Detection Of Parkinson's Disease using Machine Language

Nalaiya Thiran

Professional Readiness for Innovation, Employability & Entrepreneurship

A Project Report

Submitted by

LATHA.R (311419106012)
ARUN BALAJI . T(311419106003)
MD YUSUF FARAAZ .V (311419106014)
POORANI.S (311419106020)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

MEENAKSHI COLLEGE OF ENGINEERING,

WEST K.K. NAGAR

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER 2022

FACULTY MENTOR

R.ANANDHA PRABA

FACULTY EVALUATOR

D.SATHEESWARI

DETECTING OF PARKINSON'S DISEASE USING MACHINE LEARNING.

INTRODUCTION

Machine learning techniques are being increasingly applied in the healthcare sector. As the name implies, machine learning allows for a computer program to learn and extract meaningful representation from data in a semiautomatic manner. For the diagnosis of PD Machine learning. models have been applied to multimode of data. modalities, including handwritten patterns, movement neuroimaging, voice, cerebrospinal fluid, cardiac scintigraphy, serum and optical coherence tomography. Machine learning also allows for combining different modalities, such as magnetic resonance imaging (MRI) and single photon emission computed tomography. (SPECT) data in the diagnosis of PD. By using machine learning approaches, we may therefore identify. relevant features that are not traditionally used. in the clinical diagnose of PD and rely on these measures to detect po in preclinical stages or atypical forms.

LITERATURE REVIEW:

[1] Early detection of Parkinson's Disease using deep learning and machine learning. Accurately detecting Parkinson's disease (PD) at an early stage is certainly indispensable for slowing down its progress and Providing patients the possibility of accessing to disease modifying therapy.

Advantage: Early detection of PD is an important step to understand the causes, develop better treatments and carry out effective early management of the disease.
Disadvantage: Even though deep learning offers superior performance compared to machine learning models, it is hard to say that the deep learning dominates the others.

[2] High-Accuracy detection of Early Parkinson's disease. through Multimodal features and machine learning. TO perform clarification, we use the preclinic markers of non-motor features of RBD and Olfactory loss, CSF measurements and dopaminergic imaging features

Advantage: The dataset can be divided into 70% and it is used for training and the next 30% is used for testing
Disadvantage:
Larger the increase, higher the importance of the fracture out of bag error is estimated by comparing out of bag predicted responses and the observed responses for all observations used for training

[3]. Telemonitoring Parkinson's disease using machine learning by combining tremor and voice analysis This proposed system receives rest tremor and vowel phonation data acquired by smartphones with built in accelerometer and voice recorder sensors. The data are primarily collected from diagnosed PD patients and healthy people for briefing and optimizing machine learning models that exhibits higher performance.

Advantage: The proposed system can detect pp using. a cloud-based system for computation, data providing, and regular monitoring of voice and tremors samples captured by smartphones thus, this system. can be a solution for healthcare authorities to ensure the older population's accessibility to a better medical diagnose system in the developing countries especially in the pandemic situation tire could-19, when in-person monitoring i minimal.
Disadvantage: Not every people will have smartphones for consultation.

[4]. Early diagnosis of Parkinson's disease in brain. MRI using deep learning algorithm. In preceding year, deep learning algorithms. have accomplished striking performance in numerous. fields that include

computer vision NEP (Natural language processing) and speech recognition Many changes are being brought about by deep learning in other fields like engineering and biology

Advantage: We can generalize them to obtain optimum solutions to distinct problem making use of the consistent design.
Disadvantage: The main cause of Parkinson disease is unknow. It describes problems in selection et majority of the discriminative and precise features. that are essential for classification model building.

[5]. Parkinson's Detection from spiral and wave drawings using convolutional Neural Networks. A system design is proposed for analysing spiral drawing patterns and wave drawing pattern in patients suffering from Parkinson's disease and healthy subjects' identification of the correct biomatter with respect to particular heath issues and detection of the same is of paramount importance for the development of clinical decision support system.

Advantages: The complete model was trained on the data of 55 patients and have achieved an overall accuracy of 93.3% average recall of 94%. average precision of 93.5%.
Disadvantages: The major drawback of these kinds of diagnose needs proper interpretation of sketching and handwriting.

[6]. Detection of Parkinson's disease from handwriting wring deep learning
In this, an automatic clarification system for PD detection is developed based on online handwriting. Two deep- learning models, trained end-to-end, have been proposed for time series classification, namely the CNN and the CNN-BLSTM. For the CNN Model, two. different approaches were proposed to encode time series into images for the CNN-BLSTM Model the raw time series are directly used.

Advantage: With the spread of digitizing devices, it's now Possible to record sequences of measurements from handwriting tasks, provided by tablet and pen devices.
Disadvantage: These methods are expensive and need a high level of professional expertise.

[7.] Deep learning aided Parkinson's disease diagnose from Handwritten dynamics
In this, we cope with the problem of PD Identification by means of convolutional Neural Networks Basically, the idea is to model the handwritten dynamics as a time series, and to use it as an input to a CNN, which will be able to learn features that are used to distinguish healthy individuals from PD patients

Advantage: Section making techniques based on machine learning, might be the most fruitful ones to deal with PD
Disadvantage: Since the waiting ability & affected by Parkinson's disease, it is very usual to find such exams in hospitals only one a few works have considered them for automatic diagnostic purposes.

[8] Parkinson's Disease detection using voice and spiral drawing dataset.
Parkinson's speech dataset with different recordings was taken from the VCI machine learning repository. The dataset included 20 voice data with Parkinson's disease and 20 voice data from healthy individuals. The voice data was focussed on those vowels |a|, |o| and |u|.

Advantage: The main advantages of using voice and spiral drawing dataset are

- Highly efficient
- Accurate detection
- Real time implementation

Disadvantage: No other data except for voice and spiral are used, no other algorithm or scan reports are used

[9] It represents research effort that were undertaken to inform on how well traditional machine learning algorithm can handle this task. Thus, the primary objective of the research is to provide a literary foundation for development and improvement of algorithm for deducting PD related motor symptoms.

Advantages: We can easily know the symptoms by this. It exhibits PD- like phenotype and established behavioural test. Ease of genetic manipulation process.
Disadvantage: It relatively expensive and it has long life cycle.

