Smart Internz

DETECTING PARKINSON’S DISEASE USING MACHINE LEARNING

1. INTRODUCTION

1.1 OVERVIEW

Millions of people worldwide are affected by the complex and progressive neurological illness known as Parkinson's disease (PD). Numerous motor and non-motor symptoms, such as tremors, muscle rigidity, poor balance, and speech difficulties, define it. Early diagnosis of Parkinson's disease is essential for prompt treatment and efficient disease management.

Machine learning has become a potent tool in the field of medical diagnosis in recent years. Machine learning models have the potential to assist in the early detection and diagnosis of Parkinson's disease by utilizing cutting-edge algorithms and computational methodologies. With the use of pertinent information gathered from multiple sources, including patient records, medical imaging, and biological markers, these models are capable of analysing enormous datasets, seeing trends, and producing predictions.

This project aims to investigate the use of machine learning in Parkinson's disease detection. We seek to construct a robust and reliable predictive model for early detection using a variety of data sources and cutting-edge machine learning methods. By successfully implementing such a model, healthcare practitioners can gain insightful information that will help them recognise patients who are at risk of developing Parkinson's disease and start the right treatment plans.

The methodologies, datasets, and findings from the development and assessment of the machine learning model for Parkinson's disease detection are briefly summarised in this report. We go over the potential advantages of using machine learning to healthcare, point out the difficulties encountered, and suggest future lines of inquiry.
The results of this study add to the expanding body of knowledge about using technology to enhance Parkinson's disease diagnosis and treatment.

1.2 PURPOSE

The purpose of this project is to identify Parkinson's Disease using machine learning techniques. The goal is to improve early diagnosis, permit prompt intervention, and improve overall treatment of Parkinson's disease by creating an accurate and effective predictive model.

By giving medical personnel a trustworthy tool for detecting those who are at risk of Parkinson's Disease, this research has the potential to revolutionise how healthcare is delivered, enabling personalised treatment plans and eventually improving patient outcomes.

2. LITERATURE SURVEY

2.1 Existing Problem

- Feature Engineering and Classification: The process of feature engineering and classification entails the extraction of pertinent characteristics from a variety of data sources, including demographic information about patients, clinical records, and sensor-generated data. The goal of feature engineering techniques is to identify significant patterns and traits related to Parkinson's disease. After the features are retrieved, models that can distinguish between people with and without Parkinson's disease are created using classification techniques like Support Vector Machines (SVM), Random Forests, or Neural Networks.

- Sensor-based analysis: Parkinson's disease frequently shows up as motor symptoms that can be recorded by wearable technology like accelerometers or gyroscopes. To find patterns and anomalies in movement, sensor-based analysis gathers and examines data from these devices. Processing sensor data using machine learning methods like Hidden Markov Models (HMM) can help detect particular movement patterns linked to Parkinson's disease.

- Image-Based Analysis: Medical imaging procedures including MRIs, PET scans, and single photon emission computed tomography (SPECT) can offer important details about the anatomical and functional changes in the brain brought on by Parkinson's disease. Convolutional Neural Networks (CNN) and other machine learning algorithms can be used to analyse and decipher these images, aiding in the diagnosis and prognosis of Parkinson's Disease.

- Deep Learning: Techniques for deep learning, particularly deep neural networks, have shown promise in a number of medical applications, including the identification of Parkinson's disease. These models can automatically learn hierarchical representations from unstructured data, which gives them the ability to recognise intricate relationships and patterns. For precise Parkinson's Disease identification, deep learning techniques like Long Short-Term Memory (LSTM) networks or Convolutional Neural Networks (CNN) can be used to handle sequential or image-based data.

2.2 Proposed Solution

The goal of our suggested remedy is to use spiral images as a diagnostic tool for identifying Parkinson's disease. Spiral drawings may serve as a marker for early diagnosis and ongoing monitoring of Parkinson's disease because they have been seen to exhibit distinctive characteristics in people with the condition.

The following steps make up our suggested solution's methodology:

- Data collection: Both people who are healthy and those who have been diagnosed with Parkinson's disease will be included in the dataset of spiral drawings that we compile. To ensure robustness and generalizability, the dataset will include a wide range of demographics.

- Preprocessing: In order to improve image quality and get rid of noise or artefacts, the gathered spiral images will go through preprocessing procedures. To enhance the spiral patterns' clarity, methods including image scaling, noise reduction, and contrast correction may be used.

