1. INTRODUCTION

1.1 Overview

Parkinson's disease is a progressive disorder of the central nervous system affecting movement and inducing tremors and stiffness. It has 5 stages to it and affects more than 1 million individuals every year in India. This is chronic and has no cure yet. It is a neurodegenerative disorder affecting dopamine-producing neurons in the brain.
For detecting PD, various machine learning models such as logistic regression, naive Bayes, KNN, and forest decision tree were used, with the features used here being minimum-redundancy maximum-relevance and recursive feature elimination. The accuracy obtained was 95.3% using data from the UCI machine learning repository. The researchers 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 of the 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 spirals and waves.

1.2 Purpose

By using machine learning techniques, the problem can be solved with minimal error rate. The voice dataset of Parkinson's disease from the UCI Machine learning library is used as input. Also our proposed system provides accurate results by integrating spiral drawing inputs of normal and Parkinson's affected patients. 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 diagnosis of PD and rely on these alternative measures to detect PD in preclinical stages or atypical forms. In recent years, the number of publications on the application of machine learning to the diagnosis of PD has increased. Although previous studies have reviewed the use of machine learning in the diagnosis and assessment of PD, they were limited to the analysis of motor symptoms, kinematics, and wearable sensor data. Moreover, some of these reviews only included studies published between 2015 and 2016. In this study, we aim to comprehensively summarize all published studies that applied machine learning models to the diagnosis of PD for an exhaustive overview of data sources, data types, machine learning models, and associated outcomes, (b) assess and compare the feasibility and efficiency of different machine learning methods in the diagnosis of PD, and (c) provide machine learning practitioners interested in the diagnosis of PD with an overview of previously used models and data modalities and the associated outcomes, and recommendations on how experimental protocols and results could be reported to facilitate reproduction. As a result, the application of machine learning to clinical and non-clinical data of different modalities has often led to high diagnostic accuracies in human participants, therefore may encourage the adaptation of machine learning algorithms and novel biomarkers in clinical settings to assist more accurate and informed decision making.While Parkinson’s cannot be cured, early detection along with proper medication can significantly improve symptoms and quality of life.

2. Literature survey

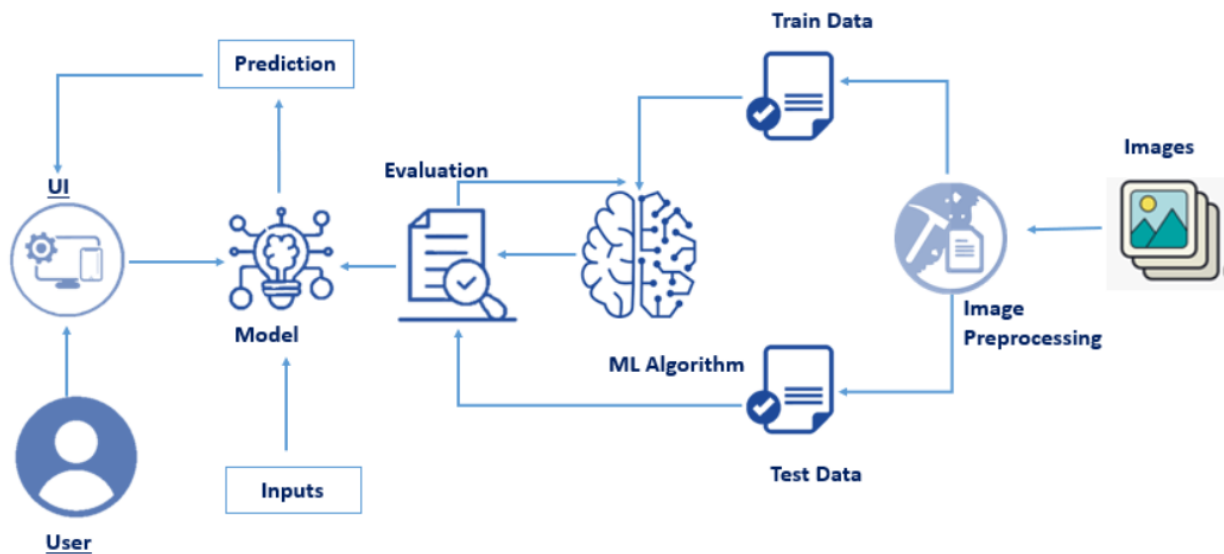
1.1 Existing problem and proposed soltuion

