILITY

Proposed project is beneficial only if it can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Here are questions that will help test the operational feasibility of a project:

Is there sufficient support for the project from management from users ?if the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance are the current business methods acceptable to the user. If they are not, users may welcome a change that will bring about a more operational and useful systems. Have the user been involved in the planning and development of the project.

Early involvement reduces the chance of resistance to the system and in general and increases the likelihood of successful project. Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

3.12.3 TECHNICAL FEASIBILITY

Evaluating the technical feasibility is the trickiest part of the feasibility. This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project.

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed systems. It is the evaluation of the hardware and software and how it meets the need of the proposed system.

3.13 SYSTEM ENVIRONMENT

Environment refers to the collection of hardware and software tools a system developer uses to build software systems. As technology improves and user expectations grow, an environment's functionality tends to change. User requirements for environments cover a broad spectrum. The functionality of environments includes support for a single user for programming the small, coordination and management of multiple users for programming-in-the-large, and management of the software development cycle. The system developer uses a collection of hardware and software tools to build software systems. The nature of the user interface is of considerable importance. Undoubtedly, the user of an environment needs to be able to customize it, either by tailoring or extending a particular tool or by creating specialized tools via generation facilities. To support this, the environment must be implemented so as to allow tools to be easily integrated into it.

3.13.1 SOFTWARE ENVIRONMENT

software environment is a collection of programs, libraries, and utilities that allow users to perform specific tasks. Software environments are often used by programmers to develop applications or run existing ones. A software environment for a particular application could include the operating system, the database system, specific development tools, or compilers. So in the development of the system there used some environments like:

- VS code
- Python 3
- jupyter / collab

All the listed software environments helped to build the system in a accurate manner. Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Python. Here VScode have been used as a source code editor.

The benefits of making Python the perfect solution for this project includes implicity and consistency,flexibility,access to powerful machine learning(ML) libraries and frameworks, platform independence, and large communities. These things increase the popularity of the language. Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Machine learning requires continuous data processing, and Python libraries allow you to access, process, and transform your data. These are some of the most extensive libraries available for ML. Scikit-learn to handle basic ML algorithms such as clustering, logistic and linear regression, regression, and classification. Python is easy to use,learn,and it is versatile too. It means that Python, which is used to develop machine learning, can run on all platforms. Developers can use software packages such as PyInstaller to prepare code to run on different platforms. That saves time and money on testing across other platforms and makes the process easier and more convenient.

3.13.2 HARDWARE ENVIRONMENT

The environment refers to a hardware platform and the operating system that is used in it. A programming environment would include the compiler and associated development tools. An operating system often implies the CPU hardware. It implies x86 because Windows has run on x86 machines for decades. However, it briefly ran on ARM CPUs with Windows RT, and starting in 2018, Windows once again was able to run on ARM (see Windows 10 on ARM). First of all to perform machine learning and deep learning on any dataset, the software program requires a computer system powerful enough to handle the computing power necessary. So the following is required:

- Central Processing Unit (CPU) — Intel Core i5 6th Generation processor or higher. An AMD equivalent processor will also be optimal.
- RAM — 8 GB minimum, 16 GB or higher is recommended.
- Graphics Processing Unit (GPU) — NVIDIA GeForce GTX 960 or higher. AMD GPUs are not able to perform deep learning regardless.
- Operating System — Ubuntu or Microsoft Windows 10.

Graphics performance is difficult to measure because of the widely varying complexity of visual scenes and the different hardware and software approaches to computing and displaying visual imagery. The most straightforward measure is given in terms of polygons/second, but this only gives a crude indication of the scene complexity that can be displayed at useful interactive update rates. Polygons are the most common building blocks for creating a graphic image.