[10] Dis-luminating progressive supranuclear palsy from Parkinson's Disease using machine Learning. Progressive supra nuclear palsy (PSP), a neurodegenerative condition may be difficult to discriminate clinically from idiopathic Parkinson's disease. Analysis of gait and relate tasks is one possible means of discrimination.

Advantage: Machine learning methods can accurately discriminate PSP from PD choice of array computing depend on content for diagnostic purpose a high specification is needed suggestion the more complete array advantage.
Disadvantage: We the explore the effect of adding one arm and one leg sensor to the lumber sensor making the three-sensor including lumber right arm and right foot.

Reference

[1]. Statistics program , computer electrical and mathematical sciences and engineering(CEMSE) Division, king Abdullah University of science and techjnology (KAUST), Thuwal 23955—6900 ,Saudi arabia Corresponding author:Fouzi Harrou (fouzi.harrou@Kaust.edu.sa)

[2]. R.Prashanth , Sumantra Dutta Roy , Pravat K .Mandal, and Shantanu Ghosh. Department of Electrical Engineering , Indian Institute of Technology Delhi,India

[3]. Md.sakibur Rahman Sajal, Md Tanvir Ehsan , Ravi Vaiudyanathan , Shouyan Wang , Tipu Aziz and Khondaker Abdullah Al Mamun. Published 2020

[4]. Sabyasachi Chakraborty, Satyabrata Aich, jong -scong -sim, Eunyoung Han, jinse park**, Hoe -C HEOL Kim*. *Department of computer engineering/Institute of digital Anti-Aging Healthcare/u-HARC, Inje university, south Korea **DEPARTMENT OF NEUROLOGY, Haneda Paik Hospital, inje University, south Korea

[5].Ayush goyal department of electrical engineering and computer science . Frank. H.Datterweich college of engineering, Texas A&M university Kingsville, united states

[6]. D. john Aravindhar prof.CSE Hindustan institute of technology and science Chennai ,India jaravindhar@hindustanuniv.ac.in

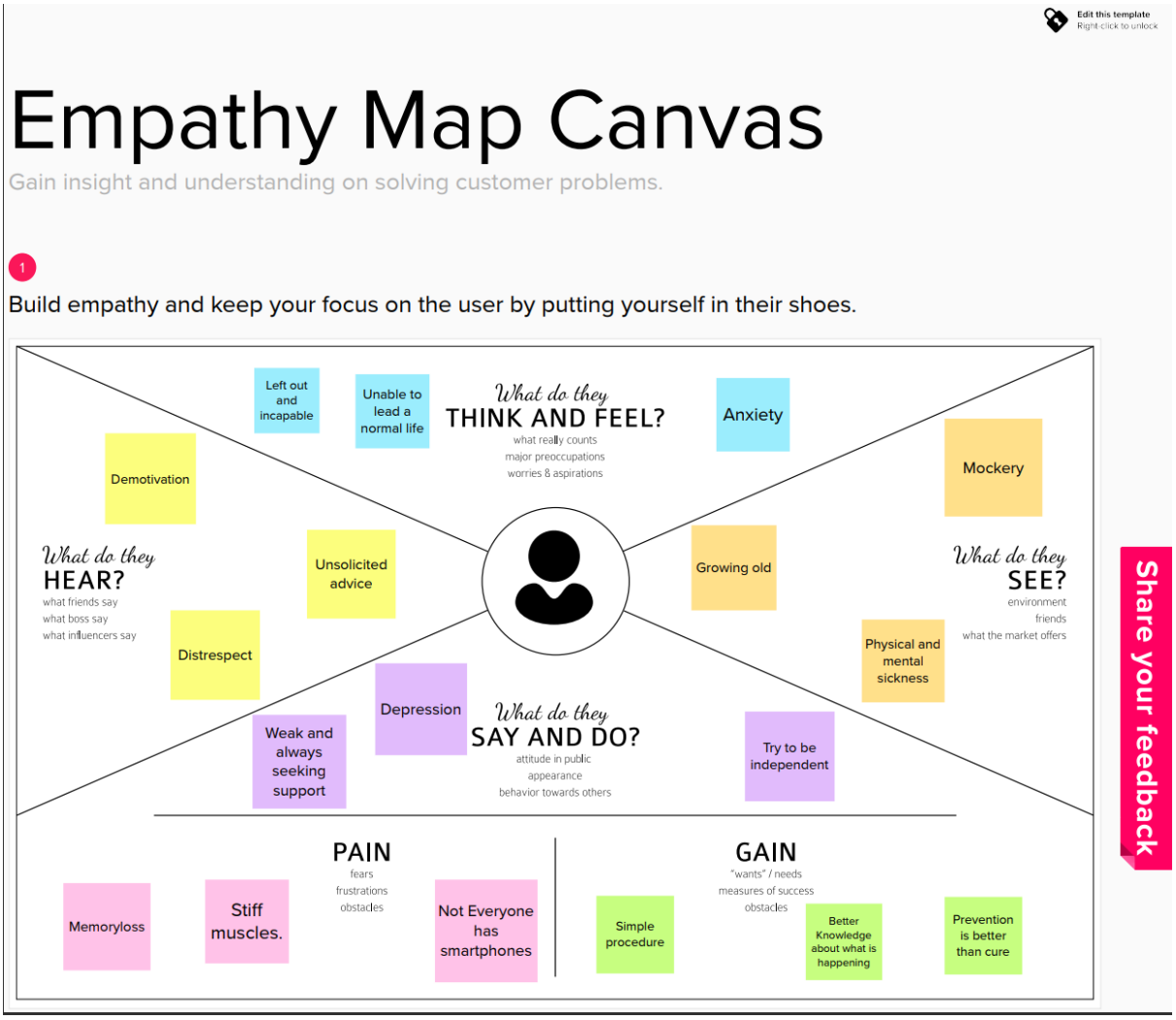
[7]. Christian hook fakultat informatk/mathematic ostbayerische tech hochschule Regensburg,Germany Christian.hook@hs-regensburg.de