The researchers 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 of the 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 spirals and waves.
In existing system, PD is detected at the secondary stage only (Dopamine deficiency) which leads to medical challenges. Also doctor has to manually examine and suggest medical diagnosis in which the symptoms might vary from person to person so suggesting medicine is also a challenge. Thus the mental disorders are been poorly characterized and have many health complications. PD is generally diagnosed with the following clinical methods as, MRI or CT scan - Conventional MRI cannot detect early signs of Parkinson's disease PET scan - is used to assess activity and function of brain regions involved in movement SPECT scan - can reveal changes in brain chemistry, such as a decrease in dopamine This results in a high misdiagnosis rate (up to 25% by non-specialists) and many years before diagnosis, people can have the disease. Thus existing system is not effective in early prediction and accurate medicinal diagnosis to

the affected people

1.2 Proposed soltuion

By using machine learning techniques, the problem can be solved with minimal error rate. The voice dataset of Parkinson's disease from the UCI Machine learning library is used as input. Also our proposed system provides accurate results by integrating spiral drawing inputs of normal and Parkinson's affected patients. We propose a hybrid and accurate results analyzing patient both voice and spiral drawing data's. Thus combining both the results, the doctor can conclude normality or abnormality and prescribe the medicine based on the affected stage

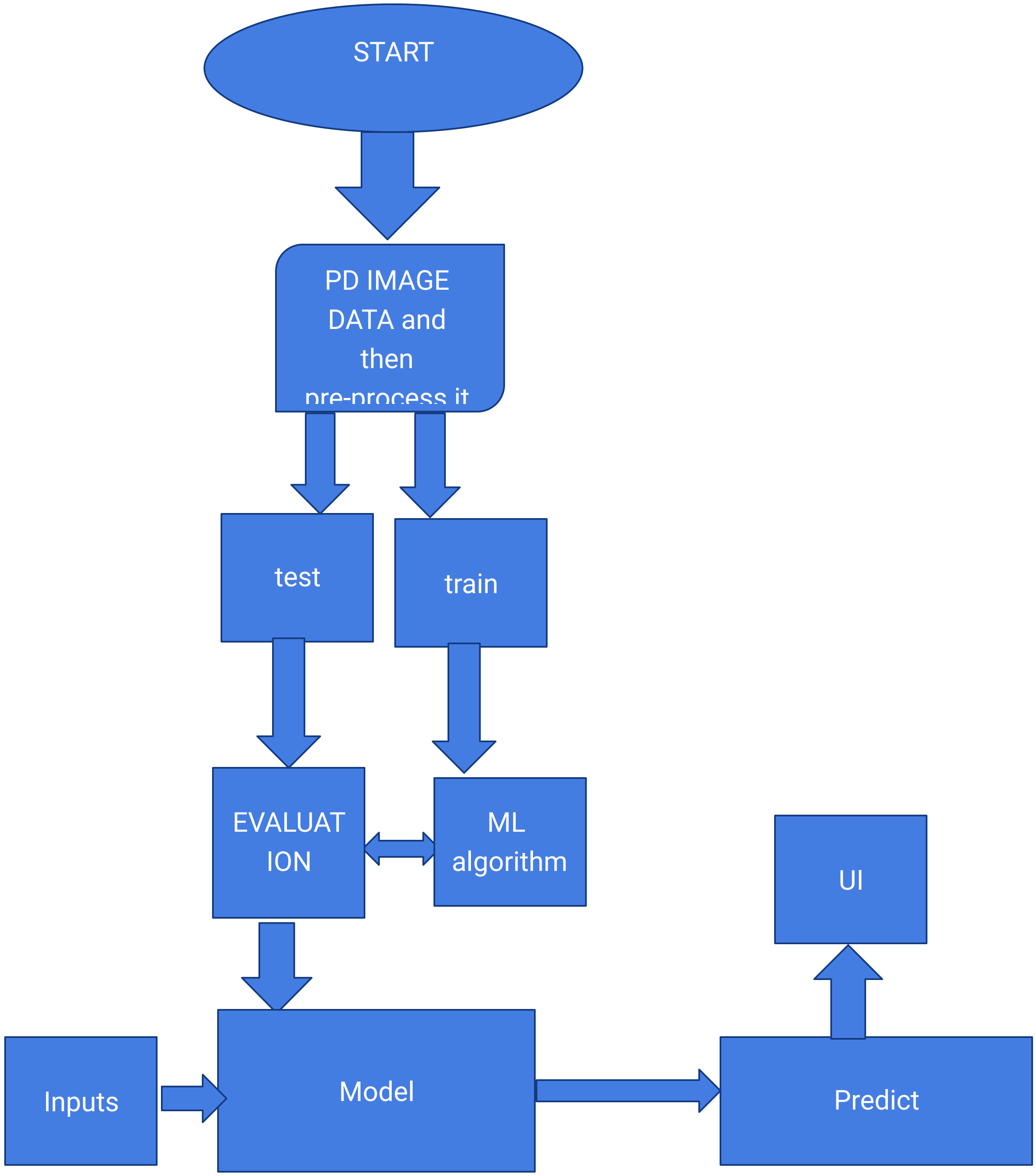


https://www.revistaclinicapsicologica.com/data-cms/articles/20210413120759pmSSCI-634.pdf

https://www.frontiersin.org/articles/10.3389/fnagi.2021.633752/full

3. THEORETICAL ANALYSIS

1.3 Block diagram


START
PD IMAGE DATA and then pre-process it
test
train
EVALUATION
ML algorithm
UI
Inputs
Model
Predict

1.2 SOFTWARE DESIGNING

For the project there need some software/packages:

Anaconda Navigator :

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook,QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

Also, requirement of packages is essential

- **Numpy**:
- It is an open-source numerical Python library. It contains a multidimensional array and matrix data structures and can be used to perform mathematical operations

- **Scikit-learn:**

- It is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy

- **Scikit-image**

- Scikit-image, or skimage, is an open-source Python package designed for image preprocessing.
- **Install imutils**
- Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, and displaying Matplotlib images easier with OpenCV
- Open anaconda prompt and type command

  "pip install imutils"
- **OpenCV**
- OpenCV is a library of programming functions mainly aimed at real-time computer vision. Here, OpenCV is used to capture frames by accessing the webcam in real-time.

- Open anaconda prompt and type command
  “pip install opencv-contrib-python”
- **Flask:**
  Web framework used for building Web applications