There is no current graphics hardware that provides this, so we must make approximations at the moment. This means living with less detailed virtual worlds, perhaps via judicious use of hierarchical data structures or off-loading some of the graphics requirements by utilizing available CPU resources instead. For the foreseeable future, multiple process or work stations will be playing a role in off-loading graphics processing. While we focus on graphics initially, it is important to note that it is the way world modeling effects picture change that is of ultimate importance.

Chapter 4

SYSTEM DESIGN

4.1 MODEL BUILDING

The model building process involves setting up ways of collecting data, understanding and paying attention to what is important in the data to answer the questions you are asking, finding a statistical, mathematical or a simulation model to gain understanding and make predictions.
With the use of data mapping, the collected dataset is separated into two parts: 80 percent training data, and 20 percent testing data. In order to allocate data points to the former and the latter in the modelling dataset, the data has been separated into training and testing sets. A model is therefore trained using a training set, then applied to a test set. Our application may be evaluated in this manner.

4.1.1 PLANNING

Project planning is the process of defining the objectives and scope, goals and milestones, and assigning tasks and budgetary resources for each step. The main objectives of the project is to develop a system that utilizes machine learning techniques to easily predict the parkinsons's disease. The plan to build thesystem is by using some image processing libraries and ML algorithm. OPEN CV is one of the main library which is used here for Object detection and Image segmentation and recognitions. The key steps in a machine learning based diagnosis of parkinson's disease are: voice and image acquisition, preprocessing, segmentation, extraction of features and classification.

4.1.2 TRAINING

The process of training an ML model involves providing an ML algorithm (that is, the learning algorithm) with training data to learn from. The term ML model refers to the model artifact that is created by the training process.

The training data must contain the correct answer, which is known as a target or target attribute. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

You can use the ML model to get predictions on new data for which you do not know the target. For example, let's say that you want to train an ML model to predict if a person has Parkinson's disease or not . You would provide ML with training data that contains dataset for which you know the target (that is, a label that tells whether a person has parkinson's disease or not). ML would train an ML model by using this data, resulting in a model that attempts to predict whether he / she has Parkinson's disease or not

4.1.3 TESTING

One of the aspects of building a Machine Learning model is to check whether the data used for training and testing the model belong to an adversary dataset. The adversary data sets are that can be used to skew the results of the model by training the model using incorrect data called as Data Poisoning Attack. Quality Assurance put test mechanisms in place to validate whether the data used for training sanitized. The tests need to be performed to identify whether there are instances of data poisoning attacks unintentionally or intentionally.

4.2 SCREENSHOTS

Some of the output screenshots are given below :