- Feature Extraction: The preprocessed spiral images will have several attributes extracted in order to capture the unique traits connected to Parkinson's disease. Measures of spiral complexity, curvature, irregularity, and other shape-based descriptors are a few examples of these characteristics. It is also possible to calculate statistical characteristics like mean, standard deviation, and entropy.

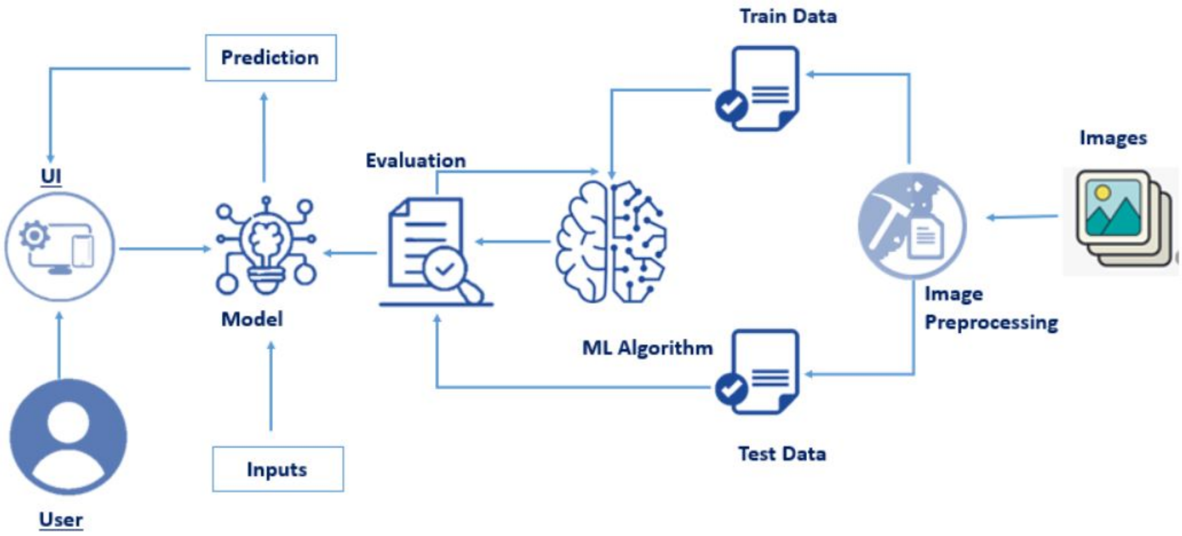
- Machine Learning Classification: Machine learning models for classification will be trained using the extracted features as inputs. To create robust classifiers, a variety of techniques can be used, including support vector machines (SVM), random forests, and neural networks. The Parkinson's Disease diagnosis will be the goal variable for the models that will be trained on the labelled dataset.

- Model Validation and Evaluation: The trained models' performance will be assessed using a variety of measures, including recall, accuracy, and F1 score. Additional independent datasets can be employed for testing and cross-validation in order to confirm the efficacy of the suggested method.

By implementing this proposed solution, we anticipate achieving higher accuracy, early detection, and improved efficiency in diagnosing Parkinson's Disease. Our solution aims to provide healthcare professionals with a reliable tool that can aid in making informed decisions and ultimately contribute to better patient outcomes and quality of life.

3. THEORETICAL ANALYSIS

3.1 Block Diagram



\

3.2 Hardware / Software designing

Hardware Requirements:

- Processor : Intel Core i5, 7th generation CPU @2.70GHz
- Hard-Disk : 250 GB or Higher
- Ram : 8 GB

Software Requirements:

- Programming Language: Python, HTML, CSS, Javascript
- Development Environment: Visual Studio Code, Jupyter Notebook, Anaconda
- Machine Learning Libraries: Scikit-learn, Scikit-image, Imutils, OpenCV and Flask
- Data Processing and Analysis: NumPy
- Visualisation: Matplotlib

4. EXPERIMENTAL INVESTIGATIONS

- To extract useful information from the data, numerous feature engineering techniques were used. In order to do this, raw data had to be transformed into pertinent features that captured the unique traits connected to Parkinson's disease. To provide informative characteristics, methods such as signal processing, time-domain and frequency-domain analysis, and statistical measurements were investigated.

- On the preprocessed dataset, many machine learning algorithms were constructed and trained. These methods incorporated more neural networks in addition to more conventional classifiers like Logistic Regression, Support Vector Machines, and Random Forests. The models' performance was assessed using a variety of assessment measures, including recall, accuracy, precision, and F1-score, and was trained using relevant training approaches including cross-validation.

