Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P); 2583-7907 (E)
Vol.13 No.2, 2024, pp.1-13

www.crijournals.org
DOI: https://doi.org/10.70112/ajcst-2024.13.2.4270

Parkinson's Disease Prediction Using Convolutional Neural Networks and Hand-Drawn Image Analysis

Uche-Jerry Nzenwata, Ayodeji G. Abiodun*, Adelola Olayinka, Oluwabamise J. Adeniyi and Akwaronwu B. Gazie
Department of Computer Science, Babcock University, Nigeria
E-mail: nzenwatau@babcock.edu.ng, olayinka0710@pg.babcock.edu.ng, adeniyi0416@pg.babcock.edu.ng, akwaronwu0329@pg.babcock.edu.ng
*Corresponding Author: abiodun0208@pg.babcock.edu.ng

(Received 1 August 2024; Revised 20 August 2024, Accepted 23 September 2024; Available online 30 September 2024)

Abstract **- Parkinson's Disease (PD) is a serious neurodegenerative disorder, with over 10 million cases globally in 2020, significantly affecting patients' quality of life. The progression of the disease has been statistically proven, underscoring the importance of early diagnosis. As many as 80% of those diagnosed with PD begin to experience spinal degeneration, leading to other impairments and disabilities within approximately 10 years. Moreover, up to 35% of patients require assistance to walk or perform daily living activities within 5 years of diagnosis. The proposed study employs a neural convolutional network (CNN) to predict PD using 64x64 pixel hand-drawn images from 244 PD patients and 228 healthy individuals. K-nearest neighbors (KNN)-based feature extraction was applied as a data pre-processing method before feeding the data into the CNN layers. Model training involved tuning hyper parameters and testing several learning rates, ranging from 0.1 to 0.00001. The highest learning rate of 0.001 yielded the best performance, achieving classification accuracies, precision, sensitivity, and F1 score of 97.93%, 92%, 80%, and 86%, respectively, with a 5% increase in performance accuracy. These results demonstrate the model's effective ability to discriminate between healthy individuals and PD patients based on hand-drawn samples.**

Keywords: **Parkinson's Disease (PD), Neurodegenerative Disorder, Neural Convolutional Network (CNN), Feature Extraction, Classification Accuracy**

I. INTRODUCTION

Parkinson's Disease (PD) is defined as a progressive disorder of the central nervous system, affecting approximately 10 million people worldwide as of 2020. It is characterized by impairments in motor functions, which include tremors, muscle rigidity, and difficulties with movement [1]. Parkinson's Disease leads to decreased motor function, resulting in tremors, stiffness, and general movement impairment. As a midbrain-affecting neurological condition, the thalamic area in the midbrain houses the substantia nigra, which contains a large number of dopamine neurons. The brain produces a chemical called dopamine, which is essential for neuronal communication. However, Parkinson's Disease occurs when there is insufficient dopamine flowing through the body. Symptoms include hand tremors, stiffness in the arms, legs, and jaw, impaired balance and coordination, speech difficulties, and other motor challenges [2]. Early detection of the disease is critical for managing symptoms and improving the quality of life for those affected [3].

Ranked next to Alzheimer's disease, Parkinson's Disease is the second leading cause of degenerative diseases, with nearly 6.3 million people suffering from it globally [4]. It is caused by the loss of dopamine-producing neurons and typically affects individuals over the age of 60. Although there is no treatment for Parkinson's Disease, early detection may mitigate the illness. Researchers have focused on identifying and monitoring Parkinson's Disease through gait analysis [5] and speech analysis data [6]. Notably, about 90% of Parkinson's Disease patients exhibit motor deficits [7].

In the current era of artificial intelligence (AI), many diseases are modeled using machine learning (ML) to better understand disease behavior (including infectious diseases) and their transmission from hosts to targets. This presents the potential to reduce the impact of disorders, such as Parkinson's Disease, for which no known treatment exists. The analytical and predictive capabilities of machine learning algorithms facilitate this process [8]. Proponents of AI describe it as the scientific endeavor to create physical devices that replicate human intellect. Conversely, machine learning is a mathematically constructed technique that enables AI technologies to learn intelligently using well-refined datasets. The study in [9] identified that the capability of machine learning and AI to automate accurate pattern detection has contributed to their increasing popularity.