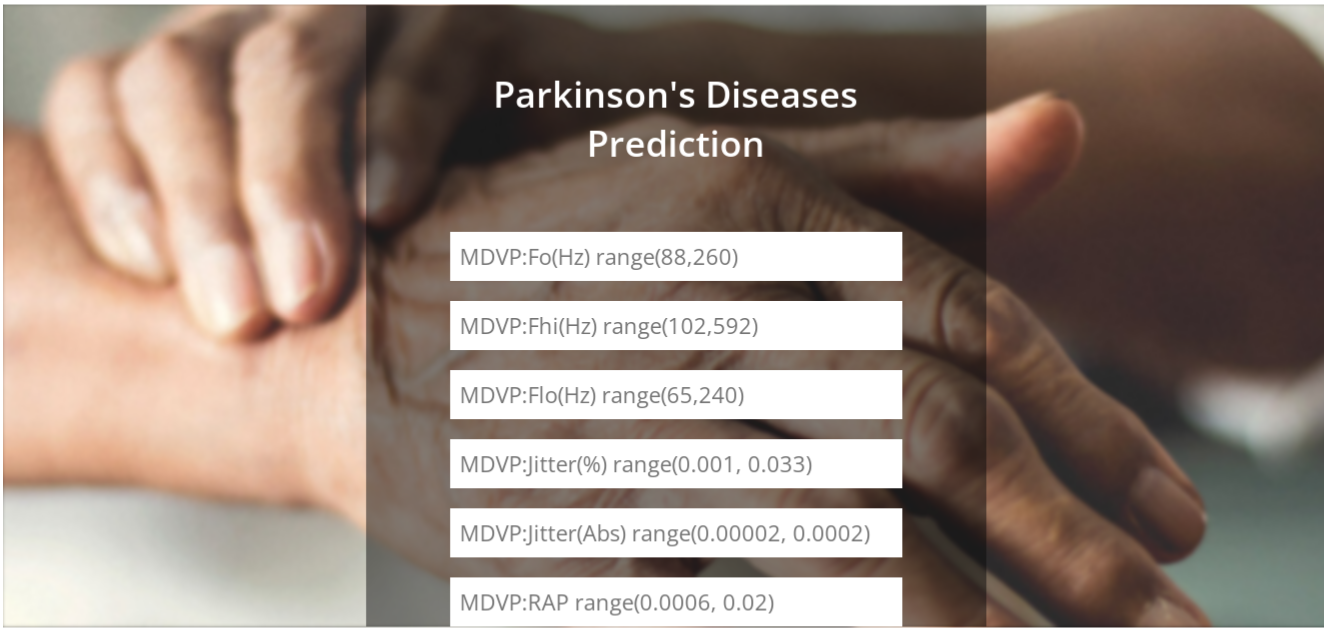

Figure 4.1: Home

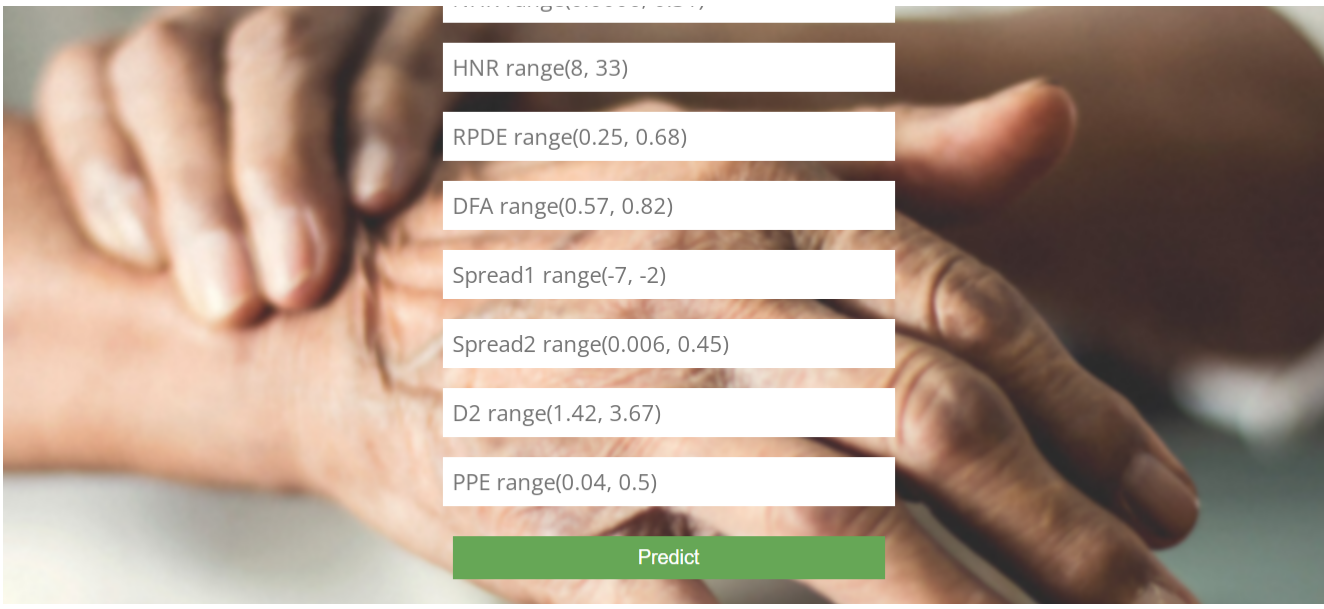

Figure 4.2: Home

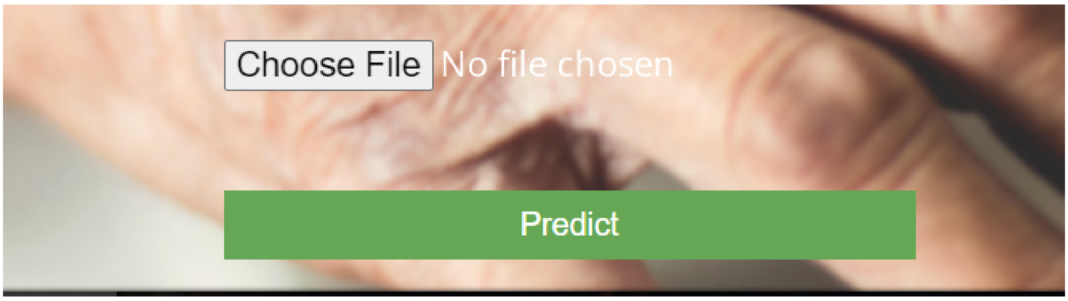

Figure 4.3: spiral input

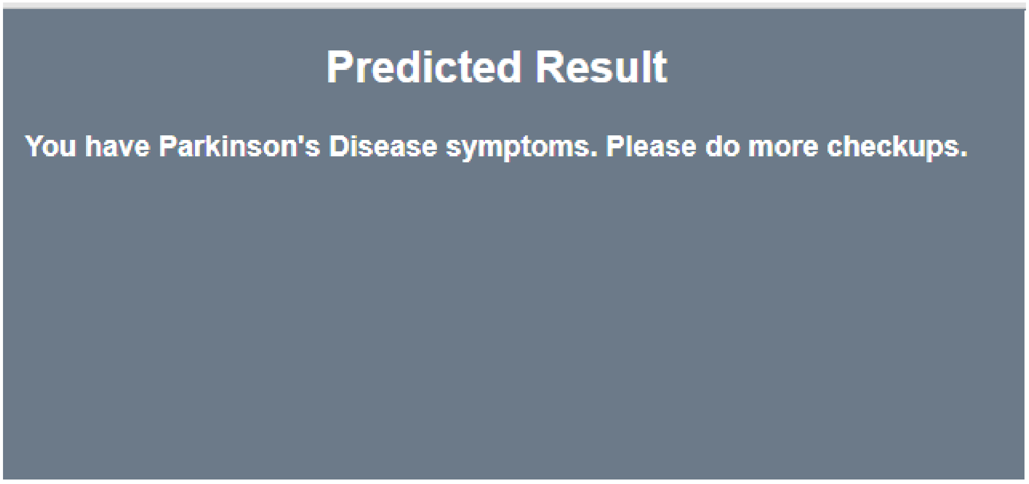

Figure 4.4: Result page

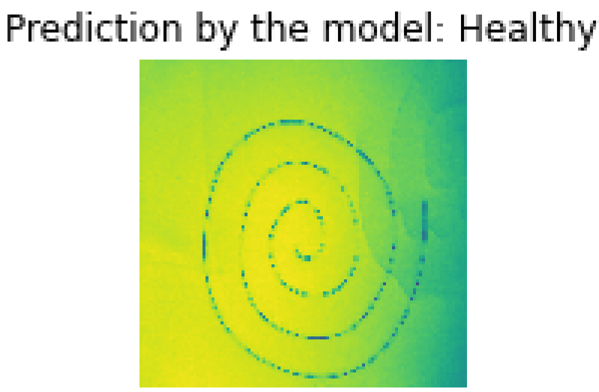

Figure 4.5: Spiral Output

Chapter 5

FINAL RESULT

The concepts of Machine Learning are discussed, while the application in Parkinson prediction is focused. The literature survey has been conducted on the Parkinson's disease. As the Parkinson's disease has dangerous implications and it has no proper cure.So, if it is detected at an early stage then we can easily treat; otherwise it becomes fatal. So by this situation the proposed system started to work on,As the system started building there come across a lot of challenges and limitations, somehow the system have finally executed and obtained the output which classifies and detect whether the patient have Parkinsons or not.