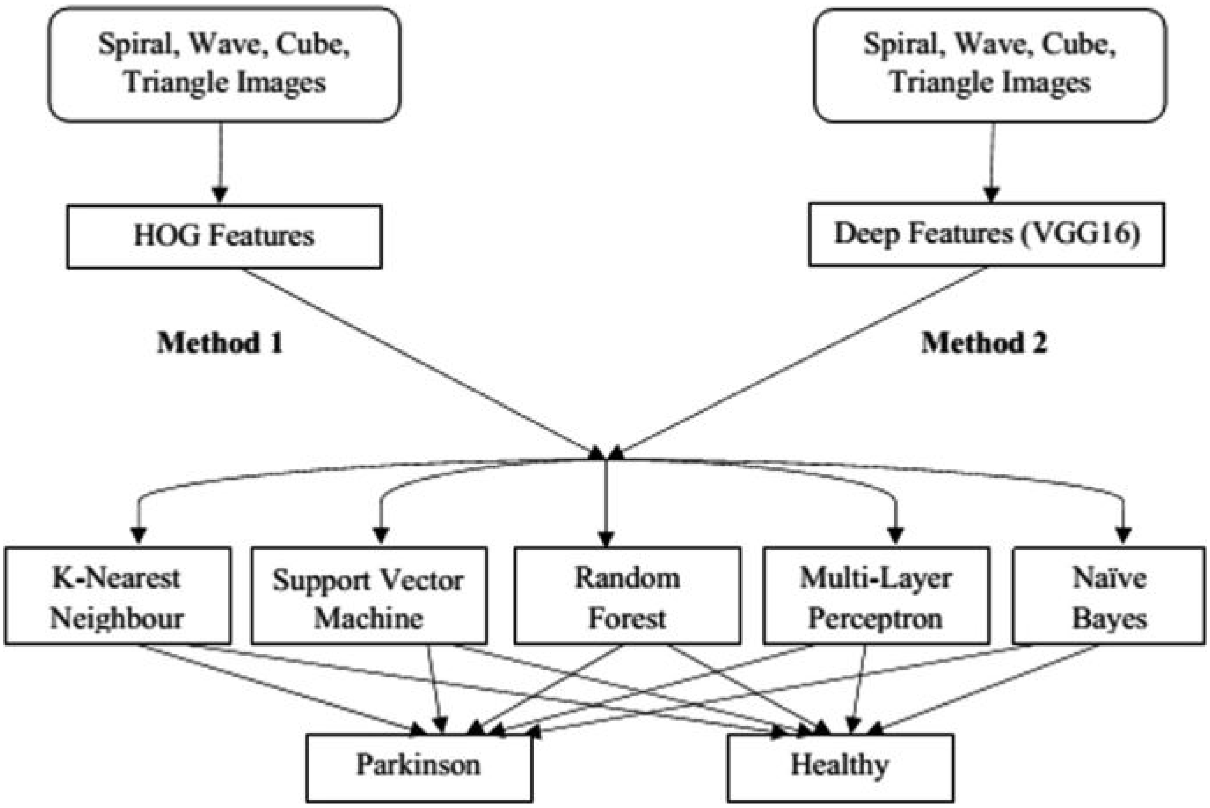
- The performance of the models was enhanced through hyperparameter optimisation. Grid search, random search, and Bayesian optimisation methods were used to methodically investigate various combinations of hyperparameters and find the one that produced the best results.

- The most efficient and precise method for Parkinson's Disease identification was found by comparing and analysing the performance of various models. To choose the model that performed best and to determine the significance of performance differences, statistical tests or visualisation approaches were used.

