CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF OUR PROJECT

Parkinson is a neurodegenerative disease that affects your ability to control movement. Parkinson's disease starts slowly and worsens over time. The cured for Parkinson's disease isstill unknown; medications might significantly improve your symptoms. Researchers suggestthat early diagnosis of Parkinson can help improve the quality of the patient's life. In this survey, handwriting or drawings is considered as an aspect for detecting Parkinson disease using machine learning algorithm such as Random Forest Classifier and for detailed analysisof the drawings we use, Histogram of Oriented Gradients (HOG). We take drawings drawn by Parkinson patients as well as healthy people as input for detecting the Parkinson disease.

1.2 PURPOSE OF THE PROJECT

Machine learning algorithm used to diagnose Parkinson's disease. The ML-based diagnosis of this subjective disease can be achieved by using symptoms as an attribute for the algorithm. The ML algorithm is used to diagnose the PD severity from the handwriting of an individual and the goals of National Institute of Neurological Disorders and Stroke -supported research on Parkinson's disease are to better understand and diagnose PD, develop new treatments, and ultimately, prevent PD.

CHAPTER 3

LITERATURE SURVEY

The purpose of the Literature Survey is to give a brief overview and also to establish complete information about the reference papers. The goal of the Literature Survey is to completely specify the technical details related to the main project concisely and unambiguously.

2.1 Parkinson's Disease Diagnosis Using Machine Learning and Voice

Author name: Timothy J. Wroge , Yasin Ozkanca ¨ , Cenk Demiroglu , DongSi , David C. Atkins and Reza Hosseini Ghomi.

Published Year :2018

Merit:

- ✓ translate audio data into diagnostic tool provide diagnoses that are cheaper and more accurate

- ✓ deep neural networks are being to accurately diagnose individuals with the disease

Demerit:

- ✓ Non- invasive voice biomarkers is automated machine learning architecture for detecting and prediction

2.2 THE PARKINSON'S DISEASE DETECTION USING MACHINE LEARNING TECHNIQUES

Author name: Dr. C k gomathy,mr. B. Dheeraj kumar reddy, ms. B. Varsha, ms. B.Varshini

Published Year: 2021

Merit:

- ✓ It shows 73.8%of efficiency in finding out the symptoms by deflections in the voice.

- ✓ 60% is used for training and 40% issued for testing

Demerit:

- ✓ Algorithms used can be more efficient to detect

2.3 Machine Learning for the Diagnosis of Parkinson's Disease: A Review of Literature

Author Name :Jie Mei 1 *, Christian Desrosiers 2 and Johannes Frasnelli1,

Published Year:2021**Merit:**

- ✓ Many methods and objectives are being useful in adaptation of machine learning language
- ✓ Databases that can be used to enlarge and enrich smaller datasets

Demerit:

- ✓ large-scale, multi-eccentric studies assessment of Powered being excluded.
- ✓ Directly outcome is difficult to compare with the different models…

2.4 PARKINSON'S DISEASE DETECTION USING MACHINELEARNING

Author name: Shikha Singh, Nikita Shingade, Priti Sarote, Deepti Yelale andNihar Ranjan

Published Year :2022

Merits:

- ✓ therapies like levodopa/carbidopa are being more useful in early-stage treatment.
- ✓ It is used to speed up to detect the disease and it is cost effective

Demerit:

- ✓ accuracy of the disease can be detected by some algorithms such as bagging, boosting and so on. By utilizing additional models and by comparing the results it is more efficient in detecting diseases

2.5 Parkinson's DiseaseDetection from Resting-State EEG Signals Using Common Spatial Pattern, Entropy, and Machine Learning Technique

Author name: Majid Aljalal , Saeed A. Aldosari, Khalil AlSharabi , Akram M. Abdurraqeeb andFahd A. Alturki

Published Year : 2022

Merit:

- ✓ Several features are being extracted from spatially filtered signals.

- ✓ Features that are being extracted from alpha and beta gives result accuracy.

Demerit:

- ✓ Channel selection, Classification robustness, Source of data can be improved

2.6 Parkinson's DiseaseDetection Analysis through Machine Learning Approaches

Author Name: Muhtasim Shafi Kader, Dr.Fizar Ahmed, AnneshaAcharjee

Published Year:2022**Merit:**

- ✓ nine ML classifiers are being more helpful in detecting the disease

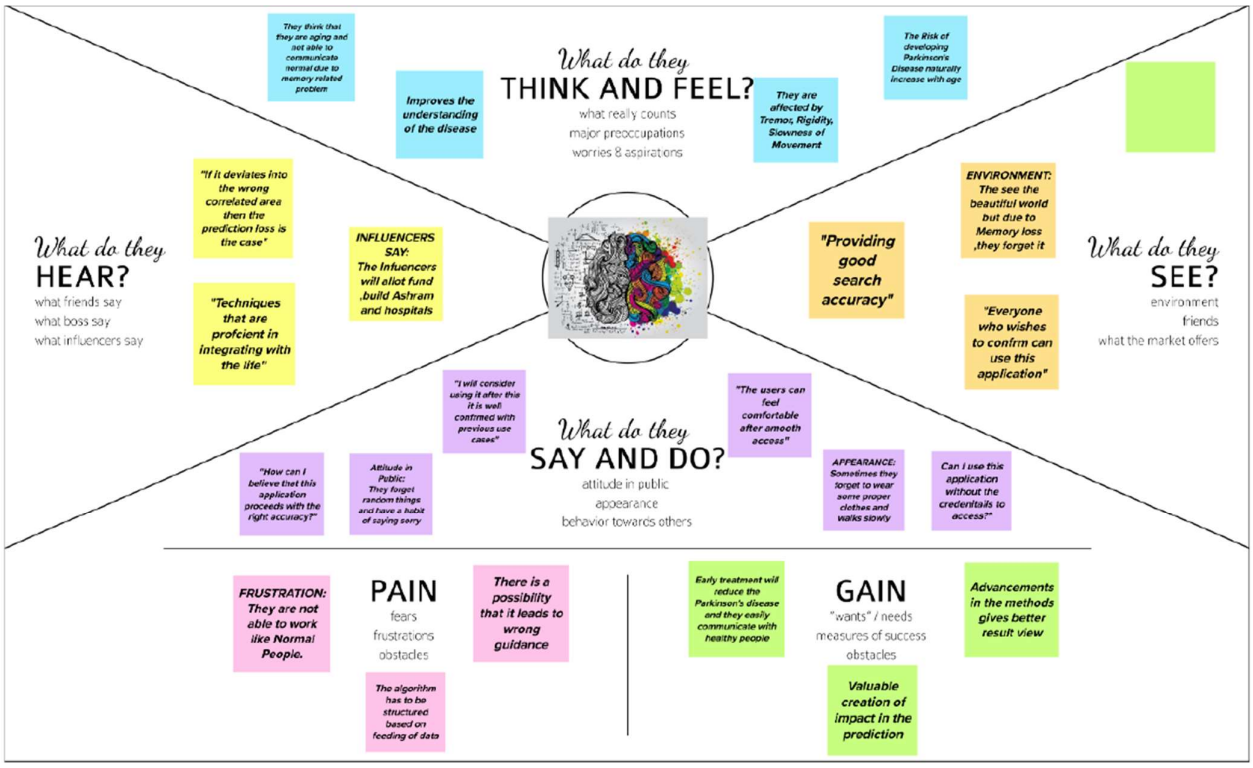
Demerit:

- ✓ predictable data sets insufficient for finding accurate results

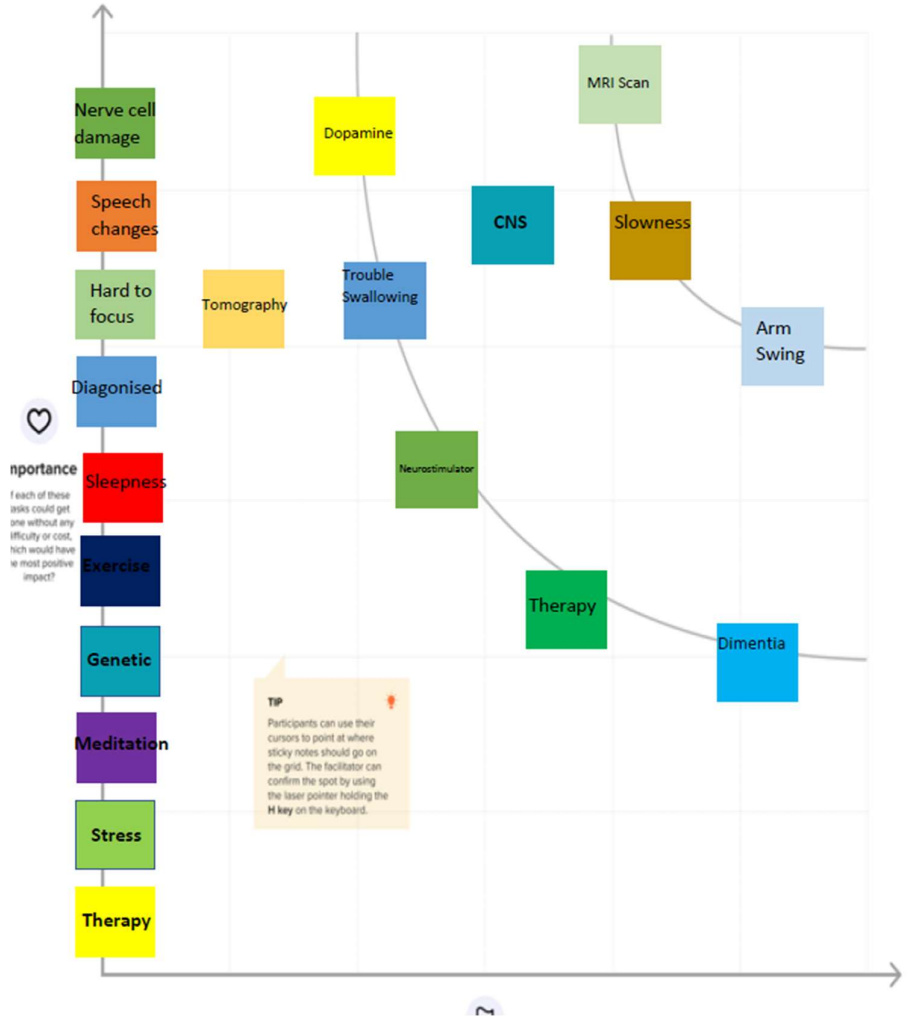
CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstormings



3.3 Proposed Solution

1. Problem Statement (Problem to be solved):

Inaccurate detection and diagnosis of Parkinson ‘s disease. Due to the lack of specific tests and misconception of other diseases sharing the same symptom, delayed or inaccurate results
are obtained.

2. Idea / Solution description:

To detect Parkinson’s disease with accuracy taking distinctive symptoms into account for larger set of people. It provides a solution for the confusion between othermotor(movement) diseases and Parkinson’s disease. The project aims to build a web application that detects Parkinson's
disease when hand-drawn images of spirals andwaves of it are given.

3. Novelty / Uniqueness:

It is found that the drawing speed was slower and the pen pressure is lower among Parkinson's patients. One of the indications of Parkinson's is tremors and rigidity in the muscles, making it difficult to draw smooth spirals and waves. It is possible to detect Parkinson's disease using the drawings alone instead of measuring the speed and pressure ofthe pen on paper. Our goal is to quantify the visual appearance (using HOG method) of these drawings and then train a machine learning model to classify them. In this project, we are using, Histogram of Oriented Gradients (HOG) image descriptor along with a Random Forest classifier to automatically detect Parkinson's disease in hand-drawn images of spiralsand waves.

4. Social Impact / Customer Satisfaction:

It becomes easier for the doctors and clinicians to diagnose Parkinson's disease among allthe other movement diseases with accuracy without much chaos and complications and thus provide the correct report and analytics

5. Business Model (Revenue Model):

The project converts the hand drawn patterns into the amount of pressure used by the patienton the pen and paper. This is integrated with other symptoms to get a clear picture of who is affected by Parkinson's disease to what extent

6. Scalability of the Solution:

The web application plays a crucial role in differentiating other motor diseases and Parkinson's diseases. It can be applied for a large dataset which saves time and brings efficiency.

3.4 Problem Solution fit:

1. CUSTOMER SEGMENT(S)

Who is your customer?

i.e working parents of 0-5 y.o kids

Neurologists Medical registrars Researchers

Patients (mainly age>60) who have defects in nervous systems

2.JOBS-TO-BE-DONE/PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides

At first, the ambiguity in diagnosing Parkinson detection is solved, since there is no fixed symptoms for the doctor to analyze from the patients.

