DETECTION OF PARKINSON DISEASE USING HANDWRITING ANALYSIS AND MACHINE LEARNING

Submitted by,

Name	Roll No.	Registration No.
Adrija Saha	11700119084	036547 of 2019-2020
Gautami Sinha	11700119099	035597 of 2019-2020
Meghna Chakraborty	11700119095	036533 of 2019-2020
Arnab Das	11700119112	036384 OF 2019-20

UNDER THE SUPERVISION OF
Mr. Rajib Saha
[Assistant Professor, HOD (CSE), RCC Institute of Information Technology]

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING, RCC INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RCC INSTITUTE OF INFORMATION TECHNOLOGY
[Affiliated to West Bengal University of Technology] CANAL SOUTH ROAD, BELIAGHATA, KOLKATA-700015

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING RCC INSTITUTE OF INFORMATION TECHNOLOGY

TO WHOM IT MAY CONCERN

I hereby recommend that the Project entitled "**DETECTION OF PARKINSON DISEASE USING HANDWRITING ANALYSIS AND MACHINE LEARNING"** prepared under my supervision by Adrija Saha (Roll No.: 11700119084), Gautami Sinha (Roll No.:11700119099), Meghna Chakraborty (Roll No.:11700119095), Arnab Das (Roll No.:11700119112), of B.Tech (8th Semester), may be accepted in partial fulfillment for the degree of **Bachelor Of Technology in Computer Science & Engineering** under Maulana Abul Kalam Azad University of Technology (MAKAUT).

……

Project Supervisor and HOD, Department of Computer Science and Engineering, RCC Institute of Information Technology, Kolkata – 700015

CONTENTS

CHAPTER-4

CERTIFICATE OF APPROVAL

The foregoing Project is hereby accepted as a credible study of an engineering subject carried out and presented by Adrija Saha (Roll No.: 11700119084), Gautami Sinha (Roll No.:11700119099), Meghna Chakraborty (Roll No.:11700119095), Arnab Das (Roll No.:11700119112), in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

Name of the Examiner **Signature with Date**

1. …… ……

2. …… ……..

3. …… ……..

4. …… ……..……

ACKNOWLEDGEMENT

We acknowledge our overwhelming gratitude & immense respect to our revered guide, Mr. Rajib Saha (Assistant Professor, HOD CSE, RCCIIT) under whose scholarly guideline, constant encouragement & untiring patience; we have the proud privilege to accomplish this entire project work. We feel enriched with the knowledge & sense of responsible approach we inherited from our guide & shall remain a treasure in our life.

Date: ……..

Adrija Saha
Reg. No. : 036547
Roll No.: 11700119084

Meghna Chakraborty
Reg. No. : 036533
Roll No.: 11700119095

Gautami Sinha
Reg. No. : 035597
Roll No.: 11700119099

Arnab Das
Reg. No. : 036384
Roll No.: 11700119112

ABSTRACT

Parkinson's disease is a neurological disease defined by tremors, stiffness, and bradykinesia. Early detection and precise disease monitoring are critical for efficient disease management and therapy. In recent years, there has been a surge of interest in non-invasive ways for detecting Parkinson's disease, and one potential approach is the study of handwriting patterns using machine learning algorithms. This study investigates Parkinson's disease detection using handwriting analysis and machine learning. As essential markers, the Archimedean spiral test, wave detection, and Micrographia are examined. The dataset, which includes handwriting examples from Parkinson's disease sufferers and healthy persons, is trained using Support Vector Machines (SVM). The SVM model distinguishes between Parkinson's disease and non-Parkinson's disease samples with excellent accuracy.. This study showcases the potential of non-invasive early detection and monitoring of Parkinson's disease using handwriting analysis and machine learning techniques. The findings contribute to the development of a cost-effective and accessible tool for Parkinson's disease detection.

LIST OF ABBREVIATIONS

Abbreviation	Full Form / Meaning
PD	Parkinson's diseases
AD	Alzheimer's disease
SVM	Support Vector Machine
CNN	Convolutional Neural Network
API	Application Programming Interface
NLD	Non-linear dynamic
MFCCs	Most-frequently considered coefficients
SVC	Support Vector Classifier

LIST OF FIGURES

I

1.1. INTRODUCTION

Parkinson’s disease (PD), a chronic and progressive neurodegenerative disorder that primarily affects the motor system is the second most common neurodegenerative disorder. This disease of ageing and Movement disorder is clinically characterized by tremor, bradykinesia, rigidity, and postural instability [1]. Neurodegenerative diseases including Alzheimer’s disease (AD) affect the structure and functions of brain regions resulting in a progressive cognitive, functional, and behavioral decline. According to research, PD is caused by the degeneration of the dopaminergic nigrostriatal neurons of the basal ganglia, resulting primarily in motor deficits: akinesia, bradykinesia, rigidity, and tremor are
Typically observed [2]. The most obvious initial symptoms are tremors, stiffness, slowness of movement, and difficulty walking. Cognitive and behavioral problems can also co-occur with depression, anxiety, and lethargy in many people with Parkinson's disease.

As the motor neurons get affected, the patient suffers during performing daily activities, such as writing. Writing is a basic skill that we take for granted - until it gets tough. When writing becomes difficult, it can greatly affect an individual's quality of life. Simple tasks like signing your name on a form, making a shopping list, or leaving a quick note for a family member can become daunting.

Handwriting in Parkinson's disease (PD) frequently varies and might be a useful tool for measuring the motor symptoms associated with the disorder. Handwriting is commonly impacted in people with Parkinson's disease in the following ways:
Micrographia: The steady loss in handwriting size over time is referred to as micrographia. Individuals with Parkinson's disease may find it increasingly difficult to maintain their regular handwriting size, resulting in smaller and more constricted writing. This alteration is frequently one of the first visible indicators of Parkinson's disease i.e. : -

1. Tremors and Oscillations: Tremors are involuntary rhythmic movements that can impair fine motor function caused by Parkinson's disease. These tremors may appear as irregular and wobbly lines or loops in the handwriting, making the writing appear less fluid and smooth.

2. Reduced Pen Pressure: People with Parkinson's disease often exhibit a reduction in pen pressure while writing. This results in lighter strokes and a loss of the characteristic variation in line thickness typically seen in handwriting.

3. Irregular Letter Formation: The precision and consistency of letter formation can be compromised in individuals with PD. Letters may appear distorted or uneven, lacking the usual symmetry and regularity.

4. Decreased Line Alignment: Maintaining a straight line or consistent alignment can become challenging for individuals with Parkinson's disease. Handwriting may become more slanted or exhibit a wavering baseline.

It is important to highlight that the extent and precise aspects of handwriting changes might differ between Parkinson's disease patients. Not all people with Parkinson's disease experience the same level of handwriting degradation, and other factors such as medication, exhaustion, or coexisting diseases can also have an impact on handwriting abilities.

Analyzing these handwriting changes, as well as other motor symptoms, can provide important insights for early detection and monitoring of Parkinson's disease. Machine learning algorithms can be used to analyse and discover trends in handwriting data, allowing for the creation of objective and quantifiable tools to aid in Parkinson's disease diagnosis and progression tracking. The characteristics of the input data are the measurable properties of the observations that one uses to analyse or classify these data instances. The task of feature extraction is to select relevant features that differentiate well between versions and are independent of each other.
The choice of feature extraction method is perhaps the most important factor to achieve high recognition performance. There are a large number of methods for extracting features from images, each with different features. Various features are used here to predict the percentage of disease from character patterns like angle of inclination, readable percentage and line quality, such as: -

1. Skewness: Skew is defined as the alignment of the text lines and words with respect to the horizontal direction. Skew can be either clockwise, called positive skew or anti-clockwise, called negative skew.[3]. It is the alignment of lines of text and words with respect to the horizontal direction. The tilt can be clockwise, called positive or counterclockwise deviation, called negative deviation. Another type of bias is the wavy bias/ wavy skew, an example of which is illustrated below.The angle of inclination is θ (theta).