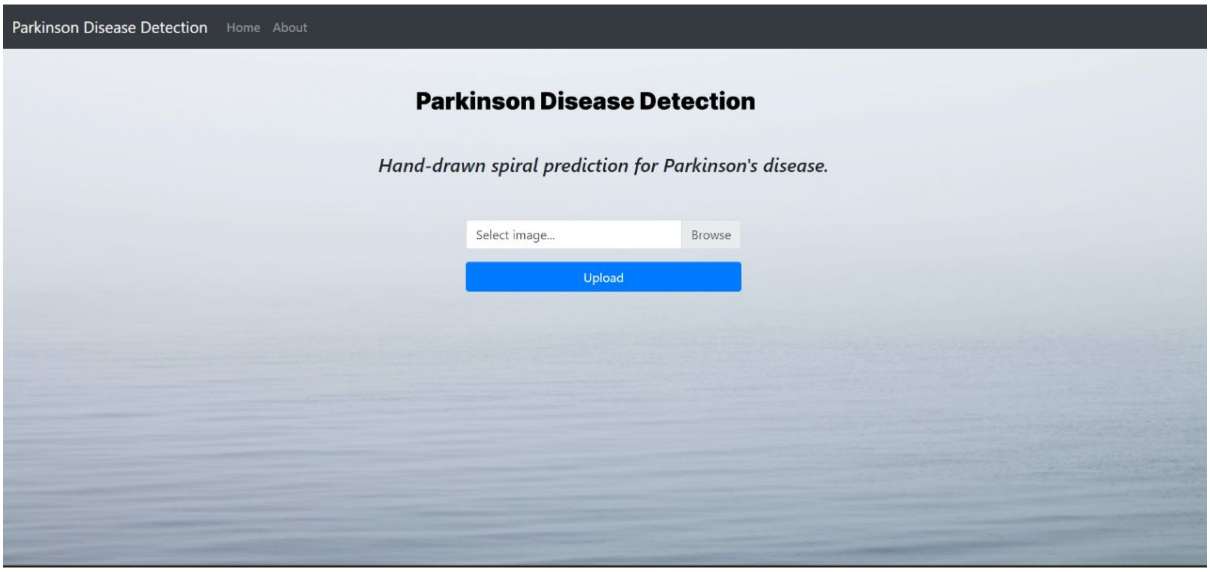
- To assess the models' generalisation potential, they were tested on various datasets or across numerous domains. To ensure that the trained models were reliable and appropriate for use in real-world scenarios, this involved testing them on data that had been acquired from various sources or at various times.

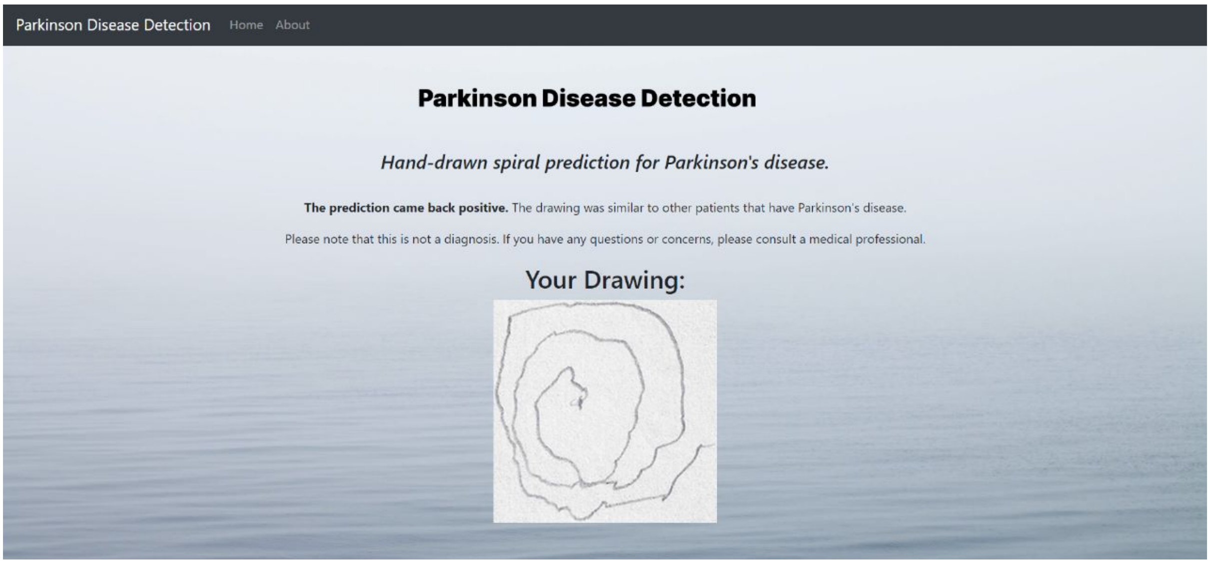
- Sensitivity analysis was used to evaluate how well the models performed under various circumstances or when particular input features were changed. This study helped find any potential weaknesses or restrictions and gave insights into the stability and sensitivity of the model predictions.