Difficulty in using the data set to retrieve reports is solved by creating an interface in a user-friendly website

CHAPTER 4

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Confirmation	Confirmation via Email
FR-3	Input Data	User feeds the data set that contains various parameters that should be taken into account, especially the hand drawn wave and spiral patterns of the patients
FR-4	User authorization	A medical professional / organization or a patient will be verified to prevent fraudulent indulgence
FR-5	Classification of data	The website gives the result by categorizing the hand drawn patterns with the help of the pressure and tremor calculated.
FR-6	Report Generation	Reports will be produced in accordance with data fed.

4.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application can be used for accurate prediction and classification of the presence and severity of Parkinson's disease (based

		on hand drawn pattern)
NFR-2	**Security**	User's data is well encrypted using efficient algorithms of machine learning
NFR-3	**Reliability**	The application is monitored regularly in terms of the ability, quality and availability of constant prediction of the correct result
NFR-4	**Performance**	It classifies the images and predicts the disease with careful accuracy output. It reduces hectic calculations of the data set. Thus it works faster and
		efficiently
NFR-5	**Availability**	The application is active throughout the day. While awaiting the prediction result, user can interact with chat bot for knowing important details. If the application doesn't respond for the user, then the automated chat bot will forward the issue to our server then it can be resolved at that instance. It is compatible to all the operating systems.

NFR-6	**Scalability**	It does not request money or bank details to setup their account and download their final medical result from the application .

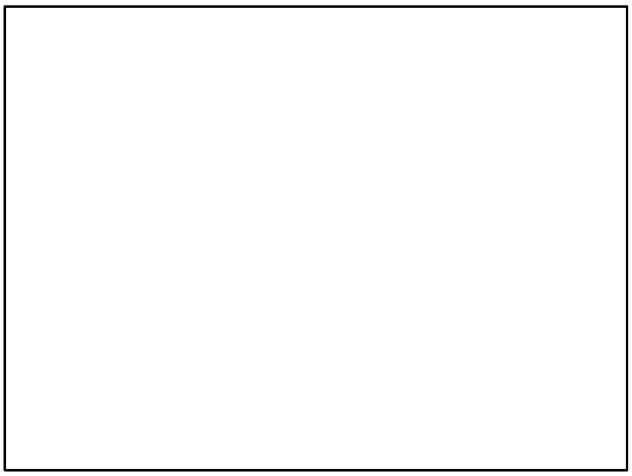
CHAPTER 5

5. PROJECT DESIGN

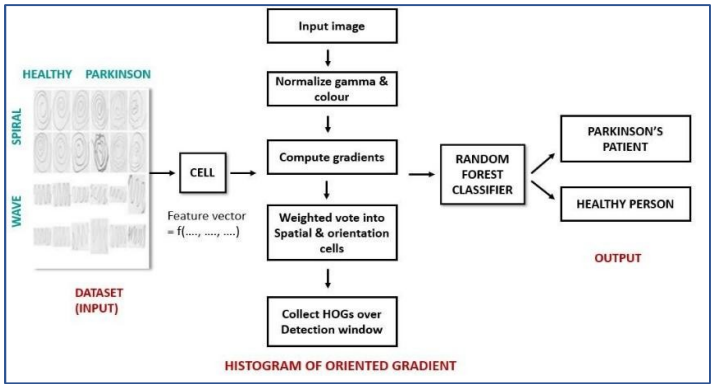
5.1 Data Flow Diagrams

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.

Example:



Example: DFD Level 0 (Detecting parkinsons disease using Machine Learning)



5.2 Solution & Technical Architecture

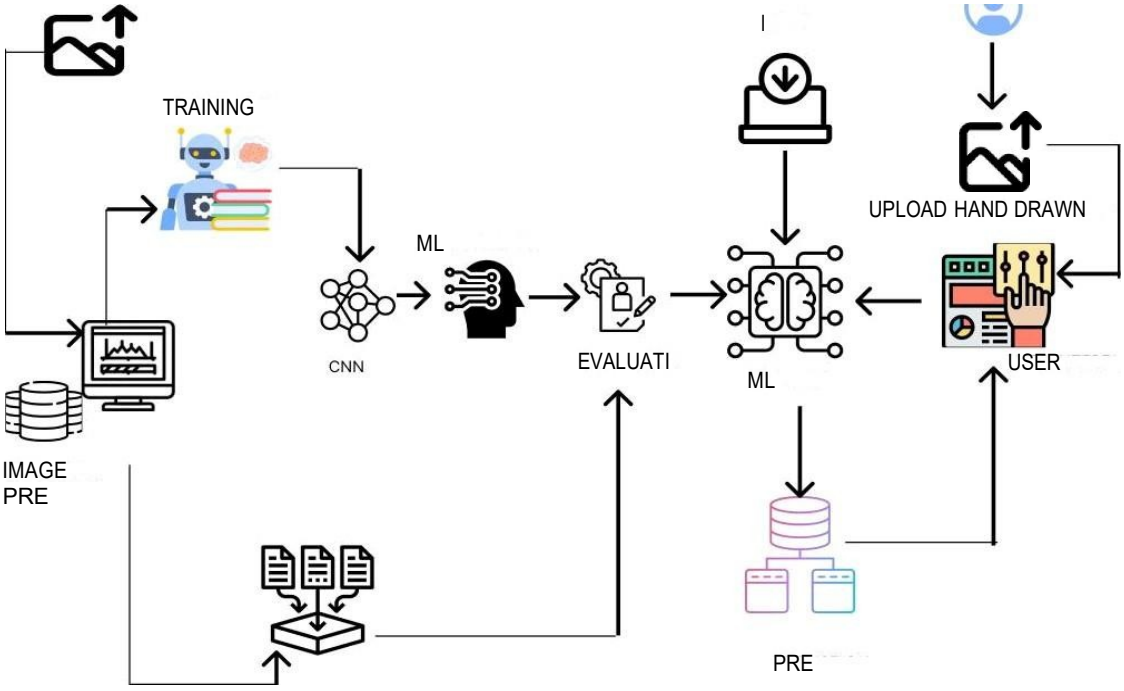


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Home Page	HTML, CSS, JavaScript
3.	Application Logic-2	Test Vital Page- Testing Image Uploading Page	HTML, CSS, JavaScript, Python, Flask
4.	Application Logic-3	Logic for a process in the application	Python, Flask
5.	Database	Data Type, Configurations etc… MySQL is an open-source relational database management system. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM Cloud, IBM DB2
7.	File Storage	File storage requirements	IBM Block Storage
8.	External API-1	Spiral and Waves hand-drawn images used for prediction of disease	Dataset for pre-processing IBM API Connect
9.	External API-2	Pre-processed dataset i.e Data analysis for Knowledge Base	Dataset for training the model