[8]. Atiienza R (2017)LSTM by example using tensor -flow accessed 3 July 2019

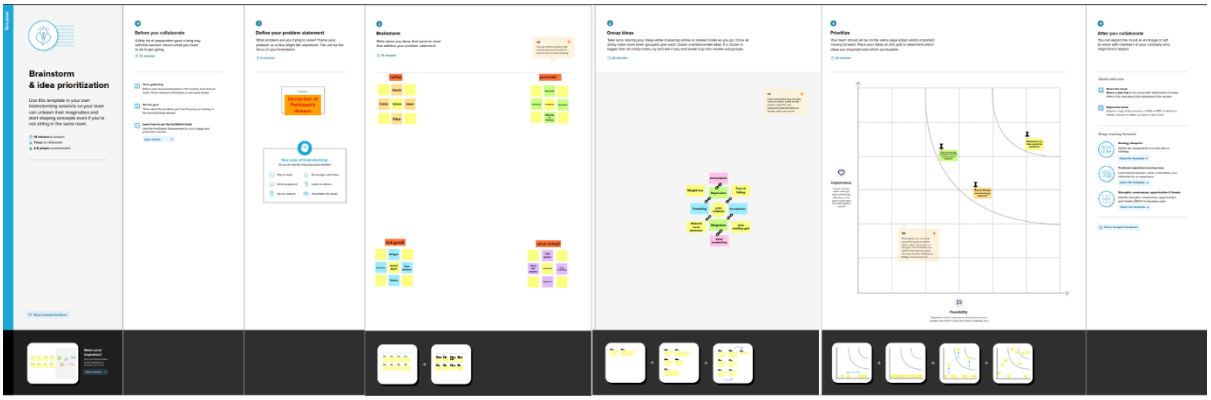
[9]. Mathematics and computer science. University of Bremen PO box 330-440,28334, Brenmen, Germany

[10]. Maarten De Vos, john prince, Tim Buchnan Ph.D., James j fitzGerald Ph.D., and Chrystalina A Antoniades Ph.D. Department of engineering science, institute of biomedical engineering, university of oxford, old road campus research building, OX3 7DQ, oxford, UK

IDEATION & PROPOSED SOLUTION



Ideation & Brainstorming

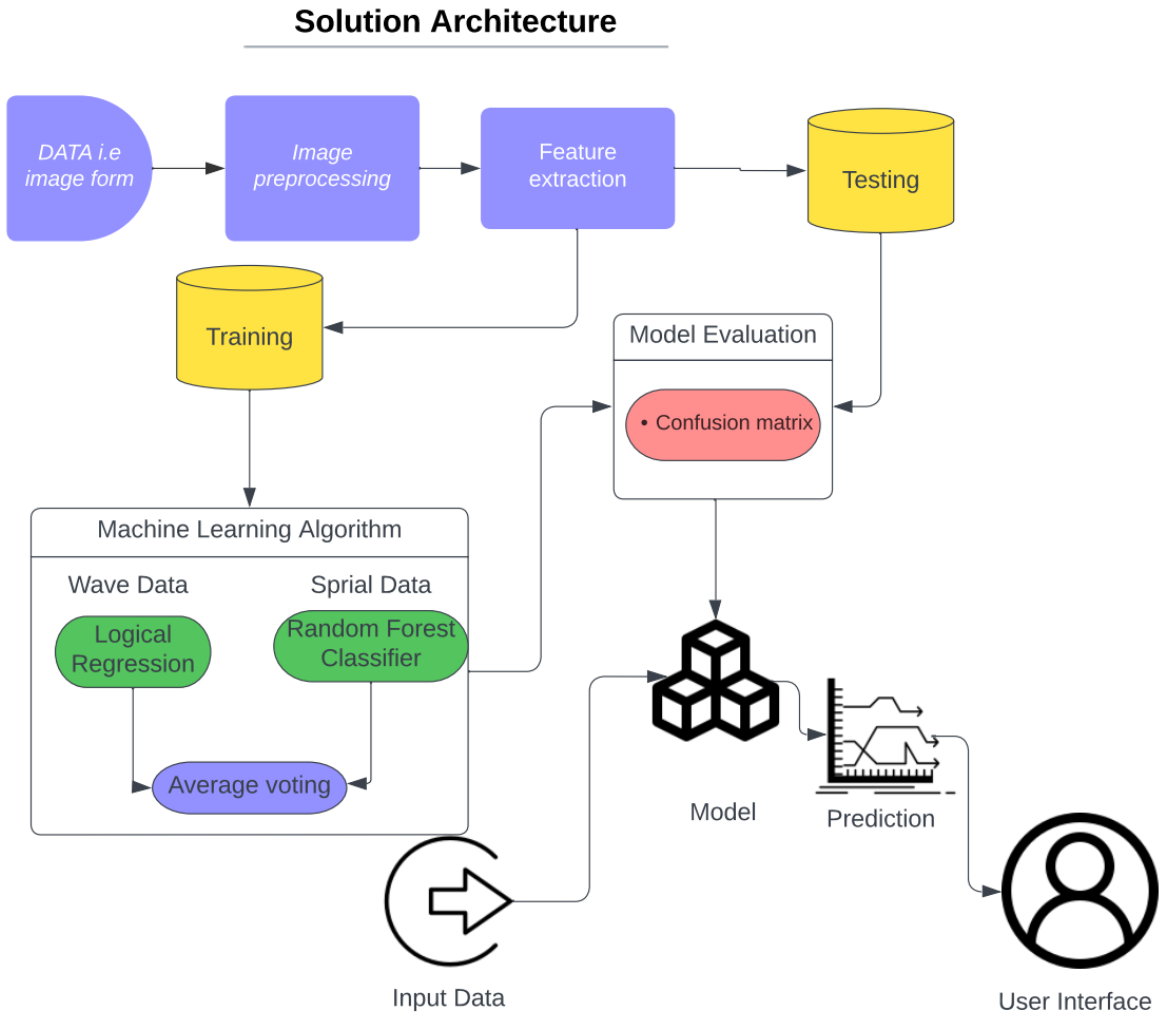


Problem Statement:

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Detection of Parkinson's Disease

2.	Idea / Solution description	Using handwriting / Movement factor and implementing it using Machine language.
3.	Novelty / Uniqueness	It is a very simple process which doesn't require any prior knowledge and can be used by everyone
4.	Social Impact / Customer Satisfaction	The easier process makes gives complete satisfaction to the user with atmost accurate results
5.	Business Model (Revenue Model)	It is a low cost model. Hence its cost efficient
6.	Scalability of the Solution	It is a very easy process which can be accessed by almost everyone.