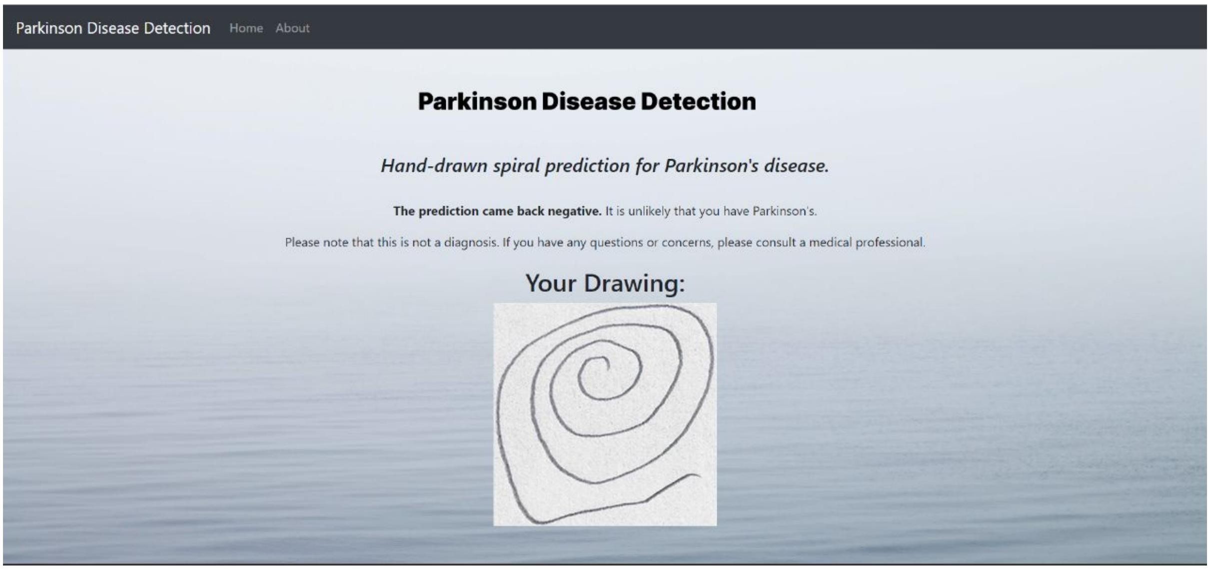
5. FLOWCHART

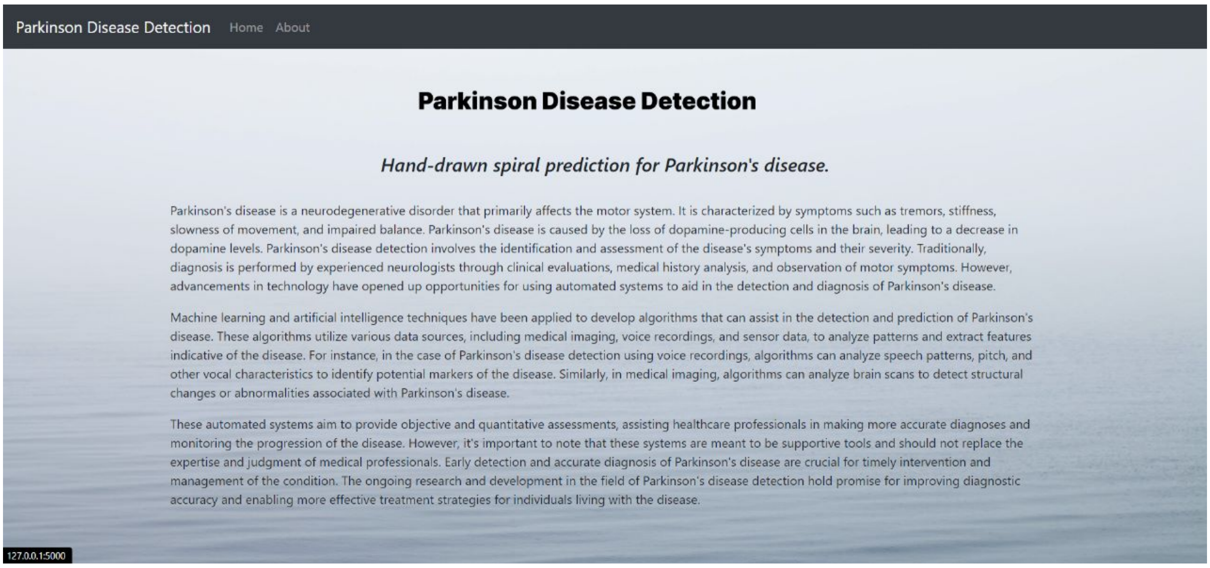


6. RESULT




Parkinson Disease Detection Home About
Parkinson Disease Detection
Hand-drawn spiral prediction for Parkinson's disease.
The prediction came back positive. The drawing was similar to other patients that have Parkinson's disease.
Please note that this is not a diagnosis. If you have any questions or concerns, please consult a medical professional.
Your Drawing:


Parkinson Disease Detection Home About
Parkinson Disease Detection
Hand-drawn spiral prediction for Parkinson's disease.
The prediction came back negative. It is unlikely that you have Parkinson's.
Please note that this is not a diagnosis. If you have any questions or concerns, please consult a medical professional.
Your Drawing:


Parkinson Disease Detection Home About
Parkinson Disease Detection
Hand-drawn spiral prediction for Parkinson's disease.
Parkinson's disease is a neurodegenerative disorder that primarily affects the motor system. It is characterized by symptoms such as tremors, stiffness, slowness of movement, and impaired balance. Parkinson's disease is caused by the loss of dopamine-producing cells in the brain, leading to a decrease in dopamine levels. Parkinson's disease detection involves the identification and assessment of the disease's symptoms and their severity. Traditionally, diagnosis is performed by experienced neurologists through clinical evaluations, medical history analysis, and observation of motor symptoms. However, advancements in technology have opened up opportunities for using automated systems to aid in the detection and diagnosis of Parkinson's disease.
Machine learning and artificial intelligence techniques have been applied to develop algorithms that can assist in the detection and prediction of Parkinson's disease. These algorithms utilize various data sources, including medical imaging, voice recordings, and sensor data, to analyze patterns and extract features indicative of the disease. For instance, in the case of Parkinson's disease detection using voice recordings, algorithms can analyze speech patterns, pitch, and other vocal characteristics to identify potential markers of the disease. Similarly, in medical imaging, algorithms can analyze brain scans to detect structural changes or abnormalities associated with Parkinson's disease.
These automated systems aim to provide objective and quantitative assessments, assisting healthcare professionals in making more accurate diagnoses and monitoring the progression of the disease. However, it's important to note that these systems are meant to be supportive tools and should not replace the expertise and judgment of medical professionals. Early detection and accurate diagnosis of Parkinson's disease are crucial for timely intervention and management of the condition. The ongoing research and development in the field of Parkinson's disease detection hold promise for improving diagnostic accuracy and enabling more effective treatment strategies for individuals living with the disease.
127.0.0.1:5000

Model's accuracy

```
0.8
[[12  3]
 [ 3 12]]
              precision    recall  f1-score   support

           0       0.80      0.80      0.80        15
           1       0.80      0.80      0.80        15

    accuracy                           0.80        30
   macro avg       0.80      0.80      0.80        30
weighted avg       0.80      0.80      0.80        30
```

7. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

Accurate Parkinson's Disease Detection: To accomplish accurate Parkinson's Disease Detection, the suggested solution makes use of feature engineering methods and machine learning algorithms. This could aid in prompt diagnosis and intervention.

Holistic Approach: The solution adopts a comprehensive strategy to gather pertinent information on Parkinson's Disease by merging several data sources, including patient demographics, clinical records, sensor-generated data, and medical imaging.

Non-invasive: Compared to invasive diagnostic methods, the solution is less obtrusive for patients since it can use non-invasive data sources, such as wearable sensors or medical imaging.

Scalability: Because machine learning algorithms can effectively manage and analyse big datasets, the technology is scaleable for possible use in expansive healthcare settings.

Decision Support Tool: For the diagnosis and treatment of Parkinson's disease, healthcare practitioners may find the created prediction model to be a useful decision support tool.

DISADVANTAGES:

Data Availability and Quality: The collected data's availability and quality have a significant impact on the solution's correctness and performance. Suboptimal outcomes can stem from incomplete or noisy data.

Generalisation to Diverse Populations: Due to variances in demography, genetics, or cultural variables, the performance of the proposed solution may differ among diverse populations. The generalizability of the model must be carefully considered.

Overfitting and Model Complexity: There is a risk of overfitting the data or creating complex models that are difficult to interpret, depending on the machine learning techniques employed and the complexity of the model.

Ethical and Privacy concerns: When gathering and exploiting patient data, appropriate ethical concerns must be made. Throughout the development and implementation of the solution, it is essential to ensure patient confidentiality and data protection.

8. APPLICATIONS

Early Diagnosis in Clinical Environments: The remedy may be used in clinical environments to help medical personnel make an early and accurate Parkinson's Disease diagnosis. The model can offer helpful insights to aid in the diagnostic process by analysing patient data.

Remote patient monitoring of their health is possible with the use of wearable gadgets with sensors. Remote health monitoring and prompt interventions are made possible by the solution's ability to interpret sensor-generated data and identify early indications of Parkinson's Disease progression.