10.	Machine Learning Model	To train the Machine Learning Model to predict the parkinson's disease using Random Forest Classifier Algorithm	Parkinson Disease prediction model using CNN, HOG, Random Forest Classifier methodologies
11	Infrastructure (Server / Cloud	Application Deployment on Local System / Cloud Local Server Configuration: Local System Cloud Server Configuration: IBM Cloud (IBM Watson)	Local, IBM Watson (IBM Cloud Service)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks & Libraries	The libraries are used for data pre- processing, data visualization and to train ML model.	Numpy, Pandas, Matplotlib, scikit-image, Open CV, imutils, scikit-learn and various other libraries
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Built-in Encryptions, BYOK
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services). Since various python libraries and Flask as server has been used, it may speed up the performance and accuracy .	Python, Flask

4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	IBM Cloud, IBM Watson
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Python, Flask (To handle multiple requests from the client side)

5.3 USER STORIES

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Customer	Login	USN-1	Entering Webpage	Enter the application	High	Sprint 1
Customer	Homepage	USN-2	Entering to the "Homepage" of the UI(Webpage)	Enter the homepage	High	Sprint 1
Customer	About	USN-3	I can click on the "About" to details	Get the	Low	Sprint 2

			about theApplication	details about the application		
Customer	Begin	USN-4	As a user I can upload my ha n d w r i t t e n da t a from the computer.	Choose my handwritten data from my device	High	Sprint 2
Customer	Predict	USN-5	As a user I can insert my dataset from the device	Insert the dataset For prediction	High	Sprint 3
Customer	Output	USN-6	Predicting byusing handwritten data	Predict whether there is change in stoke or not	High	Sprint 3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Upload Images	USN-1	As a user, I can upload the images in the website in order to obtain the prediction result of parkinson's disease	2	High	1. Rajasekar V 2. Nandhakumar G 3. Pradeesh S 4. Ragul M
Sprint-4	Test Vital Page	USN-2	As a user, I will recieve prediction result and accuracy on the test vital page.	3	High	1. Rajasekar V 2. Nandhakumar G 3. Pradeesh S 4. Ragul M

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Dashboard	USN-3	Dashboard displays the symptoms, causes and medications for the Parkinson disease	2	Low	1.Rajasekar V 2.Nandhakumar G 3.Pradeesh S 4.Ragul M
Sprint-1	Data Collection	USN-4	As an Administrator, I need to collect data (images of spirals and waves drawn by healthy people and Parkinson's patients).	2	High	1.Rajasekar V 2.Nandhakumar G 3.Pradeesh S 4.Ragul M

Sprint-1	Data Pre-Processing	USN-5	As an Administrator, I should cleanmy data and prepare it for model building by doing pre-processing activities such as resizing, visualizing the dataset and converting from RGB to grayscale	2	High	1.Rajasekar V 2.Nandhakumar G 3.Pradeesh S 4.Ragul M
Sprint-2	Model Building	USN-6	As an Administrator, I need to build the model using Random Forest Classifier for spiral images and Convolutional	3	High	1.Rajasekar V 2.Nandhakumar G 3.Pradeesh S 4.Ragul M

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Member**
Sprint-3	Deployment of Model	USN-7	As an Administrator, I need to deploy the Machine Learning model that was built.	2	Medium	1.Rajasekar 2.Nandhakumar 3.Pradeesh 4.Ragul M

Sprint-3	Building Frontend of the application	USN-8	As an Administrator, I need to build the website for the application using HTML, CSS etc.	2	High	1.Rajasekar V 2.Nandhakumar.G 3.Pradeesh 4.Ragul M	.J
Sprint-4	Connecting theML model, Frontend and Backend	USN-9	As an Administrator, I can integrate the deployed model andweb application using python flask server.	3	High	1.Rajasekar V 2.Nandhakumar.G 3.Pradeesh 4.Ragul M	

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 Nov 2022

Velocity:

For example, imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

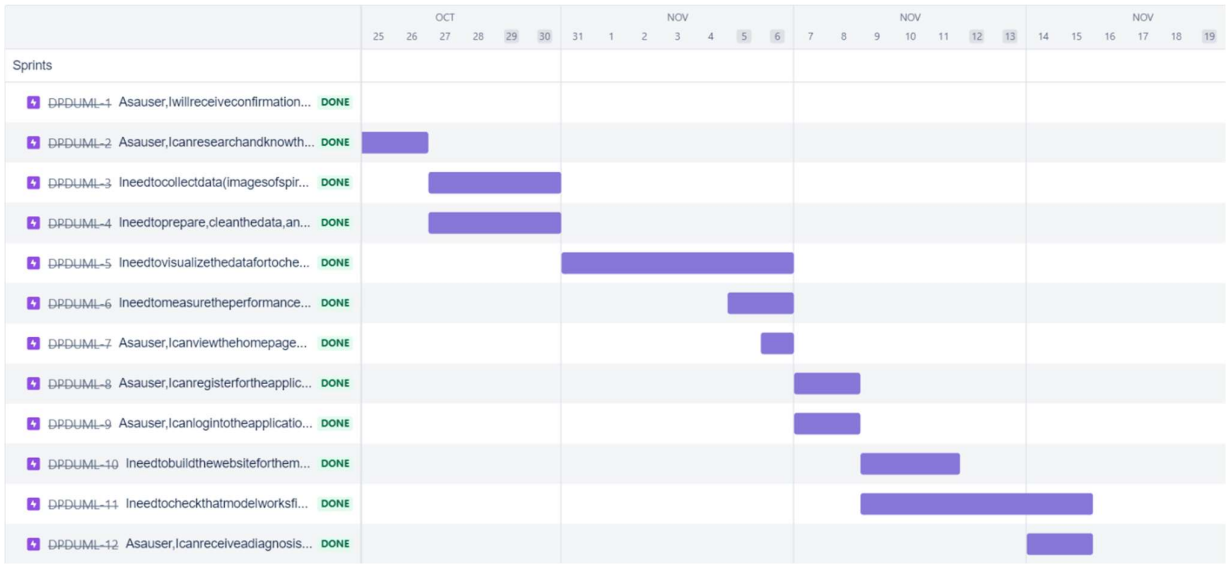
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