Problem Solution Fit:

Project Title: Early Detection of Parkinson's disease | **Project Design Phase-I - Solution Fit Template** | **Team ID:** PNT2022TMID27729

Define CS, fit into CC

1.CUSTOMER SEGMENT(S) CS

Senior citizens

6. CUSTOMER CONSTRAINTS CC

No Proper awareness about the approach

5. AVAILABLE SOLUTIONS AS

->SPECT scan

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS JP

Physical and Mental illness.Physical illness such as shivering , bad posture and mental illness such as memory loss

9. PROBLEM ROOT CAUSE RC

A loss of nerve cells in the part of the brain

7. BEHAVIOUR BE

The customer uses various applications such as ours to detect if he has Parkinsons disease

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

Doctors believe that environmental causes may help trigger Parkinson's disease. Exposure to chemicals, like pesticides .Working with heavy metals, detergents and solvents have the affects.

4. EMOTIONS: BEFORE / AFTER EM

BEFORE: Patients may experience a range of mental health issues. These can range from depression and anxiety to hallucinations and memory problems Anxiety and depression are most common mental health symptoms

AFTER: Symptoms begin gradually and gets worse over time. As the disease progresses, patient have difficulty in walking and talking. They may also have mental and behavioral changes, sleep problems, depression and memory difficulties.

10. YOUR SOLUTION SL

There are many ways people living with Parkinson's disease can improve their health and well-being, preserve physical function, symptoms and enhance quality of life. Along these things they are getting regular exercise, eating a healthy diet, staying hydrated and getting an good amount of sleep.

8.CHANNELS of BEHAVIOR CH

8.1 ONLINE

People with Parkinson's disease can find local support groups, educational programs, health and wellness activities, and events using the search tool on the ASDA's website. The Parkinson's Foundation also offers information about local resources, support groups, wellness classes, and educational programs.

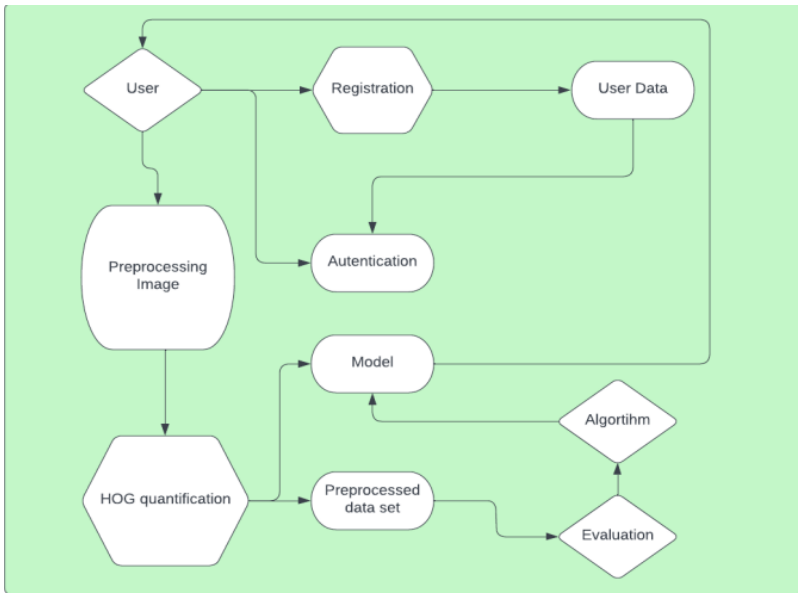
8.2 OFFLINE

There is a better way. FCP Live-In can ensure your loved one receives quality, one-on-one Parkinson's disease care at home, where they already feel comfortable and secure.

Identify strong TR & EM

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

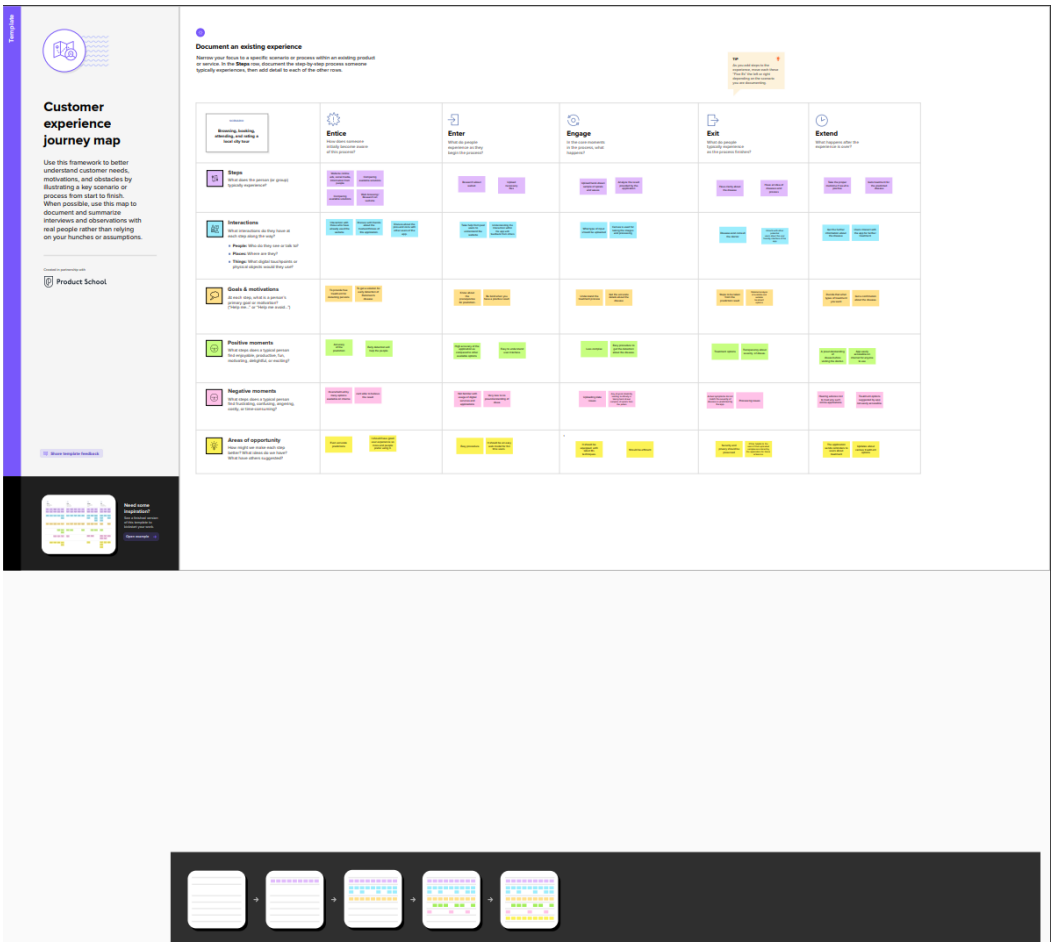


User Stories:

Use the below template to list all the user stories for the product

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Public user)	Account creation	USN-1	As a user, I can connect my google into the application	I can access my account / application dashboard	High	Sprint-1
Input data	Adding data	USN-2	As a user, I can feed my data as the input into the application for it to classify the true fake data	I can cross verify the data that entered in the initial step	High	Sprint-1
Data validation	Checking accuracy	USN-3	As a user, I can check the ability and accuracy of the model in obtaining the required information	I can log into my account and check the capability of the model	Medium	Sprint-2
Classification	Data classification	USN-4	As a user, I can view the real data	I can verify my data with the real data	Medium	Sprint-2
App work	Work flow	USN-5	As a user, I can examine the working action of the application model	I can view how the application works and responds to the actions imposed	High	Sprint-2
Image classification	Checking for the disease	USN-6	As a user, I can verify with the application that the image is identified with the actual disease with the help of the trained and tested data's	I can confirm that the data shows the accurate result	Low	Sprint-3
User interaction	AI-powered chatbot	USN-7	As a user, I can interact with the automated chatbot to engage my time till the application processed the accurate result in a meanwhile	I can see the results from the interaction with the chatbot	Low	Sprint-3
Medical assistance	Medical suggestions	USN-8	As a user, I can get medical advises and recommendations for to boost the action of curing the disease	I can get enough assistance by getting the suggestions for curing the disease	High	Sprint-3
Data extraction	Obtaining the data	USN-9	As a user, I can retrieve the result data from the application for data storage for further medical research uses.	I can download the result in the form of data as a proof to show to medical teams	Medium	Sprint-4

Customer experience journey map:



Functional Requirements:

Following are the functional requirements of the proposed solution:

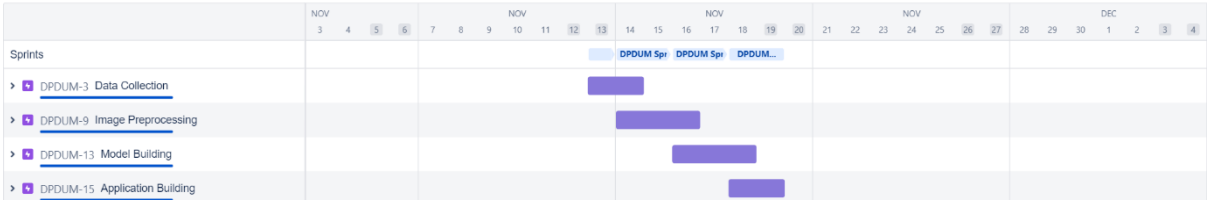
FR No.	Functional Requirement(Epic)	Sub requirements (Story/ Sub-Task
FR-1	User Registration	❖ Registration through Form ❖ Registration through Gmail ❖ Registration through LinkedIN
FR-2	User Configuration	❖ Confirmation via Email ❖ Confirmation via OTP
FR-3	User Interface	❖ Admin Login Form ❖ Graphical user interface for non-computer literate PD patients
FR-4	Database	❖ Gene4PD
FR-5	Operating System	❖ Windows 10
FR-6	Software	❖ Spyder ❖ Google collab
FR-7	Tools	❖ Web Browser
FR-8	Python Libraries	❖ Numpy, pandas, matplotlib, seaborn, sklearn,
FR-9	Detecting Parkinson's disease	❖ Identifying rare variants from PD patients using images such as Waves and Spiral
FR-10	Input File Format	❖ png
FR-11	Supported Browser	❖ Google Chrome ❖ Firefox ❖ Microsoft Edge
FR-12	Algorithm	❖ Logical Regression ❖ Random Forest Classifier

Non-Functional Requirements:
Following are the non-functional requirements of the proposed solution

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	❖ Easy-to-use
NFR-2	Security	❖ IBM Cloud offers Data security service which includes Data discovery, Data activity and encryption
NFR-3	Reliability	❖ Easy to use ❖ Portable ❖ Flexibility
NFR-4	Performance	❖ High-performance and Productivity for users
NFR-5	Scalability	❖ Perfectly scalable to handle workload and how agile it is at accomplishing that task

Road Map:

Weekly Progress



Monthly progress

USER STORIES:

Use the template to list all the stories in the product

USER TYPE	FUNCTIONAL REQUIREMENTS	USER STORY NO.	USER STORY / TASK	ACCEPTANCE CRITERIA	PRIORITY	RELEASE
USER (MOBILE)/ WEB USER	REGISTER	USN-1	As a user I can register by email, password and confirm password	I can access my account/ dashboard	HIGH	SPRINT-1
		USN-2	As a user get confirmation mail	I can get confirmation mail	HIGH	SPRINT-1
		USN-3	As a user, I can register by other sources	I can access by other	LOW	SPRINT-1
	LOGIN	USN-4	As a user, I can login by gmail	I can access the dashboard by gmail	HIGH	SPRINT-1
	PREDICTION	USN-7	As a user, I can enter my details for prediction	I can predict parkinson	HIGH	SPRINT-2
		USN-8	As a user, I can view the accuracy of the model	I can view accuracy	MEDIUM	SPRINT-3
		USN-9	As a user, I can verify it with others	I can compare it with others	LOW	SPRINT-3
	RESULTS	USN-10	As a user, I can get output for parkinson	I can check whether I have Parkinson or not	HIGH	SPRINT-4
		USN-11	As a user, I can decide to consult the doctor	I can decide to consult the doctor	MEDIUM	SPRINT-4
	LOGOUT	USN-12	As a user, I can logout from it	I can logout	HIGH	SPRINT-3
ADMIN	LOGIN/LOGOUT	USN-1,3	As a admin, I can login or logout	I can login or logout	HIGH	SPRINT-4
	ADMINISTRATION	USN-2	As a admin, I can oversee the model	I can overview	HIGH	SPRINT-4

Coding :