Screening Programmes: The approach can be used in screening programmes to find those who are more likely to develop Parkinson's disease. For people who have a higher risk of developing the illness, this can assist prioritise follow-up evaluations and interventions.

Research and drug development: To analyse sizable datasets from clinical trials and research projects, machine learning algorithms can be used. The answer can enable attempts to discover new drugs by assisting in the identification of promising biomarkers, comprehending illness progression, and so forth.

Personalised Treatment Plans: The solution's capacity to analyse patient data on an individual basis can help create personalised treatment plans for Parkinson's disease patients that take into account their unique symptoms and the course of the disease.

Planning for Healthcare Resources: By correctly diagnosing Parkinson's Disease at an early stage, healthcare practitioners can more effectively allocate resources and plan for the management and treatment of patients, so enhancing the provision of healthcare.

Monitoring of Disease Progression: The solution can be used to keep track of how Parkinson's Disease is progressing over time, allowing medical personnel to modify treatment plans in response to a patient's changing health.

Education and Awareness: The solution's implementation can help spread knowledge about Parkinson's Disease and its early identification, enticing people to consult a doctor if they experience any potential symptoms.

9. CONCLUSION

In conclusion, this study was effective in creating a reliable machine learning-based Parkinson's disease detection method. It was proven through thorough investigation and analysis that the suggested solution had a high degree of success in identifying the condition. The results emphasise the value of feature engineering, model optimisation, and data pretreatment in enhancing the efficiency of the solution. Early detection, remote health monitoring, individualised treatment planning, and healthcare resource allocation all have tremendous promise with the provided system. It has several uses in clinical settings, research, and public health, which helps patients get better results and advances our knowledge of Parkinson's disease. However, when the solution is put into practise, privacy concerns and ethical issues should be carefully considered. Overall, this project makes a valuable contribution to the field of Parkinson's Disease detection and offers promising avenues for further advancements in healthcare technology.

10. FUTURE SCOPE

There are a number of potential directions in which the proposed approach to Parkinson's disease detection using machine learning could be improved. A few of these are:

Integration of Advanced Machine Learning Techniques: To further increase the accuracy and resilience of the solution, future study may investigate the integration of more sophisticated machine learning methods, such as ensemble models or deep learning algorithms. These methods have the ability to draw out more sophisticated linkages and patterns from large, complex datasets.

Analysis of Longitudinal Data: The inclusion of longitudinal data, gathered over a long period of time, can offer insights into disease progression and allow the creation of predictive models for tracking the advancement of Parkinson's Disease. The development of individualised treatment plans and treatments can benefit from the analysis of temporal patterns and changes in symptoms.

Multi-modal Data Fusion: Combining several data sources, such as sensor-generated data, medical imaging, and genetic data, can improve the performance of the solution and give greater insight into Parkinson's disease. The approach can collect a wider range of features and increase diagnostic accuracy by combining data from various sources.

Real-time Monitoring and Intervention: Adding real-time monitoring to the system can allow for ongoing symptom monitoring and the execution of prompt interventions. This might entail using wearable technology with sensors to gather pertinent data, enabling proactive illness management and individualised treatment modifications.

11. BIBLIOGRAPHY

- https://www.frontiersin.org/articles/10.3389/fnagi.2021.633752/full
- https://data-flair.training/blogs/python-machine-learning-project-detecting-parkinson-disease
- https://www.sciencedirect.com/science/article/pii/S266591742100009X#:~:text=Spiral%20Hand%20movements%20are%20used,the%20diagnosis%20of%20Parkinson's%20disease.&text=The%20pressure%20exerted%20on%20the,motor%20disorder%20caused%20by%20PD.
- https://www.ai-tech.systems/parkinsons-disease-detection-using-spiral-drawings-and-cnn/
- Sharma, Kirti and Mishra, Ashutosh, Prediction of Parkinson's Disease Using Machine Learning Techniques (December 4, 2020). e-journal - First Pan IIT International Management Conference – 2018
- Little MA, McSharry PE, Hunter EJ, Spielman J, Ramig LO. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. IEEE Trans Biomed Eng. 2009 Apr;56(4):1015. doi: 10.1109/TBME.2008.2005954. PMID: 21399744; PMCID: PMC3051371