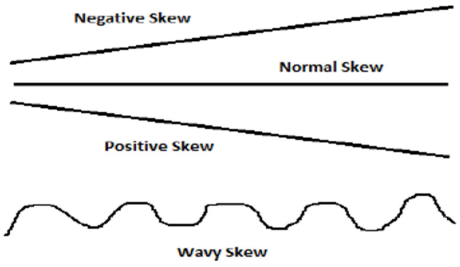


Fig 1.1.1 : Orientation of skew

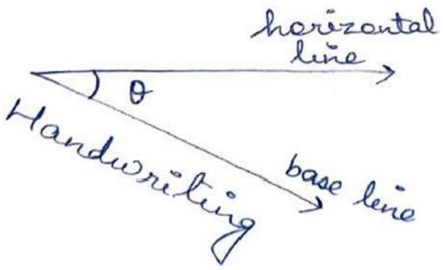


Fig 1.1.2: Positive skew

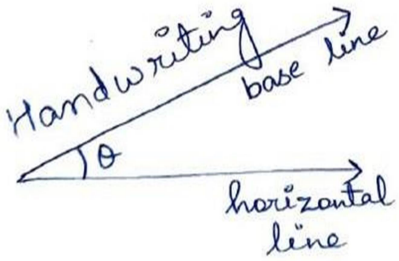


Fig 1.1.3: Negative skew

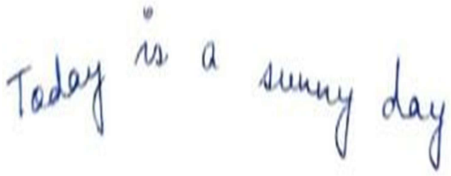


Fig 1.1.4: Wavy skew

Previous methods for skew detection and normalisation of document images have been proposed in previous research [4],[5],[6],[7],[8], such as Global Houng Transform [4], Local Houng Transform [2], Projection Histogram [2], Method of Least Squares [4], and Word Centroid Least Squares[4]. The projection histogram [4] can handle minor skew angles ranging from -10 to +10 degrees, as opposed to approaches such as the Hough Transform [4], which has a high complexity cost due to the enormous number of computations. We have developed our own algorithm where we are comparing our dataset (that comprises both healthy and PD patients), with the average range of angle of skew and finally detecting the patients with PD.

2. Archimedean Spiral: This paper proposes a system design to analyze Archimedes Spiral drawing patterns in patients with Parkinson's disease (PD) and healthy individuals. The Static spiral test is employed, where patients are asked to draw a continuous spiral. The resulting spiral provides insights into tremor presence and severity. This test aids in diagnosis and treatment planning for tremors, as the continuous movement allows for easier observation of tremors by doctors and physicians to assess the severity of tremors, ranging from obvious to subtle manifestations.

3. Micrographia: Micrographia is a common disabling symptom of Parkinson's disease (PD) and is defined as “an impairment of a fine motor skill manifesting mainly as a progressive or stable reduction in amplitude during a writing task” (Wagle Shukla et al., 2012) [9]. There are several types of Micrographia that can occur in people with Parkinson's disease. These are some examples:

- Size reduction: Micrographia frequently manifests as a steady decrease in the size of handwriting over time. Individuals may find it increasingly difficult to maintain regular letter sizes as the condition develops, resulting in smaller and more cramped handwriting.

- Reduced control and accuracy: Parkinson's disease can impair fine motor control, resulting in a loss of writing precision. Hand movements may become tremulous and less fluid, resulting in abnormalities in letter shapes and writing patterns overall.

- Letter crowding: Letter crowding is a type of Micrographia in which letters are written excessively close together, making the text appear densely packed. This can make reading and deciphering written material difficult. Content.

- Reduced line spacing: Another feature of Micrographia is reduced line spacing. Individuals may experience trouble maintaining constant lengths between lines of text as the condition progresses, resulting in a more compressed look.

The study focuses on the implementation of an algorithmic strategy that makes use of image processing techniques. We have converted the input image to grayscale, added contrast enhancement and noise reduction filters to improve image quality, and finally computed the horizontal projection profile of the handwriting are all phases in the process. The system then finds lines in the image and estimates their average height.

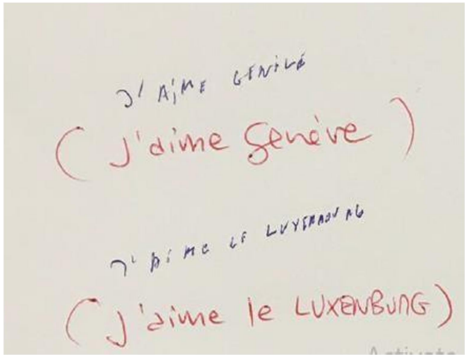


Fig 1.1.5: Handwriting sample of a patient suffering from PD showing presence of micrographia

4. Wave: Detecting the presence of Parkinson's disease can be achieved through analyzing the characteristic wave patterns generated by a person's drawing movements. Parkinson's disease affects the nervous system, causing motor abnormalities, and this can be reflected in the fluidity and regularity of a person's hand movements. By examining the waveforms created during a drawing task, experts can identify subtle irregularities that may indicate the presence of Parkinson's. These irregularities often manifest as tremors, jerky motions, or inconsistencies in the lines and curves drawn. By scrutinizing the unique signatures embedded in the wave patterns, healthcare professionals can gain valuable insights into a patient's motor function and help diagnose Parkinson's disease at an early stage, facilitating timely intervention and treatment.

Following this analysis we have achieved an accuracy of 80% for the analysis using the Archimedean Spiral, 98% for the Wave detection or comparison with the healthy person’s dataset and Parkinson’s Patients Datasets.

1.2. LITERATURE REVIEW

Handwriting is a complex neuro motor task. Over the last thirty years, it has been analyzed for identifying patients having Parkinson's Disease (PD) and considerable improvements have been brought about in this field through several research works. PD is a neuro-degenerative disease. Over time, as the disease progresses, several motor and non-motor symptoms surface. It comes with symptoms such as bradykinesia, rigidity, tremor, hypomania and others. Early stages of PD can be identified from finger and wrist movement. Handwriting has served as one of the dominant bioindicators for identification of neuro-degenerative diseases. It has been analyzed both statically as well as dynamically. [10]

One of the most common features of handwriting analysis is skew detection. Skew refers to the alignment of text with respect to the horizontal direction. It can be of three types, negative skew, positive skew and wavy. These types depend on the angle of inclination of the letters with respect to the horizontal line. It has been observed that generally patients with PD tend to have variable skew in their handwriting samples. The most common techniques of skew detection include Hough Transformation Method, Dhandra's approach. A new research using the blobbing method where the text is blobbed word by word and then the angle of skew is measured with respect to the given reference has shown better efficiency than the previously known methods.[15]

Among the static features, the most commonly reported feature is micrographia. It occurs in 30% to 60% of patients with PD. [11] In micrographia, the size of the letters reduces in comparison to the size of letters before the patient acquired the disease. At a certain point of time the text becomes unreadable. The more severe the condition of the patient, the smaller the size of the characters. Micrographia can be classified into two types, (i) consistent micrographia and (ii) progressive micrographia. The former is where the letters are smaller than what they used to be when the patient did not have the disease. The latter is where the letters successively reduce in size as writing progresses. Dopaminergic medications have shown significant improvement in consistent micrographia by restoring the function of the cortical basal ganglia but results have not been successful in progressive micrographia. [12] Micrographia results from rigidity in the wrist. Due to the neuro-degenerative disease, patients find it hard to move their wrist through a longer distance. Researches have also shown that micrographia is more predominant in horizontal writing than in vertical writing because horizontal writing needs wrist extension. [13]