For app.py:-

```
from flask import Flask,redirect, request, url_for, render_template
import sqlite3 as sql

app=Flask(__name__)

class DB:
    def _init_(self, name = 0):
```

```
        self._name = name
        self._lid = 0
        self._tot=0

    # getter method
    def get_name(self):
        return self._name

    # setter method
    def set_name(self, x):
        self._name = x
    def get_lid(self):
        return self._lid

    # setter method
    def set_lid(self, x):
        self._lid = x

    def get_tot(self):
        return self._tot

    # setter method
    def set_tot(self, x):
        self._tot = x
obj=DB()
@app.route('/')
def home():
    try:
        lid=obj.get_lid()
        print(lid)
        return render_template('login.html',data=lid)
    except Exception as e:
        return render_template('login.html')


@app.route('/login')
def login():
    # obj._lid=0
    return render_template('login.html')
@app.route("/login", methods = ["GET","POST"])
def Login():

    l_id = request.form["logname"]
    l_pass  = request.form["logpass"]

    tab=l_id+l_pass
    #print("name")
    if request.method == 'POST':
        print("name")
        l_id = request.form["logname"]
        l_pass  = request.form["logpass"]

        tab=l_id+l_pass
        print(tab)
```

```
        obj.set_name(tab)
        obj.set_lid(l_id)
        return redirect(f"/check")

    return render_template('invalid.html',invalid='Please enter a valid data')

@app.route('/sign')
def sign():
    return render_template('signup.html')

@app.route("/regis",methods=["GET","POST"])
def regis():
    u_id = request.values.get("signu_id")
    s_pass  = request.values.get("sign_pass")
    print(u_id)
    print(s_pass)
    table_name=u_id+s_pass
    print(table_name)
    try:
        conn=sql.connect('main.db')

        print("Opened database successfully")
        create="CREATE TABLE "+table_name+" (detail TEXT, cred TEXT)"
        conn.execute(create)
        #conn.execute("select * from credential")

        print("Table created successfully")
        conn.close()
        return render_template("success.html")
    except:
        # print("dsa")
        return render_template("invalid.html", a="Username and password are already taken.
Try another.")
@app.route("/check",methods = ["GET","POST"])
def cart():
    tab=obj.get_name()
    lid=obj.get_lid()
    print(lid)
    if request.method == 'GET':
        print(f"Your name is {tab}")
        try:
            con = sql.connect("main.db")
            con.row_factory = sql.Row
            cur = con.cursor()
            a=f"select * from {tab}"
            print(a)
            cur.execute(a)
            return render_template('index.html',data=lid)
        except :
            return render_template('invalid.html',a='Please enter a valid data')


from flask import Flask, render_template, request
from keras.models import load_model
```

```
from keras.preprocessing import image
import pickle
import cv2
from skimage import feature
import os.path

# app = Flask(__name__)
def predict_label(img_path):
        model = pickle.loads(open('parkin.pkl', "rb").read())
        #img_path="C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V02HE01.png"
        # Pre-process the image in the same manner we did earlier
        image = cv2.imread(img_path)
        output = image.copy()

        # Load the input image, convert it to grayscale, and resize
        output = cv2.resize(output, (128, 128))
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        image = cv2.resize(image, (200, 200))
        image = cv2.threshold(image, 0, 255,
        cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

        # Quantify the image and make predictions based on the extracted features using
the last trained Random Forest
        features = feature.hog(image, orientations=9,
        pixels_per_cell=(10, 10), cells_per_block=(2, 2),
        transform_sqrt=True, block_norm="L1")
        preds = model.predict([features])
        print(preds)
        ls=["healthy","parkinson"]
        return(ls[preds[0]])
# routes
@app.route("/", methods=['GET', 'POST'])
def main():
    return render_template("index.html")

@app.route("/about")
def about_page():
    return "Please subscribe  Artificial Intelligence Hub..!!!"

@app.route("/submit", methods = ['GET', 'POST'])
def get_output():
        if request.method == 'POST':
            # img = request.files['my_image']
            # img_path = "..static/" + img.filename
            # img.save(img_path)
            img = request.files['my_image']

            img_path = "static/" + img.filename
            img.save(img_path)
            p=predict_label(img_path)
        return render_template("index.html", prediction = p, img_path = img_path)


if __name__ =='__main__':
```

```
    #app.debug = True
    app.run(debug = True)
```

HTML:

For Index Page:-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Parkinson Detection</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
  <style>
    body{
        background: linear-gradient(to bottom,  #9333ff 20%, #ffcc66 100%);
    }
</style>
</head>
<body>

<div class="container">
  <center><h1 class="jumbotron bg-primary">Parkinson Detection</h1></center>
  <br><br>
  <form class="form-horizontal" action="/submit" method="post" enctype="multipart/form-
data">

    <div class="form-group">
      <label class="control-label col-sm-2" for="pwd">Upload Your Image :</label>
      <div class="col-sm-10">
        <input type="file" class="form-control" placeholder="Hours
Studied"  name="my_image" id="pwd">
      </div>
    </div>

    <div class="form-group">
      <div class="col-sm-offset-2 col-sm-10">
        <button type="submit" class="btn btn-success">Submit</button>
      </div>
    </div>
  </form>

  {% if prediction %}
  <img src={{img_path}} height="200px" width="200px">
  <h2> Your Prediction   : <i> {{prediction}} </i></h2>

  {% endif %}
</div>
</body></html>
```

For Invalid:-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Invalid</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <style>
        body{
            background: linear-gradient(to bottom,  #9333ff 20%, #ffcc66 100%);
        }
    </style>
</head>
<body>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <center>
    <div class="jumbotron jumbotron-fluid" style="background-color: rgb(255, 235,
210);font-family: apple-system,BlinkMacSystemFont,'Segoe UI',Roboto,Helvetica,Arial,sans-
serif;height: 385px;width: 350px;border-radius: 20px;" >
    <h4>
        {{invalid}}
        {{a}}
    </h4>
    <br>
    <br>

    <br>
    <p class="ureg-sign-in txt-align-center">
        Return to
     <a href="/" class="js-link-feedback">Sign in</a> page</p>
    <br> <br>
    </div>
</center>
    <br>
    <br>
    <br>
    <br>
    <br>
    </body>
    </html>
```