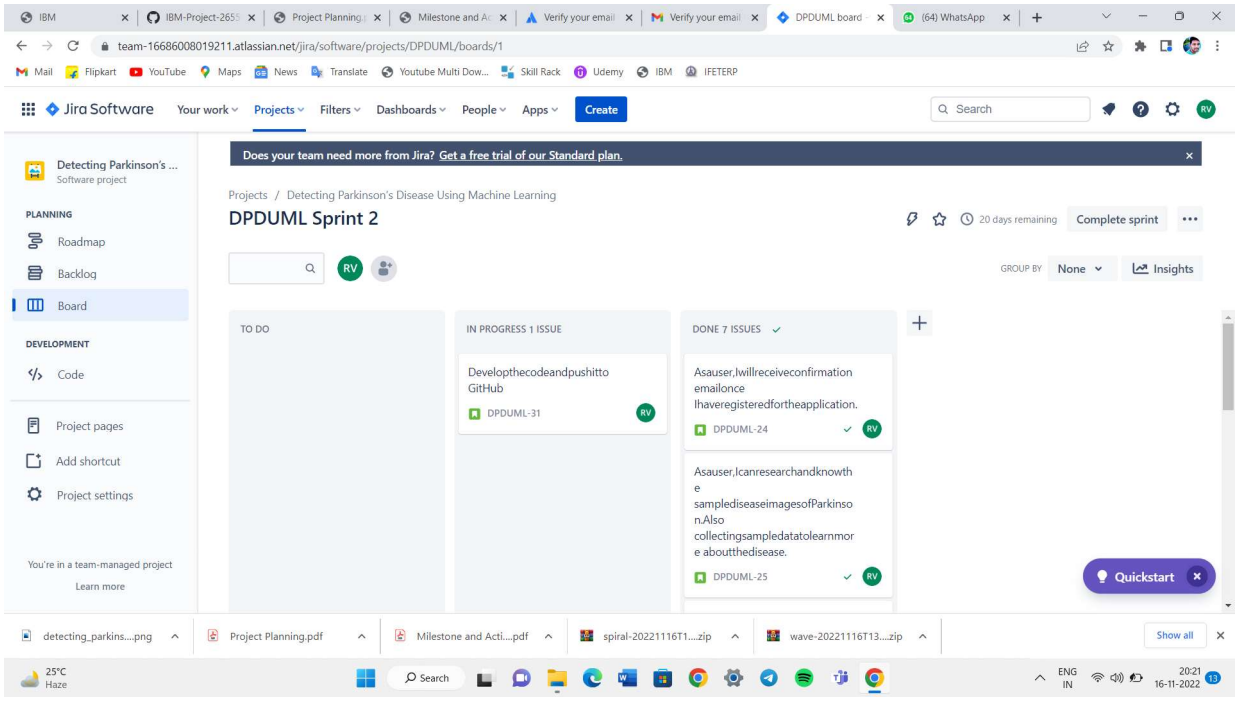
In our project, we have a 6-days sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's

average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{Sprint Duration}}{\text{Velocity}} = \frac{20}{6} = 3.3 \text{ (approx.)}$$

6.3 JIRA FILES





7. CODING & SOLUTIONING

7.1 Feature 1

home.html

```
<!DOCTYPE html>


 <html lang="en">
  <head>
   <meta charset="UTF-8" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <meta http-equiv="X-UA-Compatible" content="ie=edge" />
   <title>HomePage</title>
   <style>
    body {
     background: linear-gradient(to right, #33ccff 0%, #99ffcc 100%);
     background-size: cover;
     background-position: relative;
     background-repeat: no-repeat;
     height: 100%;
     width: 100%;
    }
    h3 {
     text-align: center;
     color: white;
    }
```

```
.main {
  margin-top: 100px;
}
p {
  color: black;
  text-indent: 10px;
  margin: 10px;
  font-size: 20px;
}

a {
  color: grey;
  float: right;
  text-decoration: none;
  font-style: normal;
  padding-right: 20px;
}

a:hover {
  background-color: black;
  color: white;
  font-size: 30px;
  padding-left: 10px;
  border-radius: 5px;
}

ul {
  align-items: center;
  display: flex;
  list-style-type: none;
  width: 100%;
  gap: 3rem;
  justify-content: center;
  font-size: 2rem;
  position: fixed;
  top: 0;
  margin: 0;
  padding: 1rem;
  background-color: white;
}

li {
  cursor: pointer;
}
li a {
  text-decoration: none;
  color: inherit;
}
```

```
    li.active {
      font-weight: bold;
      color: #A74AC7;
    }

    img {
      width:   450px;
      height:  400px;
      padding: 25px;
    }
    img:hover {
      border-color: grey;
    }
    #im {
      width: 1450px;
      height:  700px;
      padding: 25px;
    }
  </style>
</head>
<body>
  <nav>
    <ul>
      <li class="active"><a href="home.html">Home</a></li>
      <li class="active"><a href="pred.html">Prediction</a></li>
    </ul>
  </nav>
  <br /><br /><br />
  <h1>
    <center>
      <b class="pd"
        ><font color="black" size="15" font-family="Comic Sans MS"
          >Detection of Parkinson's Disease using ML</font
        ></b
      >
    </center>
  </h1>
  <div>
    <center>
      <p style="text-align: left">
        Parkinson disease (PD) is a progressive neuro degenerative disorder
        that impacts more than 6 million people around the world. Parkinson's
        disease is non-communicable, early-stage detection of Parkinson's can
        prevent further damages in humans suffering from it.
        However,Nonetheless, non-specialist physicians still do not have a
        definitive test for PD, similarly in the early stage of the diseased
        person where the signs may be intermittent and badly characterized. It
        resulted in a high rate of misdiagnosis (up to 25% among
```