Given the prevalence of Parkinson's Disease, early diagnosis is crucial to provide patients with the appropriate treatment and prognostic information. However, since the movement symptoms may resemble those of other illnesses, obtaining an accurate early diagnosis can be challenging. Parkinson's Disease is primarily diagnosed through clinical assessments and patient examination data. In some cases, brain imaging may support the diagnosis; however, no diagnostic tests are entirely accurate or specific for Parkinson's Disease. According to a study conducted in [10], approximately 10-25% of Parkinson's Disease cases are misdiagnosed, and it takes an average of 2.9 years to achieve 90% diagnostic accuracy. The most reliable method for determining the disease's identity remains autopsy.

In summary, diagnosing Parkinson's Disease poses significant challenges. The detection of Parkinson's Disease can be difficult, as no medical test can identify the disease until notable physical symptoms manifest, which are often detectable only in rare cases, except for tremors observed during writing. This tremor is considered the most common symptom noticeable in nearly every patient with Parkinson's Disease. The application of advanced AI technologies presents an opportunity to develop automated diagnostic systems that can aid in detecting Parkinson's Disease in its early stages, providing critical support for early intervention. To identify Parkinson's Disease and address associated diagnostic challenges, deep learning (DL) algorithms based on various diagnostic techniques can be implemented. Therefore, this study considers the use of a handwriting image dataset to train a deep learning algorithm that predicts whether a patient has Parkinson's Disease.

A. Statement of the Problem

Parkinson's disease is the second most common neurodegenerative disease after Alzheimer's, characterized by the degeneration of the central nervous system, which controls movements, posture, and balance. Approximately 10 million people are diagnosed with Parkinson's globally. Despite numerous attempts, no medical treatment has been found to be beneficial in either controlling or halting the disease. Various diagnostic methodologies, including neurological assessments, Support Vector Classifiers, Random Forests, Artificial Neural Networks, and K-Nearest Neighbors, have been employed for the detection of the disease. However, the accuracy reported for Parkinson's disease detection using these methods has been very low due to challenges in analyzing image datasets. To overcome this limitation, the current paper proposes hand-drawn image-based prediction of Parkinson's disease using convolutional neural networks (CNNs) and examines the effects of varied learning rates on this process.

B. Objective of the Study

The main objective of the study is to develop a system for predicting Parkinson's disease using a Convolutional Neural Network (CNN) model with different learning rates. The specific objectives are to:

1. Obtain and pre-process the image dataset into a machine learning (ML) acceptable format.
2. Design the deep learning model for predicting Parkinson's disease.
3. Implement the model.
4. Evaluate the implemented model.

C. Research Questions

1. How do accuracy, precision, and recall metrics vary with different learning rates in Convolutional Neural Network (CNN) models for predicting Parkinson's disease?
2. What is the overall performance difference between CNN models trained on spiral image datasets versus those trained on wavy image datasets?
3. What are the comparative performances of CNN models implemented with different learning rates in predicting Parkinson's disease?

II. REVIEW OF LITERATURE

Driven by the critical need for early diagnosis, machine learning is revolutionizing Parkinson's disease research. Leveraging diverse data sources, researchers are developing increasingly sophisticated models for disease identification and prediction.

The study by Y. Zhang [11] utilized support vector machines and random forests as machine learning approaches in the analysis of gait data for predicting the likelihood of suffering from Parkinson's disease. The results were promising, indicating the potential of these models for early recognition of the disease. Likewise, S. M. A. Asaduzzaman Sakib *et al.,* [12] investigated the use of deep learning to predict the onset of Parkinson's disease from neuroimaging data. This work emphasized the superiority of convolutional neural networks (CNNs) in MRI studies, thus providing a solution for imaging the disease and its follow-up. Additionally, the research conducted in [13] incorporated data gathered from wearable devices into machine learning to create a predictive model. Their model demonstrated a high degree of precision in identifying motor symptoms, paving the way for possible early intervention approaches.