For login :-

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <style>
        body{
            background: linear-gradient(to bottom,  #9333ff 20%, #ffcc66 100%);
        }
    </style>
</head>
<body>
    <br>
    <br>
    <br>
    <center style="margin-bottom: 40px;">
        <b><h2>Detection of Parkinson</h2></b>
    </center>
    <center>
    <div class="jumbotron jumbotron-fluid" style="background-color: rgb(255, 235,
210);font-family: apple-system,BlinkMacSystemFont,'Segoe UI',Roboto,Helvetica,Arial,sans-
serif;height: 375px;width: 350px; border-radius: 20px;" >
    <h4>
        Sign in
    </h4>
    <br>
    <br>
    <form action="{{ url_for('Login')}}" method="post">
        <input type="text" class="form-control" id="inputEmail4" placeholder="Username"
name="logname" style="margin: 10px;width:300px">
        <input type="password" class="form-control" id="inputPassword4"
placeholder="Password" name="logpass" style="margin: 10px;width:300px">


        <button type="submit" class="btn btn-info" style="margin: 10px;width:300px;
border-radius: 5px;">Login</button>
    </form>
    </div></center>
    <center>
        <div style="background-color: rgb(255, 235, 210);font-family: apple-
system,BlinkMacSystemFont,'Segoe UI',Roboto,Helvetica,Arial,sans-
serif;height:150px;width:350px; border-radius: 20px;" >
            <br>
            <label>Don't have an account?</label>
            <form action="/sign" ,method="POST">
                <button type="submit" class="btn btn-info" style="width:300px; border-
radius: 5px;">Sign up</button>
            </form>
    </div>
    </center>
```

```
<br>
<br>
<br>
</body>
</html
```

For Signup:-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign up</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <style>
        body{
            background: linear-gradient(to bottom,  #9333ff 20%, #ffcc66 100%);
        }
    </style>
</head>
<body>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <center>
    <div class="jumbotron jumbotron-fluid" style="background-color: rgb(255, 235,
210);font-family: apple-system,BlinkMacSystemFont,'Segoe UI',Roboto,Helvetica,Arial,sans-
serif;height: 385px;width: 350px; border-radius: 20px;" >
    <h4>
        Sign up
    </h4>
    <br>
    <br>
    <form action="/regis" ,method="POST">
        <input type="text" class="form-control" id="inputEmail4" placeholder="Username"
name="signu_id" style="margin: 10px;width:300px">
        <input type="password" class="form-control" id="inputPassword4"
placeholder="Password" name="sign_pass" style="margin: 10px;width:300px">
        <button type="submit" class="btn btn-info" style="margin: 10px;width:300px ;
border-radius: 10px;">Register</button>
    </form>
    <br>
    <p class="ureg-sign-in txt-align-center">
        Already have an account?
     <a href="/" class="js-link-feedback">Sign in</a></p>
    <br> <br>
    </div>
</center>
```

```
    <br>
    <br>
    <br>
    <br>
    <br>
    </body>
    </html>
```

For Success :-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Successfully Registered</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <style>
        body{
            background: linear-gradient(to bottom,  #9333ff 20%, #ffcc66 100%);
        }
    </style>
</head>
<body>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <center>
    <div class="jumbotron jumbotron-fluid" style="background-color: rgb(255, 235,
210);font-family: apple-system,BlinkMacSystemFont,'Segoe UI',Roboto,Helvetica,Arial,sans-
serif;height: 385px;width: 350px; border-radius: 20px;" >
    <h4>
        Successfully Registered
    </h4>
    <br>
    <br>
    <br>
    <p class="ureg-sign-in txt-align-center">
        Return to
    <a href="/" class="js-link-feedback">Sign in</a> page</p>
    <br> <br>
    </div>
</center>
    <br>
    <br>
    <br>
    <br>
    <br>
    </body>
```

```
</html>
```

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import glob
import cv2
import os
import seaborn as sns
import pandas as pd
from skimage.filters import sobel
import pickle
```

In [2]:

```
!pip install opencv-python
!pip install --ignore-installed --upgrade tensorflow==1.14
```

```
Requirement already satisfied: opencv-python in c:\users\balaj\anaconda3\lib\site-packages (4.6.0.66)
Requirement already satisfied: numpy>=1.14.5 in c:\users\balaj\anaconda3\lib\site-packages (from opencv-python) (1.21.5)

ERROR: Could not find a version that satisfies the requirement tensorflow==1.14 (from versions: 2.5.0, 2.5.1, 2.5.2, 2.5.3, 2.6.0rc0, 2.6.0rc1, 2.6.0rc2, 2.6.0, 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.7.0rc0, 2.7.0rc1, 2.7.0, 2.7.1, 2.7.2, 2.7.3, 2.7.4, 2.8.0rc0, 2.8.0rc1, 2.8.0, 2.8.1, 2.8.2, 2.8.3, 2.8.4, 2.9.0rc0, 2.9.0rc1, 2.9.0rc2, 2.9.0, 2.9.1, 2.9.2, 2.9.3, 2.10.0rc0, 2.10.0rc1, 2.10.0rc2, 2.10.0rc3, 2.10.0, 2.10.1, 2.11.0rc0, 2.11.0rc1, 2.11.0rc2, 2.11.0)
ERROR: No matching distribution found for tensorflow==1.14
```

In [3]:

```
train_images = []
train_labels = []
for directory_path in glob.glob("C:\\Users\\balaj\\Desktop\\parkinsons\\spiral\\testing\\*"):
    label = directory_path.split("\\")[-1]
    print(label)
```