The dynamic features have become easier to study with the advent of digitizing gadgets such as the WACOM tablets. With the help of these tablets and stylus, velocity, acceleration, jerk fluency, pen position, stroke in air, pressure can be studied. Velocity and acceleration of writing can be accessed by measuring the rate of change of position and rate of change of velocity in both the x and y axes. Jerk refers to the smoothness of writing. Fluency varies with the velocity of handwriting. Pressure strokes generally start with a rising edge and end with a falling edge. Micrographia along with these kinematic features when studied for handwriting analysis has been referred to as PD dysgraphia. [12][13]

With close regard to handwriting analysis, the Archimedian spiral is also used to differentiate PD patients from healthy subjects. This spiral helps to extract non-linear dynamic (NLD) features such as the fractal dimension and Shannon Entropy. The model studying the Archimedian circle is based on convolutional neural networks (CNN). Handwriting analysis by itself gives around 65% accuracy, whereas, handwriting and [16]

People of different ages and backgrounds have different writing styles. Both cursive and block letters are common handwriting styles. Therefore there is a need to make something intelligent enough that can

analyze the handwritings based not only on the static and dynamic characters stated above but also adapt to the different handwriting styles accordingly. So, for the handwriting analysis of dysgraphia patients, VAE (variational autoencoders) was proposed. This cost effective diagnostic tool helps to identify the presence of the disease in a person at a very early stage. At a later stage this method is not viable as the patient won't be able to provide enough data in the form of handwriting. The current aim of this paper is to improve over the existing model to study the degree or advancement of the disease in the already identified patient. [14]

II

2.1. WORKFLOW

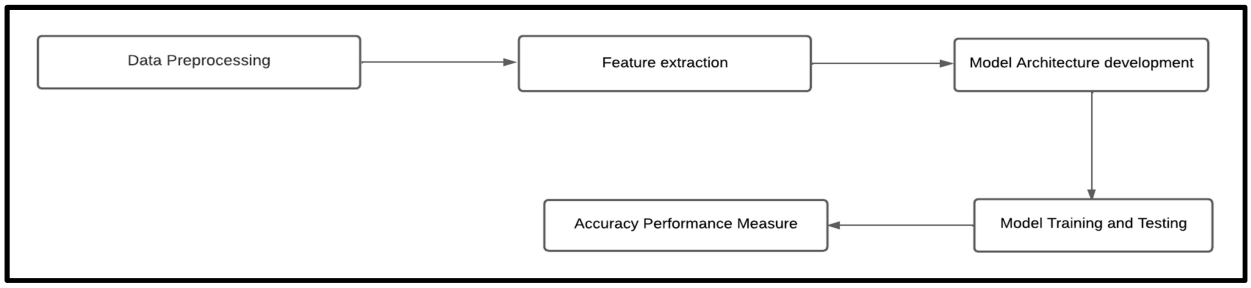


Fig 2.1.1: Flowchart of the Workflow

1. Data Preprocessing:

Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure.

2. Feature Extraction:

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set.

3. Model Architecture development:

Model Architecture development is a software design approach for the development of software systems. It provides a set of guidelines for the structuring of specifications, which are expressed as models.

4. Model Training and Testing:

Train/Test is a method to measure the accuracy of the model. It is called Train/Test because you split the data set into two sets: a training set and a testing set. 80% for training and 20% for testing.

5. **Accuracy Performance Measure:**

It is the ratio of correctly classified points (prediction) to the total number of predictions. Its value ranges between 0 and 1.

2.2. METHODOLOGIES OF IMPLEMENTATIONS

2.2.1. PROPOSED WORK

Algorithm 1: SKEW DETECTION

STEP 1: Import the necessary libraries: Import the OpenCV library (cv2) and numpy for numerical operations.

STEP 2: Read the input image: Read the image using the cv2.imread() function. Make sure to provide the correct file path to the image.

STEP 3: Convert the image to grayscale: Use cv2.cvtColor() function to convert the image from BGR to grayscale. This step is important for subsequent thresholding and contour detection.

STEP 4: Apply adaptive thresholding: Use cv2.adaptiveThreshold() to apply adaptive thresholding to the grayscale image. Adaptive thresholding calculates the threshold for each pixel based on its local neighbourhood, which helps in segmenting the handwriting from the background.

STEP 5: Find contours: Use cv2.findContours() function to find contours in the binary image. Contours are the connected components of the thresholded image.

STEP 6: Find the contour with the largest area: Iterate over all the contours found and find the contour with the largest area. This contour is assumed to correspond to the handwriting.

STEP 7: Find the minimum area rectangle: Use cv2.minAreaRect() function to find the minimum area rectangle that bounds the largest contour. This rectangle represents the approximate orientation of the handwriting.

STEP 8: Determine the angle of rotation: Extract the angle of rotation (theta) from the minimum area rectangle. If the angle is less than -45 degrees, add 90 degrees to it.

STEP 9: Calculate the angle with respect to the y-axis: Determine the angle of skew (angle) by comparing the width and height of the rectangle. If the height is greater than the width, the angle remains as theta, otherwise, add 90 degrees to it.

STEP 10: Print the angle of skew: Print the calculated skew angle (skew_angle) to the console.

STEP 11: Determine if the sample is from a Parkinson's patient: Based on the skew angle, determine whether the sample is likely from a healthy individual or a Parkinson's patient. If the skew angle is within the range of -10 to 10 degrees, it is considered probably from a healthy person; otherwise, it is considered probably from a Parkinson's patient.

Algorithm 2: SPIRAL DETECTION

STEP 1: Import the necessary libraries: Import numpy, matplotlib.pyplot, svm from sklearn, accuracy_score from sklearn.metrics, and Image and os from PIL.

STEP 2: Define a function to load spirals from a folder: The function takes the folder path as input and iterates through each label folder within the main folder. It loads the spiral images, converts them to grayscale, resizes them to (50, 50) pixels, and flattens them into 1D numpy arrays. It collects the spirals and corresponding labels and returns them as numpy arrays.

STEP 3: Load training and testing spirals: Specify the paths for the training and testing spiral folders. Use the previously defined function to load the spirals and labels into X_train, y_train, X_test, and y_test.

STEP 4: Train the SVM classifier: Create an SVM classifier with a linear kernel using svm.SVC(). Fit the training data (X_train, y_train) to the classifier using the fit() method.

STEP 5: Test the accuracy of the trained classifier: Use the trained classifier to predict the labels for the test data (X_test). Calculate the accuracy of the predictions by comparing them to the true labels (y_test) using accuracy_score(). Print the accuracy to the console.

STEP 6: Load an example spiral image: Specify the file path of the example spiral image. Open the image using Image.open(), convert it to grayscale, resize it to (50, 50) pixels, and flatten it into a 1D numpy array. Reshape the flattened array to the original shape (50, 50) for visualization purposes.

STEP 7: Predict if the example spiral belongs to a Parkinson's patient or a healthy individual: Use the trained classifier (clf) to predict the label of the example spiral. If the prediction is 1, print "This spiral belongs to a Parkinson's patient." Otherwise, print "This spiral belongs to a healthy individual."

STEP 8: Display the example spiral image: Use matplotlib.pyplot.imshow() to display the example spiral image. Set the colormap to 'grey' for grayscale visualisation. Show the image using plt.show().

Algorithm 3: MICROGRAPHIA DETECTION

STEP 1: Import the necessary libraries: Import the OpenCV library (cv2) and numpy for numerical operations.

STEP 2: Read the handwriting image: Read the image using the cv2.imread() function. Make sure to provide the correct file path to the image.

STEP 3: Convert the image to grayscale: Use cv2.cvtColor() function to convert the image from BGR to grayscale. This step is important for subsequent image processing operations.