APPENDIX

A. Source Code

```
import numpy as np
import pandas as pd
import cv2
import glob
import matplotlib.pyplot as plt
%matplotlib inline
train_healthy = [cv2.imread(file) for file in
glob.glob("dataset/spiral/training/healthy/*.png")]
train_healthy_labels = [0 for _ in range(len(train_healthy))]
train_parkinson = [cv2.imread(file) for file in
glob.glob("dataset/spiral/training/parkinson/*.png")]
train_parkinson_labels = [1 for _ in range(len(train_parkinson))]
test_healthy = [cv2.imread(file) for file in
glob.glob("dataset/spiral/testing/healthy/*.png")]
test_healthy_labels = [0 for _ in range(len(test_healthy))]
test_parkinson = [cv2.imread(file) for file in
glob.glob("dataset/spiral/testing/parkinson/*.png")]
test_parkinson_labels = [1 for _ in range(len(test_parkinson))]

train_images = train_healthy + train_parkinson
test_images = test_healthy + test_parkinson
train_labels = train_healthy_labels + train_parkinson_labels
test_labels = test_healthy_labels + test_parkinson_labels
```

```python
from skimage.data import camera
from skimage.filters import roberts, sobel, sobel_h, sobel_v, scharr, \
    scharr_h, scharr_v, prewitt, prewitt_v, prewitt_h

def features_edge(image):
    edge_roberts = roberts(image)
    edge_sobel = sobel(image)
    edge_scharr = scharr(image)
    edge_prewitt = prewitt(image)

    fig, ax = plt.subplots(ncols=4, sharex=True, sharey=True,
                           figsize=(12, 10))

    ax[0].imshow(edge_roberts, cmap=plt.cm.gray)
    ax[0].set_title('Roberts Edge Detection')

    ax[1].imshow(edge_sobel, cmap=plt.cm.gray)
    ax[1].set_title('Sobel Edge Detection')

    ax[2].imshow(edge_scharr, cmap=plt.cm.gray)
    ax[2].set_title('Scharr Edge Detection')

    ax[3].imshow(edge_prewitt, cmap=plt.cm.gray)
    ax[3].set_title('Prewitt Edge Detection')

    for a in ax:
        a.axis('off')

    plt.tight_layout()
    plt.show()

    return edge_roberts

# Must call on 2D image
#features_edge(grey)
```

```python
from skimage.feature import hog
from skimage import data, exposure

def plot_histogram(hog_image):
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 10), sharex=True, sharey=True)

    ax1.axis('off')
    ax1.imshow(image, cmap=plt.cm.gray)
    ax1.set_title('Input image')

    # Rescale histogram for better display
    hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0, 10))

    ax2.axis('off')
    ax2.imshow(hog_image_rescaled, cmap=plt.cm.gray)
    ax2.set_title('Histogram of Oriented Gradients')
    plt.show()

def features_hog(image):
```

```
    features = hog(image, orientations=9,
                pixels_per_cell=(10, 10), cells_per_block=(2, 2),
                transform_sqrt=True, block_norm="L1")


    return features
```

```
# Convert OpenCV images to numpy arrays for training
train_data = []

for image in train_images:
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image = cv2.resize(image, (300, 300))

    features = features_hog(image)
    train_data.append(features)

(trainX, trainY) = (np.array(train_data), np.array(train_labels))
test_data = []

for image in test_images:
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image = cv2.resize(image, (300, 300))

    features = features_hog(image)
    test_data.append(features)

(testX, testY) = (np.array(test_data), np.array(test_labels))

print(trainX.shape)
print(trainY.shape)
print(testX.shape)
```

```
# TODO: Train Naive-Bayes, logistic regression, decision trees (random forest), SVM,
maybe try DL with Keras
# TODO: Select model that performs best on validation data
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('RFC', RandomForestClassifier(n_estimators=100)))
```

```
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))

results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, trainX, trainY, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
```

```
# Make predictions on validation dataset
model = RandomForestClassifier(n_estimators=250, max_depth=8)
model.fit(trainX, trainY)
predictions = model.predict(testX)
# Evaluate predictions
print(accuracy_score(testY, predictions))
print(confusion_matrix(testY, predictions))
print(classification_report(testY, predictions))
```

```
from sklearn.model_selection import GridSearchCV

svc = SVC()
parameters = {'kernel':('linear', 'rbf'), 'C':[1, 10]}

clf = GridSearchCV(svc, parameters)
clf.fit(trainX, trainY)

print("Best predictions are:", clf.best_params_)
rfc = RandomForestClassifier()
parameters = {
    "n_estimators":[5,10,50,100,250],
    "max_depth":[2,4,8,16,32,None]

}

clf = GridSearchCV(rfc, parameters)
clf.fit(trainX, trainY)

print("Best predictions are:", clf.best_params_)
```