The study by M. B. Makarious *et al.,* [14] aimed to enhance the risk assessment model for developing Parkinson's disease by integrating genetic, clinical, and epidemiological data with machine learning algorithms. It was found that using various types of information produced from in vivo and clinical data sources improves accuracy, particularly for disease diagnosis, thereby underscoring the importance of a multimodal perspective. Overall, these studies indicate that machine learning models have the potential to predict whether an individual has Parkinson's disease through gait analysis, neuroimaging, sensors, and genetic data. While the results are encouraging, further study is needed to enhance the models' robustness, generalizability, and application in clinical settings.

In predicting the accuracy of detection models, the study in [15] revealed that the nature of the dataset significantly contributes to enhancing prediction model accuracy. Feature selection and classification techniques were explored in [16], focusing on methods such as support vector machines (SVM) and artificial neural networks (ANN) to classify Parkinson's disease from other conditions with a high level of accuracy. Similarly, [17] employed wearable sensors and deep learning models, comprising CNNs and recurrent neural networks (RNNs), to detect patterns and variations accompanying the development of Parkinson's disease.

Studies conducted by P. Arora, L. Ali, and S. Chakraborty [18]-[20] also utilized similar datasets to detect Parkinson's disease from a given handwritten image dataset. Another

model was constructed using long short-term memory (LSTM) and multi-layer perceptron (MLP) algorithms, which successfully detected and classified 73% of the image data, achieving recall, precision, and F scores of 88.2%, 84.5%, and 85.0%, respectively. Additionally, [19] reported increased accuracy when training a cascaded learning system by combining a Chi-squared model with an adaptive boosting (AdaBoost) method, achieving classification accuracy of 76.44%, sensitivity of 70.94%, and specificity of 81.94%. Lastly, the work in [20] analyzed spiral and wave drawing patterns using two differently structured CNN models on data from 55 patients tilting at cervical bend angles, resulting in an overall accuracy of 83.3%, with average recall, precision, and F values of 84%, 83.5%, and 83.94%, respectively.

From this review, it is evident that using the CNN algorithm to discriminate between healthy and diseased patients from handwriting datasets is a viable method for predicting Parkinson's disease, as it can identify subtle patterns. This current study, therefore, re-examines the CNN model on a similar dataset to investigate whether hyperparameter tuning can improve the accuracy of existing models.

III. METHODOLOGY

A. Dataset

The information used in this research was obtained from the online resource Kaggle and is based on the publication "Distinguishing Different Stages of Parkinson's Disease Using Composite Index of Speed and Pen Pressure of Sketching a Spiral" by Zham *et al.,* published in Frontiers in Neurology in 2017 [21]. These image datasets were developed from the research in [21], where the composite index of speed (CISP) and pen pressure technique was employed to distinguish individuals with Parkinson's disease from those without it. Hand-drawn sketches were utilized, with either healthy individuals or individuals with or suspected of having Parkinson's disease sketching spiral and wavy drawings. Such drawings were included in the dataset. The images for this study consisted of 204 grayscale images, of which 60 were obtained from healthy subjects and 144 from subjects with Parkinson's disease. The images are classified into two groups: spiral and wavy sketches.

1. Data Pre-Processing: The image dataset was first separated into the spiral and wave image datasets. The images are grayscale images, meaning they consist of white and black backgrounds and have different dimensions, making them unsuitable for training the Convolutional Neural Network (CNN) deep learning architecture. Therefore, it is necessary to resize the images to ensure they have the same dimensions. This study adopts a size of 256 x 256 pixels to promote computational efficiency and accommodate the characteristics of the grayscale image dataset.

The image dataset was reviewed and cleaned to confirm that there are no unwanted images and that the images included are strictly those necessary for this study. The images were further preprocessed using image filtering and segmentation techniques. This preprocessing step is essential for feature extraction. The extracted features will be used to categorize the dataset into healthy and Parkinson's disease groups.

2. Data Split: The dataset was separated into two major folders: the spiral folder and the wavy folder. Fig. 1 shows samples of the spiral and wavy images present in the dataset.