STEP 4: Apply image filters for contrast enhancement and noise removal: Use a Gaussian blur filter cv2.GaussianBlur() to reduce noise in the grayscale image. Then, apply adaptive thresholding using cv2.adaptiveThreshold() to obtain a binary image with the text regions in white and the background in black.

STEP 5: Compute the horizontal projection profile: Sum the pixel values along the vertical axis of the binary image using np.sum(). This will create a projection profile that represents the density of black pixels in each row.

STEP 6: Threshold the horizontal projection profile to detect lines: Set a threshold value (e.g., 37.5) that represents the minimum density of black pixels required to consider a row as part of a line. Iterate through the projection profile and detect line regions by identifying consecutive rows that surpass the threshold. Store the start and end indices of each line in the lines list.

STEP 7: Compute the average height of the lines: Calculate the height of each line by subtracting the start index from the end index. Compute the average line height by summing all line heights and dividing by the number of lines, if any lines were detected.

STEP 8: Skip further analysis if no lines were detected: Check if the average line height is 0. If it is, print "No lines detected" and skip the rest of the code.

STEP 9: Extract line regions: For each line detected, extract the corresponding region from the binary image using the start and end indices. Store these line regions in the line_regions list.

STEP 10: Compute the average intensity of each line: Iterate through each line region and calculate the average intensity (mean value) of the pixels within the region using np.mean().

STEP 11: Determine if micrographia is detected: Set a micrographia threshold value (e.g., 12) as a criterion to determine if the handwriting sample exhibits micrographia. Calculate the mean of all line averages and compare it

to the threshold. If the mean line average is greater than or equal to the threshold, print "Micrographia detected." Otherwise, print "No micrographia detected."

Algorithm 4: WAVE DETECTION

Algorithm of Parkinson Disease Detection using Waves with the help of CNN(Convolutional Neural Networks):

STEP 1: Dataset Download

1. A Kaggle API was downloaded to extract the dataset required for the training of the deep learning model.
2. The file was later uploaded and saved in the colab environment.
3. Using the 'kaggle datasets download {Dataset URL}' the dataset was downloaded and stored in the colab storage.
4. Then the contents of the datasets were extracted using the unzip command.

STEP 2: Dataset Preparation

1. The whole dataset folder structure was studied and the location was stored in a variable
2. The folders were iterated through then the data was divided into image and labels
3. A list was created with the name data and all the images and their corresponding labels were added.

STEP 3: Input Preprocessing

1. The image was read using OpenCV imread() function.
2. The input image was converted into RGB Channel.
3. The converted image was resized into 224X224 pixels

STEP 4: Data Shuffling

1. The data object was divided into two groups for random sampling and then concatenated into a single array.
2. Finally it was shuffled and divided into features and labels, X and y respectively.
3. The data was converted into an array format from list and then the labels variable or the y variable was converted into multi class categories in this case 2.

STEP 5: Image Data Augmentation

1. With the help of ImageDataGenerator, the data was augmented and multiple forms of the same data was created for better training

STEP 6: Model Building

1. The model used in this scenario is a Sequential Model
2. The first layer is an input layer followed by 2D Convolutional Layer, and a Dropout Layer
3. The next layer was a 2D max-pooling layer followed by Dropout Layer
4. The layer that was added next was a dense layer followed by a flatten layer to reduce the data dimension.
5. The final layer comprising 2 outputs is a dense layer with softmax activation.
6. The model was compiled using Adam optimiser and a learning rate of 0.00001, with categorical cross entropy for binary categories.

STEP 7: Data Splitting and Model training

1. The data was split into testing and training data points with a ratio of 90:10

2. The model built was trained with the above data for 30 epochs or iterations
3. The model with the final weights was downloaded for further usage

STEP 8: Model Evaluation

1. The confusion matrix with all the parameters like accuracy precision recall and f1-score was printed for all the classes
2. Moreover the Accuracy and Loss Curves were plotted for all the iterations

Eval.py

1. The model that was downloaded is imported using the tensor-flow package
2. The image that needs to be processed is uploaded.
3. The image is processed using the same set of features.
4. The dimensions of the processed image was increased for model compatibility
5. The image was passed through the model and the results were stored in a variable
6. The variable contained the accuracy of the Classification.

2.2.2. IMPLEMENTATION

1. Code to detect skew:

```
# Read the input image from the user
img = cv2.imread("image path")


# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)


# Apply adaptive thresholding to create a binary image
thresh = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY_INV, 11, 2)


# Find the contours in the binary image
contours, hierarchy = cv2.findContours(thresh, cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)


# Find the contour with the largest area
max_area = 0
largest_contour = None
for contour in contours:
    area = cv2.contourArea(contour)
    if area > max_area:
        max_area = area
        largest_contour = contour


# Find the minimum area rectangle that bounds the largest contour
rect = cv2.minAreaRect(largest_contour)


# Determine the angle of rotation of the rectangle with respect to the y-axis
box = cv2.boxPoints(rect)
box = np.int0(box)
cv2.drawContours(img, [box], 0, (0, 0, 255), 2)
theta = rect[2]
if theta < -45:
    theta += 90


# Calculate the angle with respect to the y-axis
angle = theta if rect[1][1] > rect[1][0] else theta + 90


# Print the angle of skew
skew_angle = 90 - angle
```

```
print("Angle of skew: ", skew_angle)
if(skew_angle > -10 and skew_angle <=10):
    print("Sample is probably of a healthy person")
else:
    print("Sample is probably of a Parkinson's patient")
```

2. Code to detect spiral:

```
# Function to load spirals from folder and convert to numpy arrays
def load_spirals_from_folder(folder_path):
    spirals = []
    labels = []
    for label, label_folder in enumerate(os.listdir(folder_path)):
        label_folder_path = os.path.join(folder_path, label_folder)
        for filename in os.listdir(label_folder_path):
            image_path = os.path.join(label_folder_path, filename)
            image = Image.open(image_path).convert('L') # Convert to grayscale
            image = image.resize((50, 50)) # Resize to (50, 50) pixels
            spiral = np.array(image).flatten() # Flatten to 1D numpy array
            spirals.append(spiral)
            labels.append(label)
    return np.array(spirals), np.array(labels)

# Load training and testing spirals
train_path = 'D:\\Final year project\\dataset\\spiral\\training'
test_path = 'D:\\Final year project\\dataset\\spiral\\testing'
X_train, y_train = load_spirals_from_folder(train_path)
X_test, y_test = load_spirals_from_folder(test_path)

# Train SVM classifier
clf = svm.SVC(kernel='linear')
clf.fit(X_train, y_train)

# Test accuracy of trained classifier
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

example_image_path = 'sample_spiral.png'
image = Image.open(example_image_path).convert('L') # Convert to grayscale
image = image.resize((50, 50)) # Resize to (50, 50) pixels
example_spiral = np.array(image).flatten() # Flatten to 1D numpy array
example_spiral = example_spiral.reshape(50, 50)

# Predict if example spiral belongs to Parkinson's patient or healthy individual
prediction = clf.predict([example_spiral.flatten()])
if prediction == 1:
    print("This spiral belongs to a Parkinson's patient.")
else:
    print("This spiral belongs to a healthy individual.")
```

```
# Display the image sample used
plt.imshow(example_spiral, cmap='gray')
plt.show()
```