non-specialists) and many years before treatment, patients can have the disorder. A more accurate, unbiased means of early detection is required, preferably one that individuals can use in their home setting.However, it has been observed that PD's presence in a human is related to its hand-writing as well as hand-drawn subjects. From that perspective, several techniques have been proposed by researchers to detect Parkinson's disease from hand-drawn images of suspected people. But the previous methods have their constraints.

```
  </p>
 </center>
 <h4>
  <center>
   <b class="pd"
    ><font color="#C12283" size="12" font-family="Cambria Math"
     >Causes and Symptoms of Parkinson's Disease</font
    >
   </b>
  </center>
 </h4>
 <span>
  <img
   src="https://www.narayanahealth.org/blog/wp-content/uploads/2015/04/parkinson.png"
   title="Disease"
  />
 </span>
 <span>
  <img

src="https://stanfordmedicine25.stanford.edu/the25/parkinsondisease/_jcr_content/main/panel_builder_0/panel_0/panel_builder_0/panel_0/panel_builder/panel_0/image.img.476.high.png/1.png"
   title="Symptoms"
  /></span>
```

Pred.html

```
<div>
        <form id="upload-file" method="post" enctype="multipart/form-data">
          <center>
            <label for="imageUpload" class="upload-label">
              Choose...
            </label>
            <input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
          </center>
        </form>

       <center> <div class="image-section" style="display:none;">
          <div class="img-preview">
            <div id="imagePreview">
            </div></center>
```

```
    </div>
    <center>
      <div>
        <button type="button" class="btn btn-primary btn-lg " id="btn-predict">Predict!</button>
      </div>
    </center>
  </div>

  <div class="loader" style="display:none;"></div>

  <h3 id="result">
    <span> </span>
  </h3>

</div>
```

FEATURE 2

Image Pre-Processing

Splitting up of training and testing data

```
trainX = []
testX = []
outputs = []
trainY = []
testY = []

for i in spiral_train_healthy:
  image = cv2.imread(fp_spiral_train_healthy+i)
  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]
  features = quantify_image(image)
  trainX.append(features)
  trainY.append('healthy')

for i in spiral_train_park:
  image = cv2.imread(fp_spiral_train_park+i)
  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]
  features = quantify_image(image)
  trainX.append(features)
  trainY.append('parkinson')

for i in spiral_test_healthy:
  image = cv2.imread(fp_spiral_test_healthy+i)
  outputs.append(image)
  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]
  features = quantify_image(image)
  testX.append(features)
  testY.append('healthy')
```

```
for i in spiral_test_park:
  image = cv2.imread(fp_spiral_test_park+i)
  outputs.append(image)
  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]
  features = quantify_image(image)
  testX.append(features)
  testY.append('parkinson')
```

Testing the Model

In [ ]:

```
testpath=list(paths.list_images(fp_spiral_train_healthy))
idxs=np.arange(0,len(testpath))
idxs=np.random.choice(idxs,size=(25,),replace=False)
images=[]

for i in idxs:
    image=cv2.imread(testpath[i])
    output=image.copy()
    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]

    features= quantify_image(image)
    preds=model.predict([features])
    label=le.inverse_transform(preds)[0]
    if label=="healthy":
      color=(0,255,0)
    else:
      (0,0,255)
    cv2.putText(output,label, (3,20),cv2.FONT_HERSHEY_SIMPLEX,0.5,color,2)
    images.append(output)
```

In [ ]:

```
'''montage = build_montages(images,(128,128),(5,5))[0]
cv2.imshow(montage)
cv2.waitKey(0)'''

montage=build_montages(images,(128,128),(5,5))[0]
cv2_imshow(montage)
cv2.waitKey(0)
```

Predicting the model-Accuracy and Confusion Matrix

In [ ]:

```
predictions = model.predict(testX)

cm = confusion_matrix(testY, predictions).flatten()
print(cm)
(tn, fp, fn, tp) = cm
accuracy = (tp + tn) / float(cm.sum())
print(accuracy)
```

8.4 TEST CASE

8.4.1 UNIT TEST

Unit testing involves the planning of test cases that validate that the inner program logic is functioning properly, which program inputs manufacture valid outputs. All call branches and internal code flow ought to be valid. it's the testing of individual software package units of the appliance. it's done when the completion of a personal unit before integration. this can be a structural testing, that depends on information of its construction and is invasive. Unit checks perform basic tests at element level and test a particular business method, application, and/or system configuration. Unit tests make sure that every distinctive path of a business method performs accurately to the documented specifications and contains clearly outlined inputs and expected results.

8.4.2 INTEGRATION TEST:

Software integration testing is that the progressive integration testing of 2 or additional integrated software package parts on one platform to provide failures caused by interface defects.

The task of the mixing check is to envision that parts or software package applications, e.g., parts during a software or – one accelerates – software package applications at the corporate level – move while not error.

Test results: All the check cases mentioned higher than have passed with success and no defects encountered. Integration testing is specifically aimed toward exposing the issues that arise from the mixture of parts.

8.4.3 FUNCTIONAL TEST:

Functional tests offer systematic demonstrations that functions tested at obtainable as mere by the business and technical needs, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked.

organization and preparation of purposeful tests is targeted on needs, key functions, or special check cases. additionally, systematic coverage concerning determine Business method flows; information fields, predefined processes, and consecutive processes should be thought of for testing. Before purposeful testing is complete, further tests are known and therefore the effective worth of current tests is decided.

8.4.4 SYSTEM TEST:

System testing ensures that the complete integrated software meets the necessities. It tests a configuration to confirm far-famed and certain results. associate example of system checking is that the configuration destined system integration test. System testing is predicated on method descriptions and flows, accenting pre-driven method links and integration points.

8.4.5 ACCEPTANCE TEST:

User acceptance testing may be a crucial part of any project and needs vital participation by the top user. It additionally ensures that the system meets the purposeful needs.

Test results: All the check cases mentioned higher than have passed with success and no defects encountered.

8.4.6 INTERFACE TEST:

The Interface Testing is performed to verify the interfaces between sub modules whereas playacting integration of sub modules aiding master module recursively.

8.5 TESTING STRATEGIES

A number of software testing strategies have been proposed in the literature All offer the software package developer with a guide for testing and everyone have the subsequent generic characteristics:

• Testing begins at the element level and works “outward” toward the mixing of the complete computer-based system.