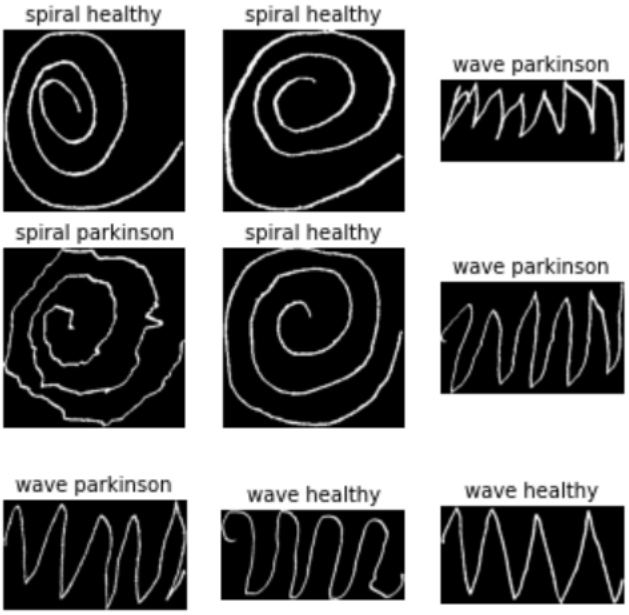

Fig. 1 Sample Image Dataset

B. Image Filtering

This study utilized an edge filter to extract necessary features based on edge points, line thickness, intersections, and the number of pixels. These features are essential for characterizing grayscale images. The study will also employ the K-nearest neighbors (KNN) algorithm to segment the images based on notable features such as the number of pixels, number of edges, and line thickness present in the images. The KNN algorithm will leverage existing similarities in the proximity of edges and intersections, which are grouped into endpoints. The proximity of these endpoints will be used to differentiate between images drawn by healthy individuals and those drawn by individuals with Parkinson's disease.

C. Model Building

In building the model, this study employed the Convolutional Neural Network (CNN) deep learning algorithm. This algorithm was selected due to its effectiveness in object detection, making it a robust tool for image classification. Fig. 2 shows the general flow of the proposed model.

1. Model Design: During the course of the study, the following design tools were included:

a. Wondershare EdrawMax: This diagramming software enables users to create and design various types of visual

diagrams. In this study, it was applied to visualize the proposed model as well as other related diagrams.

b. Microsoft Snipping Tool: This tool was used to capture images and text from reference sources and parts of the computer screen. The captured items were then edited and incorporated into this study. The diagrammatic representation of the model was created using the EdrawMax software to illustrate how the model operates. The model's operation begins with the pre-processing of data features. The dataset will be split into an 80% training set and a 20% testing set according to machine learning standards, and the features will be properly normalized before being fed into the CNN model for training. The study will yield two different models for the spiral dataset and the wavy dataset; following successful model training, the models will be evaluated.

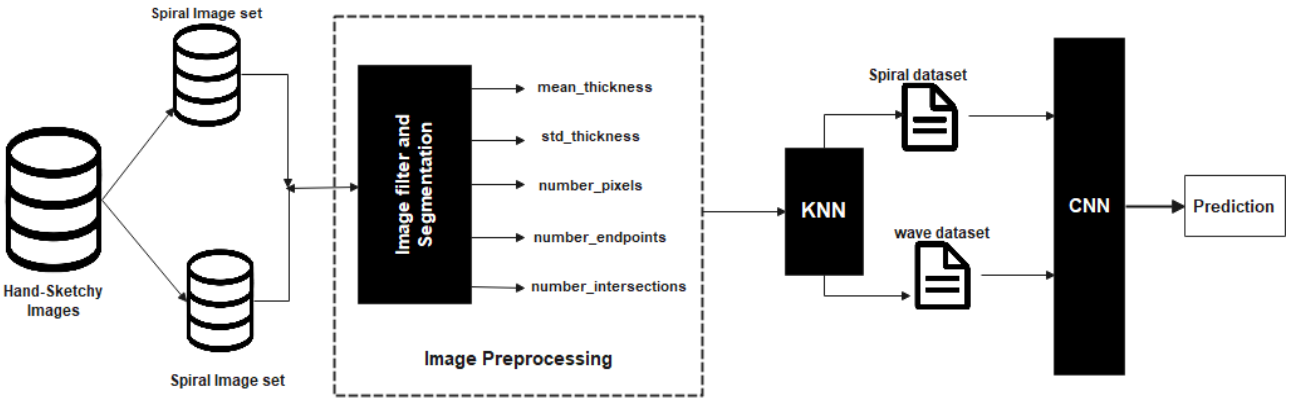

Fig. 2 The CNN-based Parkinson Disease Detection Model [Researcher's Model]