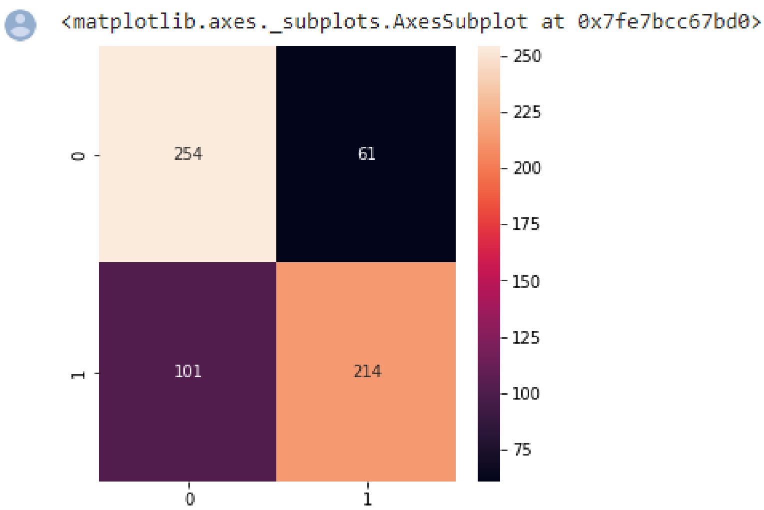

Figure 5.1: Confusion Matrix

Chapter 6

MODEL DEPLOYMENT

Model deployment is the process of putting machine learning models into production. This makes the model's predictions available to users, developers or systems, so they can make business decisions based on data, interact with their application (like recognize a face in an image) and so on.
The model deployment includes different steps to follow in order to build and deploy the system.

6.0.1 MAIN SOURCE CODE FOR VOICE ANALYSIS

importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
mport numpy as np
import pandas as pd
from sklearn.model selection import train test split
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.metrics import accuracy-score
```

Data Collection

```
parkinsons_data = pd.read.csv('/content/parkinsons.data')
# distribution of target Variable
parkinsons_data['status'].value_counts()
```

Data Pre-Processing

```
X = parkinsons_data.drop(columns=['name','status'], axis=1)
Y = parkinsons_data['status']
```

Splitting the data to training data And Test data

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random
```

Data Standardization

```
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Model Training (SVM)

```
model = svm.SVC(kernel='linear')
# training the SVM model with training data
model.fit(X_train, Y_train)
\textbf{Model Evaluation}
# accuracy score on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
# accuracy score on training data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
```

Writing different model files to file

```
with open( 'modelForPrediction.sav', 'wb') as f:
    pickle.dump(model,f)
with open('standardScalar.sav', 'wb') as f:
    pickle.dump(sc,f)
```

Prediction :-

```
 scaler = pickle.load(open('standardScalar.sav', 'rb'))
           prediction=loaded_model.predict(scaler.transform([[mdvp_fo,mdvp_fhi
               mdvp_rap,mdvp_ppq, jitter_ddp, mdvp_shim, mdvp_shim_db,shimm_ap
           print('prediction is', prediction)
           if prediction == 1:
               pred = "You have Parkinson's Disease. Please consult a speciali
           else:
               pred = "You are Healthy Person."
```

6.0.2 MAIN SOURCE CODE FOR SPIRAL ANALYSIS

Importing Libraries

```
import numpy as np
import cv2
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, Dropout,
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.optimizers import Adam, SGD
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
from tensorflow.keras.regularizers import l2
import tensorflow as tf
import pandas as pd
```

Data collection