Code.ipynb:-

```
        img = cv2.imread(img_path, cv2.IMREAD_COLOR)
        img = cv2.resize(img, (150, 150))
        img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
        train_images.append(img)
        train_labels.append(label)
for directory_path in glob.glob("C:\\Users\\balaj\\Desktop\\parkinsons\\wave\\training\\*"):
    label = directory_path.split("\\")[-1]
    print(label)
    for img_path in glob.glob(os.path.join(directory_path, "*.png")):
        print(img_path)
        img = cv2.imread(img_path, cv2.IMREAD_COLOR)
        img = cv2.resize(img, (150, 150))
        img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
        train_images.append(img)
        train_labels.append(label)

train_images = np.array(train_images)
train_labels = np.array(train_labels)
print("-----------------------------------------------------------")
test_images = []
test_labels = []
for directory_path in glob.glob("C:\\Users\\balaj\\Desktop\\parkinsons\\spiral\\testing\\*"):
    fruit_label = directory_path.split("\\")[-1]
    print(fruit_label)
    for img_path in glob.glob(os.path.join(directory_path, "*.png")):
        print(img_path)
        img = cv2.imread(img_path, cv2.IMREAD_COLOR)
        img = cv2.resize(img, (150, 150))
        img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
        test_images.append(img)
```



```
for directory_path in glob.glob("C:\\Users\\balaj\\Desktop\\parkinsons\\wave\\testing\\*"):
    fruit_label = directory_path.split("\\")[-1]
    print(fruit_label)
    for img_path in glob.glob(os.path.join(directory_path, "*.png")):
        print(img_path)
        img = cv2.imread(img_path, cv2.IMREAD_COLOR)
        img = cv2.resize(img, (150, 150))
        img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
        test_images.append(img)
        test_labels.append(fruit_label)

test_images = np.array(test_images)
test_labels = np.array(test_labels)
```

```
healthy
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V01HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V02HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V03HE1.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V04HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V05HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V06HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V07HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V08HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V09HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V10HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V11HE01.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V55HE12.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V55HE13.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V55HE14.png
C:\Users\balaj\Desktop\parkinsons\spiral\testing\healthy\V55HE15.png
```

In [4]:
```python
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
le.fit(test_labels)
test_labels_encoded = le.transform(test_labels)
le.fit(train_labels)
train_labels_encoded = le.transform(train_labels)

x_train, y_train, x_test, y_test = train_images, train_labels_encoded, test_images, test_labels_encoded
```

In [5]:
```python
x_train, x_test = x_train / 255.0, x_test / 255.0
```

In [6]:
```python
def feature_extractor(dataset):
    x_train = dataset
    image_dataset = pd.DataFrame()
    for image in range(x_train.shape[0]):
        df = pd.DataFrame()
        input_img = x_train[image, :,:,:]
        img = input_img

        pixel_values = img.reshape(-1)
        df['Pixel_Value'] = pixel_values

        num = 1
        kernels = []
        for theta in range(2):
            theta = theta / 4. * np.pi
            for sigma in (1, 3):
                lamda = np.pi/4
                gamma = 0.5
                gabor_label = 'Gabor' + str(num)
                ksize=9
                kernel = cv2.getGaborKernel((ksize, ksize), sigma, theta, lamda, gamma, 0, ktype=cv2.CV_32F)
                kernels.append(kernel)
                fimg = cv2.filter2D(img, cv2.CV_8UC3, kernel)
                filtered_img = fimg.reshape(-1)
                df[gabor_label] = filtered_img
                print(gabor_label, ': theta=', theta, ': sigma=', sigma, ': lamda=', lamda, ': gamma=', gamma)
                num += 1

        image_dataset = image_dataset.append(df)

    return image_dataset

image_features = feature_extractor(x_train)

```



```
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
```


In [8]:
```python
n_features = image_features.shape[1]
image_features = np.expand_dims(image_features, axis=0)
X_for_RF = np.reshape(image_features, (x_train.shape[0], -1))  #Reshape to #images, features

from sklearn.ensemble import RandomForestClassifier
RF_model = RandomForestClassifier(n_estimators = 80, random_state = 10, max_features = 0.2)
RF_model.fit(X_for_RF, y_train)
```

Out[8]: `RandomForestClassifier(max_features=0.2, n_estimators=80, random_state=10)`

In [9]:
```python
test_features = feature_extractor(x_test)
test_features = np.expand_dims(test_features, axis=0)
test_for_RF = np.reshape(test_features, (x_test.shape[0], -1))
```

```
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
```

In [10]:
```
test_prediction = RF_model.predict(test_for_RF)
test_prediction = le.inverse_transform(test_prediction)
```

In [11]:
```
from sklearn import metrics
print ("Accuracy = ", metrics.accuracy_score(test_labels, test_prediction))

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(test_labels, test_prediction)

fig, ax = plt.subplots(figsize=(6,6))
sns.set(font_scale=1.6)
sns.heatmap(cm, annot=True, ax=ax)
```

```
Accuracy =  0.8813559322033898
```

Out[11]: <AxesSubplot:>

In [12]:
```
import random
n=random.randint(0, x_test.shape[0]-1)
img = x_test[n]
plt.imshow(img)
input_img = np.expand_dims(img, axis=0)
input_img_features=feature_extractor(input_img)
input_img_features = np.expand_dims(input_img_features, axis=0)
input_img_for_RF = np.reshape(input_img_features, (input_img.shape[0], -1))
#Predict
img_prediction = RF_model.predict(input_img_for_RF)
img_prediction = le.inverse_transform([img_prediction])
print("The prediction for this image is: ", img_prediction)
print("The actual label for this image is: ", test_labels[n])
```

```
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
The prediction for this image is:  ['healthy']
The actual label for this image is:  healthy
```

```
C:\Users\balaj\AppData\Local\Temp\ipykernel_9108\2417115076.py:29: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  image_dataset = image_dataset.append(df)
C:\Users\balaj\anaconda3\lib\site-packages\sklearn\preprocessing\_label.py:154: DataConversionWarning: A column-vector y was pa
ssed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```



```
n=random.randint(0, x_test.shape[0]-1)
img = x_test[n]
plt.imshow(img)
input_img = np.expand_dims(img, axis=0)
input_img_features=feature_extractor(input_img)
input_img_features = np.expand_dims(input_img_features, axis=0)
input_img_for_RF = np.reshape(input_img_features, (input_img.shape[0], -1))
#Predict
img_prediction = RF_model.predict(input_img_for_RF)
img_prediction = le.inverse_transform([img_prediction])
print("The prediction for this image is: ", img_prediction)
print("The actual label for this image is: ", test_labels[n])
```

```
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
The prediction for this image is:  ['healthy']
The actual label for this image is:  healthy
```

```
C:\Users\balaj\AppData\Local\Temp\ipykernel_9108\2417115076.py:29: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  image_dataset = image_dataset.append(df)
C:\Users\balaj\anaconda3\lib\site-packages\sklearn\preprocessing\_label.py:154: DataConversionWarning: A column-vector y was pa
ssed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```


Reference:

Github link: -

https://github.com/IBM-EPBL/IBM-Project-11419-1659327603

Youtube: -

https://youtu.be/6Flbc6pQC6c