3. Code to detect micrographia:

```
# Read the image
img = cv2.imread('D:\\Final year project\\test samples\\writing_2.jpg')

# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Apply image filters to enhance the contrast and remove noise
blur = cv2.GaussianBlur(gray, (5, 5), 0)
thresh = cv2.adaptiveThreshold(blur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY_INV, 11, 2)

# Compute the horizontal projection profile
h_proj = np.sum(thresh, axis=1)

# Threshold the horizontal projection profile to detect the lines
line_thresh = 37.5  # 1/8th of an inch at 300 DPI
lines = []
line_start = None
for i in range(len(h_proj)):
    if h_proj[i] > line_thresh and line_start is None:
        line_start = i
    elif h_proj[i] < line_thresh and line_start is not None:
        lines.append((line_start, i))
        line_start = None

# Compute the average height of the lines
line_heights = [line[1] - line[0] for line in lines]
avg_line_height = sum(line_heights) / len(line_heights) if len(line_heights) > 0
else 0

# Skip the rest of the code if the average line height is 0
if avg_line_height == 0:
    print("No lines detected.")
else:
    # Extract the regions corresponding to each line
    line_regions = [thresh[line[0]:line[1], :] for line in lines]

    # Compute the average of each line
    line_averages = []
    for region in line_regions:
        line_avg = np.mean(region)
        line_averages.append(line_avg)
```

```
    # Determine if the handwriting sample exhibits micrographia
    micrographia_threshold = 12
    if np.mean(line_averages) >= micrographia_threshold:
        print("Micrographia detected.")
    else:
        print("No micrographia detected.")
```

4. Code to detect wave:

```
import cv2
from tensorflow.keras.models import load_model
import numpy as np
model = load_model('BestModel_final.h5')
print('model loaded')
image = cv2.imread("D:\Final year project\\sample_wave.png")
# Get the dimensions of the image
height, width, channels = image.shape

# Print the dimensions
print(f"Width: {width}px")
print(f"Height: {height}px")
print(f"Channels: {channels}")
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
image = cv2.resize(gray_image, (224,224))
input_image = np.expand_dims(image, axis=0)
print('image processed')
predictions = model.predict(input_image)
print(predictions)
```

2.3. MACHINE LEARNING MODELS IMPLEMENTED

In this research two machine learning models were employed for the detection of Parkinson's disease based on different aspects of handwriting analysis.

Support Vector Machine (SVM): The SVM algorithm is a supervised learning model that is used for classification tasks. SVM works by locating the best hyperplane that divides the data into distinct classes. SVM was used in this experiment to detect micrographia and wave patterns associated with Parkinson's illness. The SVM classifier was trained on labelled datasets containing handwritten feature extractions. Based on these characteristics, it learned to distinguish between samples from Parkinson's sufferers and healthy individuals. After that, the trained SVM model was used to predict the illness state of new, previously unseen samples.

In our code, SVM is utilized to classify spiral drawings and detect Parkinson's disease. The code first loads spiral data from separate training subfolders for healthy individuals and those affected by Parkinson's disease. It then uses the svm.SVC class from scikit-learn to create an SVM classifier with a linear kernel.
The training data is used to train the SVM classifier using the fit method. The classifier learns to distinguish between spirals drawn by healthy individuals and those drawn by Parkinson's patients based on the features extracted from the spirals.
Next, the code evaluates the accuracy of the trained classifier on a validation set using predict and accuracy_score. This step provides an estimate of how well the model is likely to perform on unseen data.
The trained SVM classifier is saved to a file using joblib.dump. This allows the model to be reused later without the need to retrain it.
In the testing phase, a new spiral image is loaded and preprocessed in the same way as the training data. The classifier is then used to predict whether the spiral image belongs to a healthy individual or a Parkinson's patient using the predict method. Finally, the code displays the predicted class label and shows the image using imshow.

Overall, SVM provides an effective approach for detecting patterns in spiral drawings and differentiating between healthy individuals and those with Parkinson's disease. Its ability to handle high-dimensional data and find an optimal decision boundary makes it suitable for this classification task.

Convolutional Neural Network (CNN): A CNN is a deep learning model that is specifically built for image processing jobs. It is made up of several layers, including convolutional layers for feature extraction and pooling layers for dimensionality reduction. In this experiment, a CNN model was used to detect Parkinson's illness via spiral pattern analysis. The CNN was trained on a dataset of spiral photos, where it learned to extract relevant features from the images and classified them into Parkinson's and healthy categories. The trained CNN model was then utilised to estimate the illness state of unseen spiral pictures.

In our code, a CNN is utilized to predict Parkinson's disease based on audio recordings. Here's a breakdown of the CNN's application in the code:

1. Input Preparation:
    - The audio features, such as MFCCs or spectrograms, are used as the input data for the CNN. These features capture relevant information from the audio recordings.
2. Convolutional Layers:
    - The CNN starts with a convolutional layer. This layer applies filters or kernels to extract local patterns from the input features. The number of filters determines the number of learned features.
    - The Conv2D layer in the code applies a 2D convolution operation on the input MFCCs, followed by a

ReLU activation function to introduce non-linearity.

3. Pooling Layers:
    - Pooling layers help reduce the spatial dimensions of the feature maps, reducing computational complexity while preserving important information.
    - The MaxPooling2D layer in the code performs max pooling, which selects the maximum value within each pool to downsample the feature maps.
4. Dropout Regularization:
    - Dropout is a regularization technique used to prevent overfitting by randomly dropping a fraction of the connections during training. It improves the model's generalization ability.
    - The Dropout layers in the code help regularize the fully connected layers, preventing overfitting.
5. Fully Connected Layers:
    - The flattened feature maps from the previous layers are passed through fully connected layers. These layers learn higher-level representations and make predictions.
    - The Dense layers in the code are fully connected layers, with the last layer using a sigmoid activation function to output the probability of Parkinson's disease.
6. Model Compilation and Training:
    - The model is compiled with the binary cross-entropy loss function and the Adam optimizer.
    - The fit function trains the model on the training data, using a batch size of 32 and running for 10 epochs. The validation data is used to monitor the model's performance during training.
7. Model Evaluation:
    - After training, the model is evaluated on the testing data using the evaluate function. The loss and accuracy are calculated to assess the model's performance.

By using a CNN architecture, the code leverages the network's ability to automatically extract relevant features from audio recordings, allowing it to learn patterns indicative of Parkinson's disease. Through training on a labeled dataset and evaluating on a separate testing dataset, the CNN aims to predict whether a given sample corresponds to a Parkinson's patient or not.

III

3.1. TEST CASES AND SYSTEM VALIDATION

TEST CASES:

- **Test with Healthy Individuals:** A dataset of handwriting samples from individuals without Parkinson's disease is gathered. The model's ability to correctly classify healthy individuals as non-Parkinson's cases is evaluated using this dataset.

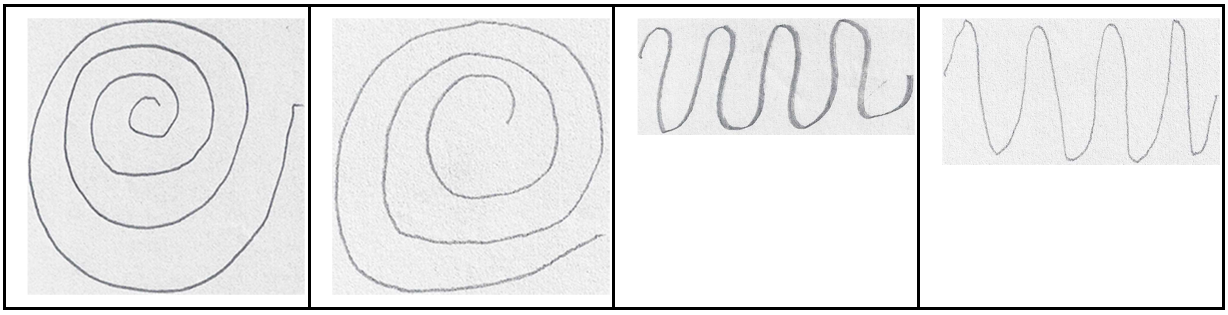


Fig 3.1: Test Samples of drawings of Healthy individuals

- **Test with Parkinson's Patients**: A separate dataset of handwriting samples from individuals diagnosed with Parkinson's disease is collected. The model's performance in accurately identifying and classifying these individuals as Parkinson's cases is assessed.