```
data_train = np.load('train_set.npz', allow_pickle=True)
data_test = np.load('test_set.npz', allow_pickle=True)
```

Data Distribution of Train Set

```
unique_train, count = np.unique(y_train, return_counts=True)
plt.figure(figsize=(20, 10))
sns.barplot(unique_train, count).set_title("Number of training images per categ
plt.show()
```

Data Distribution of Test Set

```
unique_test, count_test = np.unique(y_test, return_counts=True)
plt.figure(figsize=(20, 10))
sns.barplot(unique_test, count_test).set_title("Number of testing images per ca
plt.show()
```

Augmenting the Dataset

```
train_data_generator = ImageDataGenerator(rotation_range=360,
                                          width_shift_range=0.0,
                                          height_shift_range=0.0,
                                          horizontal_flip=True,
                                          vertical_flip=True)
```

Visualizing the Images in Train and Test Set

```
figure1 = plt.figure(figsize=(5, 5))
idx_healthy = [i for (i, v) in enumerate(y_train) if v=='healthy']
img_healthy = x_train[idx_healthy[-1]]
plt.imshow(img_healthy)
plt.title('Spiral Drawing by a Healthy Person')
plt.axis('off')
plt.show()
```

Preprocessing the Images

```
for i in range(len(x_train)):
    img = x_train[i]
    img = cv2.resize(img, (128, 128))  # changing the size of images to (128, 1
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
    x_train[i] = img
for i in range(len(x_test)):
    img = x_test[i]
    img = cv2.resize(img, (128, 128))  # changing the size of images to (128, 1
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    x_test[i] = img
```

Defining the Model

```
    def parkinson_disease_detection_model(input_shape=(128, 128, 1)):
    regularizer = tf.keras.regularizers.l2(0.001)
    model = Sequential()
    model.add(Input(shape=input_shape))
    .
    .
    .
    model= parkinson_disease_detection_model(input_shape=(128, 128, 1))
    model.summary()
```

Training the Model

```
hist = model.fit(x_train, y_train, batch_size=128, epochs=20, validation_data=(
```

Confusion Matrix