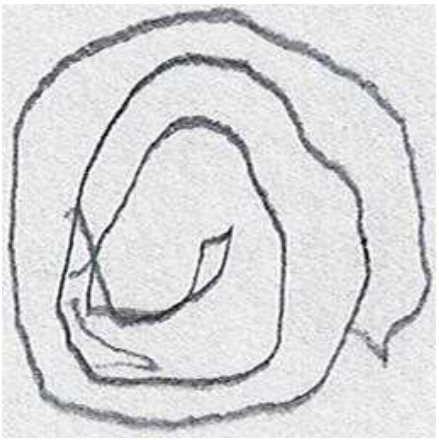
• Different testing techniques are applicable at totally different points in time.

• The developer of the s/w conducts testing and for big comes, freelance check cluster.

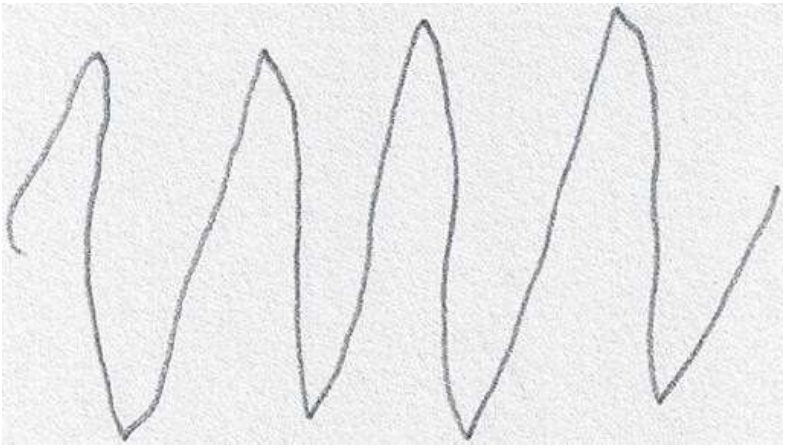
9 RESULT

9.2 Performance Metrics

The ML model which is developed using Random Forest Classifier Algorithm generates accuracy of 0.8666 approximately is deployed using IBM Watson which enhances scalability, reliability, security and performance of the ML model.



PARKINSON'S DISEASE



HEALTHY

10. ADVANTAGES & DISADVANTAGES

Advantages:

It serves as an important tool for a doctors and medical professionals to significantly diagnose Parkinson's disease that has not got any fixed or definite symptoms earlier.

It facilitates patients to be clarified with the appropriate medical illness and get treated earlier

The hand drawn pattern which is used as the parameter for the detection plays a major role in neglecting all the other factors which caused confusion thus providing a clear conclusion.

Disadvantages:

It sometimes fails to produce accurate results which ends up to be inefficient sometimes

Improper data set may lead to inappropriate results , thus it has a high rate of dependency

Hand drawn patterns may some times fail to be the main factor to help the prediction

11. CONCLUSION

Artificial intelligence and medical research have formed a partnership that aids in the treatment of ubiquitous disorders such as Parkinson's disease. For early detection of Parkinson's disease, symptoms such as Bradykinesia, Tremor at rest, Rigidity, and Voice Impairment can be noticed. There is no specific medical method or diagnosis for a person's parkinsonism, which also applies to bioinformatics. Strong techniques like Machine Learning, on the other hand, have sped up the process of detecting Parkinson's disease by making it more cost-effective and efficient. Machine learning can help doctors detect the parkinson's disease using hand drawn patterns which analyses the tremor by the pressure applied to drawn the spirals and waves.

12. Future Scope:

The study used a single model for each purpose to detect and assess the severity of Parkinson's disease. The research can be expanded by utilizing additional models and comparing the results to establish the most optimized and efficient models for disease detection and determining the degree of disease in the patient

13. APPENDIX

1.Mei J, Desrosiers C, Frasnelli J. Machine Learning for the Diagnosis of Parkinson's Disease: A Review of Literature. Front Aging Neurosci. 2021 May 6;13:633752. doi: 10.3389/fnagi.2021.633752. PMID: 34025389; PMCID: PMC8134676.

2. Gomathy, C K. (2021). THE PARKINSON'S DISEASE DETECTION USING MACHINE LEARNING TECHNIQUES.

3. Singh, Shikha & Sarote, Priti & Shingade, Nikita & Yelale, Deepti & Ranjan, Nihar. (2022). Detection of Parkinson's Disease using Machine Learning Algorithm. International Journal of Computer Applications. 184. 24-29. 10.5120/ijca2022922016.

4. Shafi, Muhtasim & Ahmed, Fizar. (2022). Parkinson's Disease Detection Analysis through Machine Learning Approaches.

5. Li, Alex & Li, Chenyu. (2022). Detecting Parkinson's Disease through Gait Measures Using Machine Learning. Diagnostics. 12. 2404. 10.3390/diagnostics12102404.

6. Wroge, Timothy & Özkanca, Yasin & Demiroglu, Cenk & Si, Dong & Atkins, David & Ghomi, Reza. (2018). Parkinson's Disease Diagnosis Using Machine Learning and Voice. 10.1109/SPMB.2018.8615607.

7. Fahim, Nasif & Eshti, Samik & Nura, Khadiza & Abir, Md & Mahbub, Mueem. (2020). Parkinson Disease Detection: Using XGBoost Algorithm to Detect Early Onset Parkinson Disease.

8. Marimuthu, Marimuthu. (2021). Detection of Parkinson's disease using Machine Learning Approach. Annals of the Romanian Society for Cell Biology. 25. 2544-2550.

9. Nissar, Iqra & Rizvi, Danish & Masood, Sarfaraz & Mir, Aqib. (2018). Voice-Based Detection of Parkinson's Disease through Ensemble Machine Learning Approach: A Performance Study. EAI Endorsed Transactions on Pervasive Health and Technology. 5. 162806. 10.4108/eai.13-7-2018.162806.

10. Pramanik, Moumita & Borah, Samarjeet & Nandy, Parvati & Pradhan, Ratika. (2022). A Machine Leaning based Parkinson's Detection using Acoustic Features. 1-4. 10.1109/ICCSEA54677.2022.9936311.