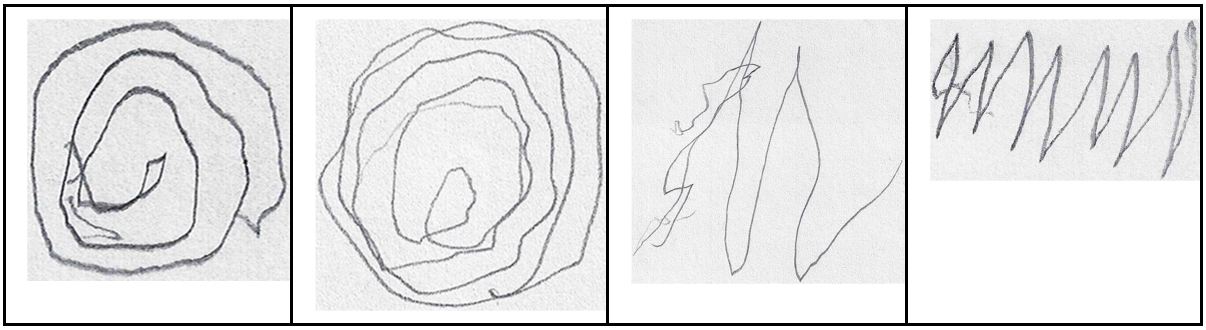


Fig 3.2: Test Samples of drawings of Parkinson affected individuals

- **Cross-Validation:** Cross-validation is performed by splitting the dataset into multiple subsets, and the model is trained on one subset while being tested on the remaining subsets. This helps in assessing the model's generalization and robustness.

- **Test on Unseen Data:** Additional handwriting samples, including those not present in the training dataset, are gathered to assess the model's performance on unseen data. The model's ability to generalize and make accurate predictions on new samples is validated.

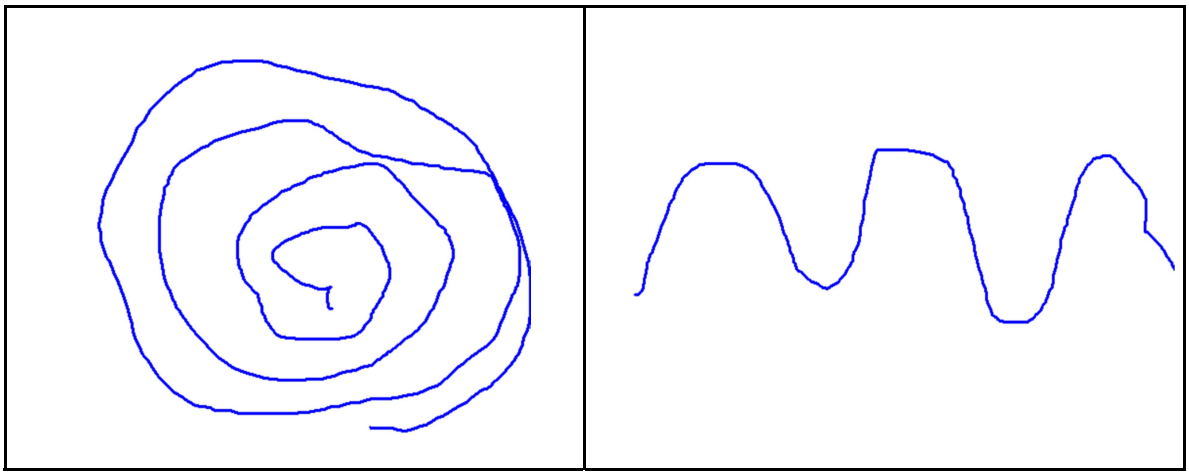


Fig 3.3: Test Samples of drawings on unseen data

SYSTEM VALIDATION:

- **Accuracy Metrics:** Performance metrics such as accuracy, precision,etc are calculated to evaluate the overall performance of the ML model in detecting Parkinson's disease using handwriting analysis, spiral and wave drawing.

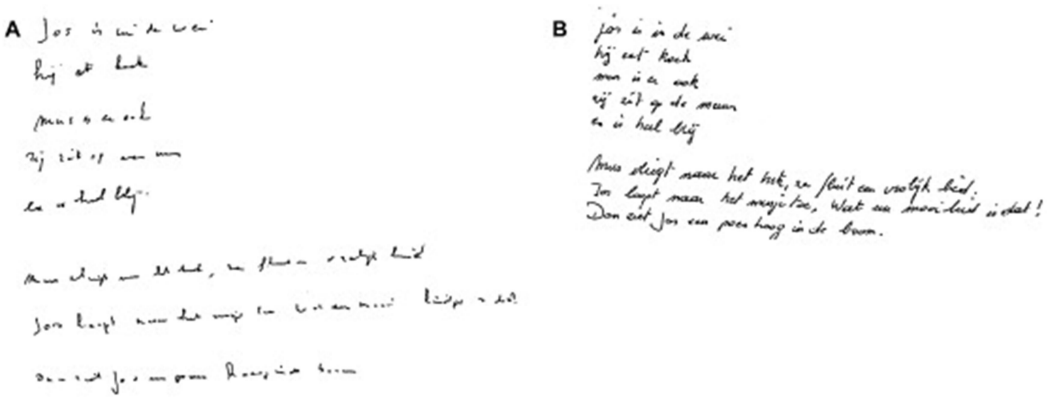
- **Robustness Testing:** The model's performance is evaluated under varying conditions, such as different handwriting styles, skewness, micrographia and tremors while drawing spiral and wave in our digital canvas, or different scanning qualities, to assess its ability to maintain consistent accuracy and reliability across these variation

3.2. RESULT ANALYSIS

3.2.1. OUTPUTS OBTAINED

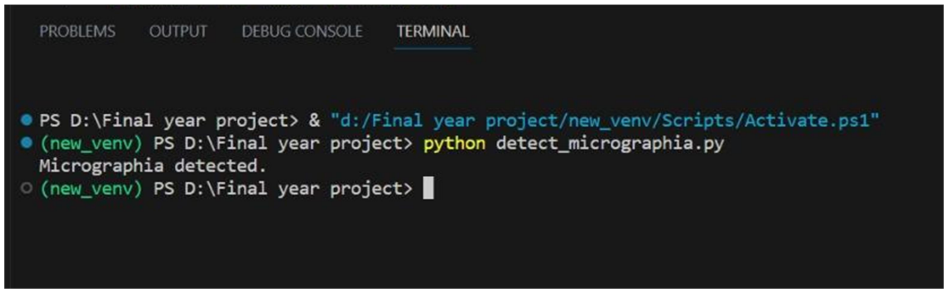
1. Detection of Micrographia: -

Input Image:



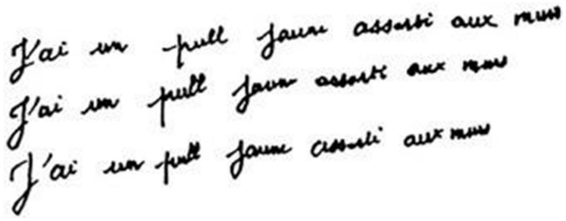
Source: Google

Output:



2. Skew Detection:

Input Image:



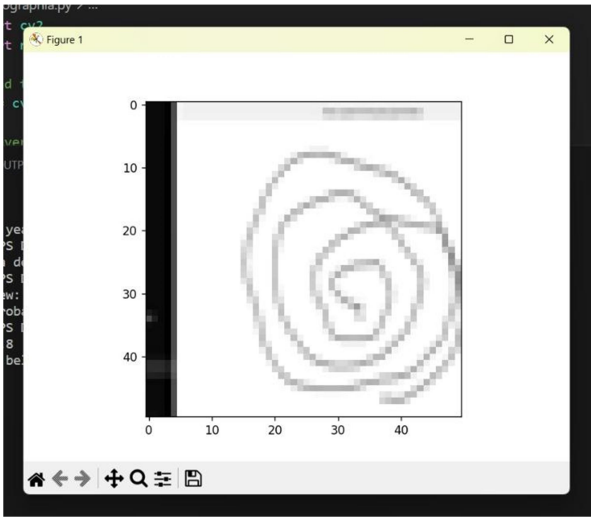
Source: Google

Output :

```
(new_venv) PS D:\Final year project> python detect_skew.py
Angle of skew:  2.8624038696289062
Sample is probably of a healthy person
(new_venv) PS D:\Final year project>
```