```
matrix = confusion_matrix(y_test_pred, ypred)
```

textbfSaving the Model

```
model.save('parkinson_disease_detection.h5')
```

Testing Models

```
labels = ['Healthy', 'Parkinson']
image_healthy = cv2.imread('test_image_healthy.png')
image_parkinson = cv2.imread('test_image_parkinson.png')
ypred_healthy = model.predict(image_healthy)
ypred_parkinson = model.predict(image_parkinson)
```

Running the Web app With Flask

app.run(debug=True)

Uploaded project to Github

Used git to upload the local project or can just drag and drop the folder to an empty repository on GitHub. By clicking the “Upload files”button in the toolbar at the top of the file tree. Or,you can drag and drop files from your desktop on to the file tree.

Once you’ve added all the files you want to upload,you can commit them directly to your default branch or create a new branch and open a pull request. – Files are uploaded in a sequential manner in which all the code segments are added in a list of view which anyone who visit the repository can easily access the code because it publically available in github. – All of the design code have also added for design the system using the best possible way to match the code segment. The requirements and the essential libraries needed to be installed have been specifically mentioned in the Readme file in the repository.

Chapter 7

CONCLUSION

Parkinson's disease is among widespread age-related neuro degenerative diseases, early diagnosis of which is crucial in decreasing its development rate. The availability of data in this era has motivated scientists to use this data for their purposes, one of which to be medical purposes.

A variety of data is published for the objective of studying PD, including gait, handwriting, neuro imaging, and voice records. Using machine-learning algorithms, scientists have devoted their time, studying these data to predict the disease. In this research, we tried to review some studies devoted to PD using data and developed our models using vocal data. Processing vocal signals gives rise to applicable features. SVM is believed to be a practical model trained on this data. Our studies also support this belief.

Furthermore, we try to introduce the application of auto encoders for the purpose. Training auto encoders and using the encoder section for extracting a nonlinear combination of features is shown to be useful. Stacking the developed models also resulted in predictions that are more accurate and precise.

Chapter 8

FUTURE ENHANCEMENTS

In this study machine learning techniques are discussed. In future, the implementation of the proposed work is plan for the early detection of the Parkinson's disease using the machine learning techniques. In the future, different types of attributes are also plan to use for the classification of patients and also try to identify the different stages of Parkinson's disease. In future we can directly get the input value that is the voice features so we can overcome the disadvantage of typing long data

In future work, we can focus on different techniques to predict the Parkinson disease using different datasets. In this research, we using binary attribute (1- diseased patients, 0-non-diseased patients) for patient's classification. In the future we will use different types of attributes for the classification of patients and also identify the different st ages of Parkinson's disease.

In the future, this research aims to collect a dynamic dataset by utilizing an electronic pen-pad for samples. So that the handwriting motion and the number of frames required to complete one set of hand drawings can be evaluated. As PD debilitates the movement in the handwriting of patients. Moreover, this will make the diagnosis of PD more accurate and effective.

Chapter 9

APPENDIX

9.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The environment refers to a hardware platform and the operating system that is used in it. A programming environment would include the compiler and associated development tools. An operating system often implies the CPU hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible and sometimes incompatible hardware devices for a particular operating system or application. First of all to per form machine learning and deep learning on any dataset,the software program requires a computer system powerful enough to handle the computing power necessary. So the following is required:

- Processor : Intel Pentium Core i5 and above
- Primary Memory : 8GB RAM and above
- Storage : 320 GB hard disk and
- Operating system : Used windows 10
- Display : VGA Colour Monitor
- Key Board : Windows compatible
- Mouse : Windows compatible

- Input Device : Mouse, Keyboard
- Output Device : Monitor

9.2 Software Requirements

A Software Requirement Specification(SRS), a requirements specification for a software system, is a complete description of the behaviour of the system to be developed and may include a set of use cases that describes interactions the users will have with the software. In addition it also contains non-functional requirements. Nonfunctional requirements impose constraints on the design or implementation the software requirement specification document enlists all necessary requirements that are required for the project development. A software environment is a collection of programs, libraries, and utilities that allow users to perform specific tasks. Software environments are often used by programmers to develop applications or run existing ones. A software environment for a particular application could include the operating system, the database system, specific development tools, or compilers. To derive the requirements we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with the project team and customer.

- Language : Python
- IDE : VSCode , Collab
- Connection: Flask

All the listed software environments helped to build the system in a accurate manner. Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Python.

REFERENCES

[1] A Detection Support System for Parkinson's Disease Diagnosis Using Classification and Regression Tree *AHadjahamadi, A.H. and Askari, T.J*

[2] A step towards the automated diagnosis of Parkinsons *Pereira, C.R*

[3] Detecting of Parkinson Disease through Voice Signal Features *Alemami, Y. and Almazaydeh, L*

[4] A comparison of multiple classification methods for diagnosis of Parkinson disease.Expert Systems with Applications *Das. R(2010)*

[5] Classification of Parkinson disease dat awith artificial neural networks *Yasar, Saritas, Sahman , Cinar(2012)*

[6] Prevalence of Parkinson's disease across North America *C. Marras et al(2018)*

[7] Early Detection of Parkinson's Disease through Patient Questionnaire and Predictive Modelling *Prashanth R, Dutta Roy S*

[8] Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks. *Adrien Payan, Giovanni Montana*

[9] https://www.kaggle.com/margot234/parkinsonsdisease-dataset

[10] https://www.analyticsvidhya.com/blog/2021/07/parkinson-disease-onset-detection-using-machine-learning/Parkinson's.