Source Code

home.html

```
<!DOCTYPE html>


 <html lang="en">
  <head>
   <meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<meta http-equiv="X-UA-Compatible" content="ie=edge" />
<title>HomePage</title>
<style>
 body {
  background: linear-gradient(to right, #33ccff 0%, #99ffcc 100%);
  background-size: cover;
  background-position: relative;
  background-repeat: no-repeat;
  height: 100%;
  width: 100%;
 }
 h3 {
  text-align: center;
  color: white;
 }
 .main {
  margin-top: 100px;
 }
 p {
  color: black;
  text-indent: 10px;
  margin: 10px;
  font-size: 20px;
 }

 a {
  color: grey;
  float: right;
  text-decoration: none;
  font-style: normal;
  padding-right: 20px;
 }

 a:hover {
  background-color: black;
  color: white;
  font-size: 30px;
  padding-left: 10px;
  border-radius: 5px;
 }

 ul {
  align-items: center;
  display: flex;
  list-style-type: none;
  width: 100%;
  gap: 3rem;
```

```
    justify-content: center;
    font-size: 2rem;
    position: fixed;
    top: 0;
    margin: 0;
    padding: 1rem;
    background-color: white;
   }

   li {
    cursor: pointer;
   }
   li a {
    text-decoration: none;
    color: inherit;
   }
   li.active {
    font-weight: bold;
    color: #A74AC7;
   }

   img {
    width:   450px;
    height:  400px;
    padding: 25px;
   }
   img:hover {
    border-color: grey;
   }
   #im {
    width: 1450px;
    height:  700px;
    padding: 25px;
   }
  </style>
</head>
<body>
  <nav>
   <ul>
    <li class="active"><a href="home.html">Home</a></li>
    <li class="active"><a href="pred.html">Prediction</a></li>
   </ul>
  </nav>
  <br /><br /><br />
  <h1>
   <center>
    <b class="pd"
     ><font color="black" size="15" font-family="Comic Sans MS"
```

```
        >Detection of Parkinson's Disease using ML</font
      ></b
     >
    </center>
   </h1>
   <div>
    <center>
     <p style="text-align: left">
      Parkinson disease (PD) is a progressive neuro degenerative disorder
      that impacts more than 6 million people around the world. Parkinson's
      disease is non-communicable, early-stage detection of Parkinson's can
      prevent further damages in humans suffering from it.
      However,Nonetheless, non-specialist physicians still do not have a
      definitive test for PD, similarly in the early stage of the diseased
      person where the signs may be intermittent and badly characterized. It
      resulted in a high rate of misdiagnosis (up to 25% among
      non-specialists) and many years before treatment, patients can have
      the disorder. A more accurate, unbiased means of early detection is
      required, preferably one that individuals can use in their home
      setting.However, it has been observed that PD's presence in a human is
      related to its hand-writing as well as hand-drawn subjects. From that
      perspective, several techniques have been proposed by researchers to
      detect Parkinson's disease from hand-drawn images of suspected people.
      But the previous methods have their constraints.
     </p>
    </center>
    <h4>
     <center>
      <b class="pd"
       ><font color="#C12283" size="12" font-family="Cambria Math"
        >Causes and Symptoms of Parkinson's Disease</font
       >
      </b>
     </center>
    </h4>
    <span>
     <img
      src="https://www.narayanahealth.org/blog/wp-content/uploads/2015/04/parkinson.png"
      title="Disease"
     />
    </span>
    <span>
     <img

src="https://stanfordmedicine25.stanford.edu/the25/parkinsondisease/_jcr_content/main/panel_builder_
0/panel_0/panel_builder_0/panel_0/panel_builder/panel_0/image.img.476.high.png/1.png"
      title="Symptoms"
    /></span>
```

Pred.html

```
<div>
            <form id="upload-file" method="post" enctype="multipart/form-data">
              <center>
                <label for="imageUpload" class="upload-label">
                  Choose...
                </label>
                <input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
              </center>
            </form>

           <center> <div class="image-section" style="display:none;">
              <div class="img-preview">
                <div id="imagePreview">
                </div></center>
              </div>
              <center>
                <div>
                  <button type="button" class="btn btn-primary btn-lg " id="btn-predict">Predict!</button>
                </div>
              </center>
            </div>

            <div class="loader" style="display:none;"></div>

            <h3 id="result">
              <span> </span>
            </h3>

          </div>
```

FEATURE 2

Image Pre-Processing

Splitting up of training and testing data

```
trainX = []
testX = []
outputs = []
trainY = []
testY = []

for i in spiral_train_healthy:
  image = cv2.imread(fp_spiral_train_healthy+i)
  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]
  features = quantify_image(image)
```

```
  trainX.append(features)
  trainY.append('healthy')

for i in spiral_train_park:
  image = cv2.imread(fp_spiral_train_park+i)
  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]
  features = quantify_image(image)
  trainX.append(features)
  trainY.append('parkinson')

for i in spiral_test_healthy:
  image = cv2.imread(fp_spiral_test_healthy+i)
  outputs.append(image)
  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]
  features = quantify_image(image)
  testX.append(features)
  testY.append('healthy')

for i in spiral_test_park:
  image = cv2.imread(fp_spiral_test_park+i)
  outputs.append(image)
  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]
  features = quantify_image(image)
  testX.append(features)
  testY.append('parkinson')
```

Testing the Model

In [ ]:

```
testpath=list(paths.list_images(fp_spiral_train_healthy))
idxs=np.arange(0,len(testpath))
idxs=np.random.choice(idxs,size=(25,),replace=False)
images=[]

for i in idxs:
    image=cv2.imread(testpath[i])
    output=image.copy()
    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]

    features= quantify_image(image)
    preds=model.predict([features])
    label=le.inverse_transform(preds)[0]
    if label=="healthy":
      color=(0,255,0)
    else:
      (0,0,255)
    cv2.putText(output,label, (3,20),cv2.FONT_HERSHEY_SIMPLEX,0.5,color,2)
    images.append(output)
```

In [ ]:

'''montage = build_montages(images,(128,128),(5,5))[0]
cv2.imshow(montage)
cv2.waitKey(0)'''

montage=build_montages(images,(128,128),(5,5))[0]
cv2_imshow(montage)
cv2.waitKey(0)

Predicting the model-Accuracy and Confusion Matrix

In [ ]:

predictions = model.predict(testX)

cm = confusion_matrix(testY, predictions).flatten()
print(cm)
(tn, fp, fn, tp) = cm
accuracy = (tp + tn) / float(cm.sum())
print(accuracy)

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-22034-1659801656