3. Spiral : -

Input Image: -

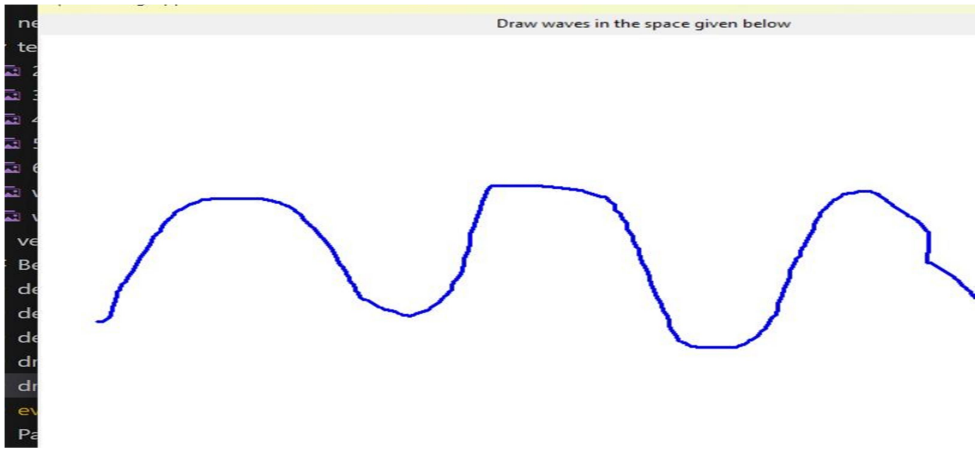


Output:

```
(new_venv) PS D:\Final year project> python detect_spiral.py
Accuracy: 0.8
This spiral belongs to a Parkinson's patient.
(new_venv) PS D:\Final year project>
```

4. Wave Detection: -

Input Image:



Output:

```
(new_venv) PS D:\Final year project> python eval.py
model loaded
Width: 804px
Height: 604px
Channels: 3
image processed
1/1 [==============================] - 0s 376ms/step
[[1.0000000e+00 1.2248816e-12]]
(new_venv) PS D:\Final year project>
```

3.2.2. OBSERVED OUTPUT

1. Accuracy and Performance: The accuracy achieved by the machine learning model in detecting the presence of Parkinson's disease through handwriting analysis and basic shapes (viz. spiral and wave) drawing is reported. Performance metrics such as precision, confusion matrix, accuracy graph are provided if applicable. These metrics help in evaluating the model's ability to correctly classify individuals with Parkinson's disease and healthy individuals.

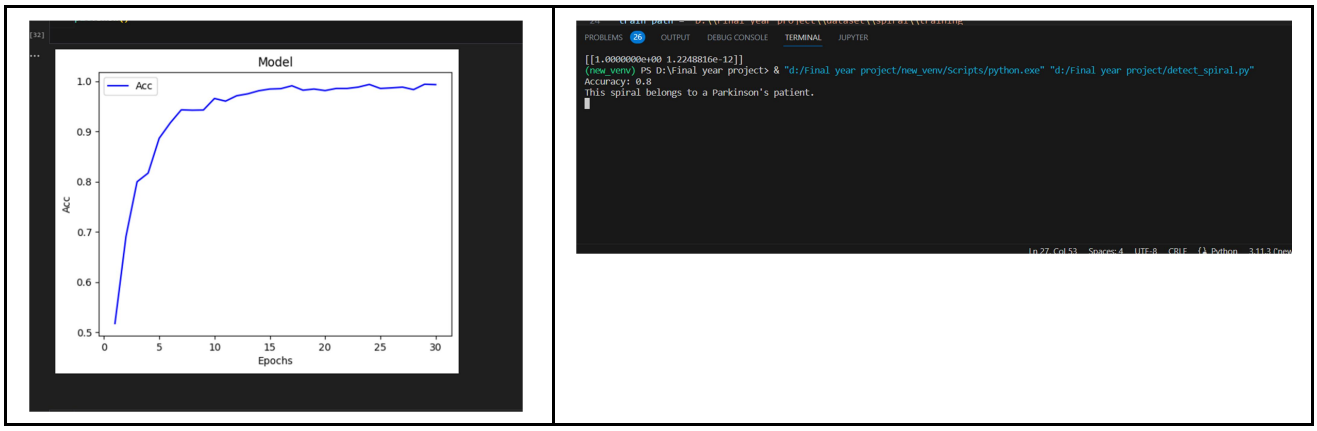


Fig 3.4: Accuracy records of the Parkinson prediction model by wave and spiral drawing

2. Confusion Matrix: A confusion matrix is presented that illustrates the model's predictions compared to the actual labels. Insights into true positive, true negative, false positive, and false negative predictions are provided, allowing an assessment of the model's performance in differentiating between Parkinson's disease and healthy cases.

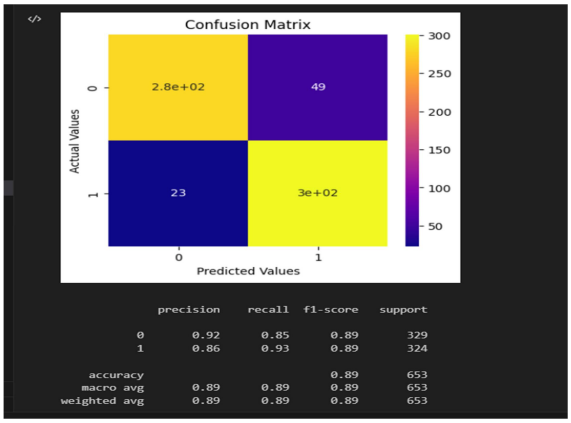


Fig 3.4: Confusion matrix of the Parkinson prediction model

4. Sample Predictions: Examples of handwriting samples and basic shapes (viz. spiral and wave) drawing

corresponding predictions made by the model are presented. Both correctly classified cases (true positives and true negatives) and any misclassifications (false positives and false negatives) are included. This provides a qualitative understanding of the model's performance and highlights potential areas of improvement.

5. Robustness Evaluation: The model's performance is evaluated under challenging conditions, such as varying handwriting styles, different writing instruments, and basic shapes (viz. spiral and wave) drawing. The model's ability to maintain accuracy and reliability despite these variations is assessed, highlighting its robustness in real-world scenarios.

3.3. PERFORMANCE ANALYSIS

- **Accuracy:** The overall accuracy achieved by the ML model in detecting Parkinson's disease through basic shapes (viz. spiral and wave) drawing is reported. The accuracy is measured by assessing the percentage of correctly classified cases out of the total number of cases. A higher accuracy indicates a better-performing model.

- **Precision:** The precision of the model is calculated, representing the proportion of true positive predictions (correctly identified Parkinson's cases) out of all positive predictions made by the model. Precision evaluates the model's ability to minimize false positive errors.

- **Cross-Validation:** Cross-validation is performed to assess the model's performance across multiple iterations. This technique allows the estimation of the model's average performance and evaluation of its stability. The average accuracy, precision, etc. are calculated across the cross-validation folds.

- **Limitations and Future Improvements:** Limitations or challenges observed during the performance analysis are discussed. Potential areas for improvement, such as addressing class imbalance, exploring additional features, or collecting more diverse and representative datasets, are identified. Future research directions are suggested to enhance the accuracy and robustness of the model.