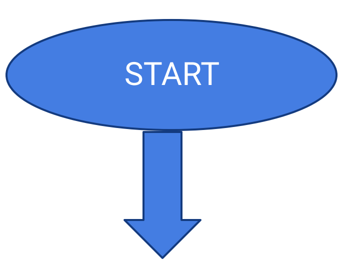
4. EXPERIMENTAL INVESTIGATIONS

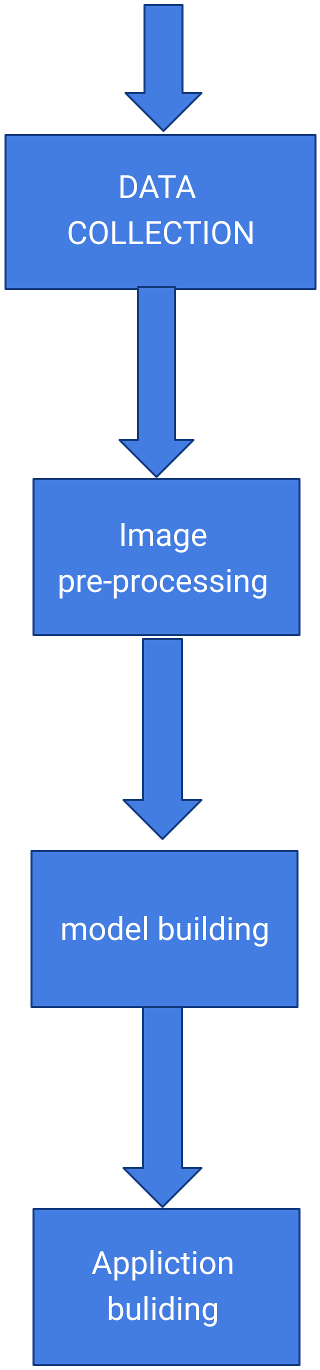
The analysis was made by creating project main folder and include below files in the folder

- Training folder contains:
- Parkinson_detect.py
- model file “parkinson.pkl”
- Flask App folder contains:
- static folder with the style sheets and the image required
- templates folder with the HTML pages
- app.py, a python script (Flask file) for server-side computing.

It was first concluded that Parkinson_detect.py is correctedly formated in jupyter notebook and model file is created by using testing and training appliances on the project. Then Flask application is made with static folder and html pages , for making those pages connecting, w have to make app.py and then finally result came whn all were attached carefully.

5. FLOWCHART





So, after starting we have to collect the data from the data set which will be required for running our project

- Collect the dataset or Create the dataset

Then after the collection of data set, we have perform pre-processing

- Importing the required libraries
- Loading Train data and Test data
- Quantifying images
- Label Encoding

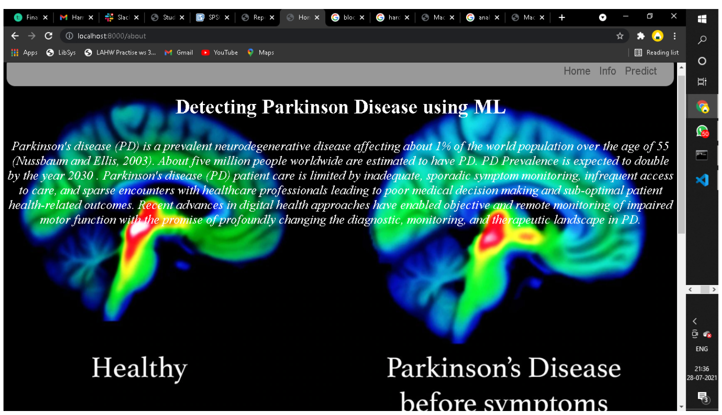
Now, model building task will begin

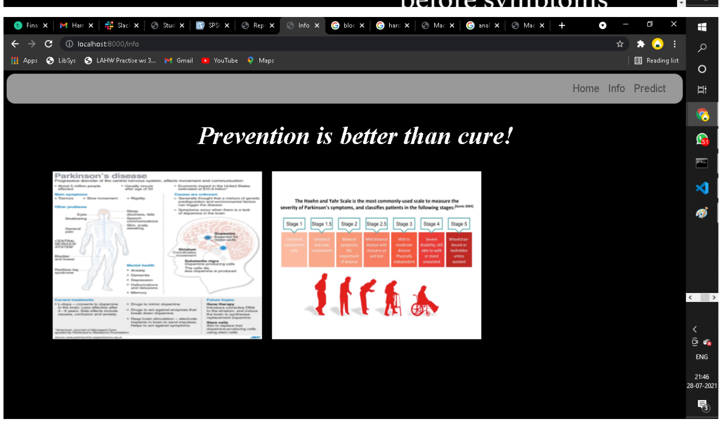
- Training the model
- Testing the model
- Model Evaluation
- Saving the model