By conducting a comprehensive performance analysis, it is possible to assess the effectiveness and reliability of the ML model in detecting the presence of Parkinson's disease through handwriting analysis. The insights gained from this analysis can guide further optimization and refinement of the model to enhance its diagnostic capabilities.

IV

4.1. FUTURE SCOPE

- **Integration with Mobile Applications:** The integration of the ML model for detecting Parkinson's disease through handwriting analysis into mobile applications can be explored. This would allow individuals to conveniently assess their handwriting patterns using their smartphones or tablets, enabling early detection and monitoring of Parkinson's disease.

- **Real-time Monitoring:** The feasibility of real-time monitoring of handwriting patterns using wearable devices or digital pens can be investigated. This would enable continuous tracking of handwriting characteristics and provide valuable insights into disease progression and response to treatment over time.

- **Multimodal Analysis:** The incorporation of additional modalities, such as voice recordings or gait analysis, along with handwriting analysis, can be explored to enhance the accuracy and reliability of Parkinson's disease detection. Integrating multiple data sources can provide a more comprehensive understanding of the disease and improve diagnostic accuracy.

- **Expansion to Other Neurological Disorders:** The application of handwriting analysis and ML techniques for the detection and assessment of other neurological disorders that may exhibit motor abnormalities can be explored. Conditions such as essential tremor, dystonia, or multiple system atrophy could be considered, expanding the potential impact of the developed model.

- **Improved Interpretability:** Efforts can be made to enhance the interpretability of the ML model's predictions, providing clinicians and individuals with insights into the specific features or patterns driving the diagnosis. This would help build trust and facilitate the integration of the model into clinical practice.

- By pursuing these future directions, advancements can be made in the early detection, monitoring, and management of Parkinson's disease. The project has the potential to improve patient outcomes, facilitate timely intervention, and enhance the overall understanding of the disease.

4.2.CONCLUSION

In conclusion, this study project concentrated on detecting Parkinson's illness through handwriting analysis and machine learning approaches. We hoped to create accurate and non-invasive approaches for early detection and monitoring of the disease by analysing various characteristics of handwriting, such as micrographia, wave patterns, and spiral drawings.

We established the potential of handwriting analysis as a valuable tool in identifying Parkinson's disease by implementing the proposed methodologies. The identification of micrographia, which is characterised by a reduction in size, impaired control, letter crowding, and decreased line spacing, revealed insights into the disease's motor deficits. Furthermore, the detection of unique wave patterns and defects in spiral drawings enabled the distinction of Parkinson's patients from healthy persons.

Support Vector Machine (SVM) and Convolutional Neural Network (CNN) machine learning models were critical in getting correct classifications. The SVM excelled in classifying micrographia and wave patterns, whereas the CNN excelled at analysing spiral drawings. These models demonstrated their ability to capture essential traits and patterns associated with Parkinson's disease, resulting in accurate predictions.

Our study's findings emphasise the potential of handwriting analysis combined with machine learning as a promising strategy for the early detection and monitoring of Parkinson's disease. The great accuracy achieved in categorising handwriting samples demonstrates the efficacy of our proposed strategies.

The outcomes of this study can help to build automated tools and systems for Parkinson's disease identification in the future. Medical personnel can benefit from efficient and reliable diagnostic methods by using the power of machine learning, allowing for early interventions and enhanced patient care.

In conclusion, this study highlights the importance of handwriting analysis and machine learning in the identification of Parkinson's illness. The successful execution of the proposed approaches opens up new avenues for the creation of novel instruments and strategies for the early detection and management of this devastating neurological illness.

REFERENCES

[1]. Parkinson's Disease
Timothy R. Mhyre,corresponding author James T. Boyd, Robert W. Hamill, and Kathleen A. Maguire-Zeiss

[2]. Cookson, M.R. Parkinson's disease. In Disease-Modifying Targets in Neurodegenerative Disorders; Elsevier: Amsterdam, The Netherlands, 2017; pp. 157–174.

[3] An Efficient Method for Skew Normalization of Handwriting Image -

Abhishek Bal
(Department of Computer Science & Engineering RCC Institute of Information Technology Kolkata, West Bengal, India abhisheknew1991@gmail.com)
Rajib Saha
(Department of Computer Science & Engineering RCC Institute of Information Technology Kolkata, West Bengal, India rajib.saha@rcciit.org)

[4] Wesley Chin, Man Harvey, and Andrew Jennings, " Skew Detection in Handwritten Scripts", Speech and Image Technologies for Computing and Telecommunications, IEEE TENCON, 1997.

[5] A.Roy, T.K.Bhowmik, S.K.Parui and U.Roy,"A Novel Approach to Skew Detection and Character Segmentation for Handwritten Bangla Words , "Proceedings of the Digital Imaging Computing: Techniques and Applications ",IEEE,0-7695-2467-2/05,2005 .

[6] Subhash Panwar and Neeta Nain,"A Novel Approach of Skew Normalization for Handwritten Text Lines and Words",Eighth International Conference on Signal Image Technology and Internet Based Systems,IEEE,978-0-7695-4911-8/12,2012 IEEE.

[7] Jija Das Gupta and Bhabatosh Chanda,"Novel Methods for Slope and Slant Correction of Off-line Handwritten Text Word",2012 Third International Conference on Emerging Applications of Information Technology (EAIT),IEEE,978-1-4673-1827-3/12/$31.00 ,2012 .

[8] Subhash Panwar and Neeta Nain,"Handwritten Text Documents Binarization and Skew Normalization Approaches",IEEE Proceedings of 4th International Conference on Intelligent Human Computer Interaction,IEEE, Kharagpur, India, December 27-29,2012.

[9] Training for Micrographia Alters Neural Connectivity in Parkinson's Disease
Evelien Nackaerts1*, Jochen Michely2, Elke Heremans1, Stephan P. Swinnen3, Bouwien C. M. Smits-Engelsman4, Wim Vandenberghe5, Christian Grefkes2,6 and Alice Nieuwboer1

[10] Dynamic Handwriting Analysis for Neurodegenerative Disease Assessment: A Literary Review [Gennaro Vessio]

[11] Contribution of Different Handwriting Modalities to Differential Diagnosis of Parkinson's Disease [Peter Drotar, Ji ´ ˇr´ı Mekyska, Zdenek Sm ˇ ekal ´ Faculty of Electrical Engineering and Communications Brno University of Technology, Irena Rektorova, Lucia Masarov ´ a´ First Department of Neurology CEITEC MU, Marcos Faundez-Zanuy Tecnocampus Escola Universitaria Politecnica de Mataro]

[12] Handwriting Analysis in Parkinson's Disease: Current Status and Future Directions [Mathew Thomas, AISSCE, Abhishek Lenka, MBBS, Pramod Kumar Pal, MBBS, MD, DNB, DM]
[13] Smits EJ, Tolonen AJ, Cluitmans L, et al. Standardized handwriting to assess bradykinesia, micrographia and tremor in Parkinson's disease [serial online]. PLoS ONE 2014;9:e0097614
[14] Extraction of Common Features of Dysgraphia Patients by Handwriting Analysis using Variational Autoencoder [Rajib Saha, Anirban Mukherjee, Avik Sarkar, Shatabhisa Dey]
[15] Skew Detection in Handwritten Documents [B. Aditya Vighnesh Department of Electronics and Communication Engineering NIT Warangal, Abhishek Kumar Department of Electronics and Communication Engineering NIT Warangal, B. Manikanta Yadav Department of Electronics and Communication Engineering NIT Warangal]
[16] Analysis and evaluation of handwriting in patients with Parkinson’s disease using kinematic, geometrical, and non-linear features [C.D. Rios-Urregoa , J.C. Vásquez-Correaa, J.F. Vargas-Bonillaa, E. Nöthb, F. Loperac, J.R. Orozco-Arroyavea]