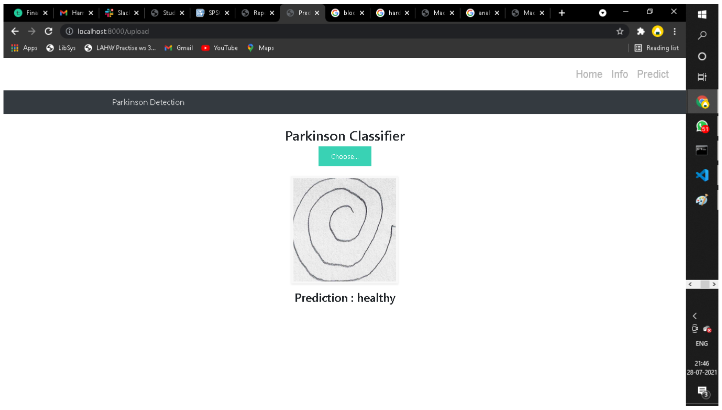
The last step, will be Application building

- Create an HTML file
- Build Python Code

6. RESULTS







7. ADVANTAGES AND DISADVANTAGES

ADVANATGES:

1. less time consuming
2. more accuracy in the model
3. easily implemented

DISADVANATGES:

1. packages to be installed
2. not too efficent
3. data collection difficult

8. APPLICATIONS

1. Web developemnt
2. mobile phones
3. app development
4. Electroencephalography (EEG)
5. Electromyography (EMG)
6. Tremor, Grip Strength

9. CONCLUSION

The researchers 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 of the 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 spirals and waves. To the best of our knowledge, the present study is the first review which included results from all studies that applied machine learning methods to the diagnosis of PD. Here, we presented included studies in a high-level summary, providing access to information including (a) machine learning methods that have been used in the diagnosis of PD and associated outcomes, (b) types of clinical, behavioral and biometric data that could be used for rendering more accurate diagnoses, (c) potential biomarkers for assisting clinical decision making, and (d) other highly relevant information, including databases that could be used to enlarge and enrich smaller datasets. In summary, realization of machine learning-assisted diagnosis of PD yields high potential for a more systematic clinical decision-making system, while adaptation of novel biomarkers may give rise to easier access to PD diagnosis at an earlier stage. Machine learning approaches therefore have the potential to provide clinicians with additional tools to screen, detect or diagnose PD.

10. Bibliography

1. **https://www.researchgate.net/publication/336513175_A_Systematic_review_on_Application_based_Parkinson's_disease_Detection_Systems#:~:text=The%20various%20parameters%20like%20Electroencephalography,disease%20progression%20in%20later%20stages.**
2. **https://www.frontiersin.org/articles/10.3389/fnagi.2021.633752/full**
3. **https://ieeexplore.ieee.org/document/8615607**
4. **https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8134676/**

APPENDIX

```python
# import the necessary packages
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix
from skimage import feature
from imutils import build_montages
from imutils import paths
import numpy as np
import cv2
import os
import pickle    #importing the pickle file
```

```python
def load_split(path):
    # grab the list of images in the input directory, then initialize
    # the list of data (i.e., images) and class labels
    imagePaths = list(paths.list_images(path))
    data = []
    labels = []

    # loop over the image paths
    for imagePath in imagePaths:
        # extract the class label from the filename
        label = imagePath.split(os.path.sep)[-2]

        # load the input image, convert it to grayscale, and resize
        # it to 200x200 pixels, ignoring aspect ratio
        image = cv2.imread(imagePath)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        image = cv2.resize(image, (200, 200))

        # threshold the image such that the drawing appears as white
        # on a black background
        image = cv2.threshold(image, 0, 255,
            cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

        # quantify the image
        features = quantify_image(image)

        # update the data and labels lists, respectively
        data.append(features)
        labels.append(label)

    # return the data and labels
    return (np.array(data), np.array(labels))
```

```python
print("[INFO] training model")
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)
```

```python
# loop over the testing samples
for i in idxs:
    # load the testing image, clone it, and resize it
    image = cv2.imread(testingPaths[i])
    output = image.copy()
    output = cv2.resize(output, (128, 128))

    # pre-process the image in the same manner we did earlier
    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]
```