2. Model Implementation: Among the tools used for the implementation of this model are

a. TensorFlow 2.5 and Keras API: The designed model was implemented using TensorFlow, an open-source framework developed by Google, specifically designed for deep learning and other artificial intelligence solutions. TensorFlow 2.5 is well integrated with the Keras library, an application programming interface (API) that simplifies the implementation of deep learning algorithms. Keras serves as a practical neural network API built on top of the TensorFlow framework. TensorFlow was selected as the implementation framework due to its superior functionality and features compared to other leading deep learning frameworks, such as PyTorch, Theano, and Keras, which are critical for the development of advanced neural network models. Thus, TensorFlow version 2.5 was utilized in this study to develop and train the Deep Learning Ensemble Model.

b. Python Programming Language: The code for the model was developed using Python 3.x (version 3.11.4). Python was chosen for its ease of use and the availability of effective deep learning (DL) and machine learning (ML) libraries and frameworks.

c. PyCharm: The development environment used for writing and debugging the Python code was the PyCharm Integrated Development Environment (IDE). This IDE is highly effective due to its integrated code completion and suggestion features, significantly reducing development time.

d. Google Colaboratory: Commonly referred to as Colab, this web service allows users to create and execute Python programs within a browser window. It was effectively used as an environment for executing the Python code.

D. Performance Evaluation Metrics

The performance of the proposed model is evaluated using key performance metrics, including but not limited to accuracy, precision, recall, and the F1 score, among other metrics that aim to assess the effectiveness and reliability of the model.

1. Accuracy: This metric measures the percentage of the total dataset that was correctly classified by the algorithm. It is calculated using the following operation: the sum of true negatives (TN) and true positives (TP) is divided by the sum of TP, TN, false positives (FP), and false negatives (FN). True positives and true negatives refer to correctly classified instances, while false positives and false negatives represent instances where the algorithm has made classification errors. Accuracy is one of the most significant measures of a model's performance. The mathematical expression for accuracy is:

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP} \quad (1)$$

2. Precision: This metric captures the true positive (TP) rate in relation to the sum of true positives (TP) and false positives (FP). A higher positive predictive value (PPV) indicates a greater ability of the model to distinguish positive instances. Precision and recall are generally inversely related, meaning that as precision increases, recall may decrease, and when precision is low, recall can be very high. The formula for precision is:

$$\text{Precision} = \frac{TN}{TP+FP} \quad (2)$$

Where TN is True Negative, TP is True Positive, and FP is False Positive.

3. Recall: More commonly referred to as sensitivity or true positive rate, recall is the ratio of positive instances correctly identified to all actual positives. It is calculated by dividing TP by the sum of TP and FN. This metric describes how many of the actual positive instances were correctly identified by the model. The formula for recall is:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

Where TP: True Positive and FN: False Negative.

4. F1-Score: The F1 score is a summary metric that addresses the limitations of using precision or recall in isolation. It combines both metrics into a single score by calculating their harmonic mean. This is particularly useful in situations where a balance between precision and recall is necessary, helping to assess the overall performance of the model. The F1 score is computed using the following equation:

$$F1\ score = \frac{2\times precision\times recall}{precision+recall} \quad (4)$$

IV. RESULTS AND DISCUSSION

A. Data Pre-processing

The data pre-processing stage was conducted to ensure that the image datasets were properly prepared for effective machine learning algorithm training. The image datasets consisted of two types of images:

1. Waveform images
2. Spiral form images

The image datasets were categorized based on whether the images were obtained from healthy patients or patients with Parkinson's Disease. Fig. 3 shows the tabular representation of the first 15 images.

	path	img_id	disease	validation	activity
108	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V04HE03	healthy	training	spiral
157	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V09PE03	parkinson	training	spiral
40	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V09HO02	healthy	training	wave
169	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V11PE01	parkinson	training	spiral
39	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V08HO02	healthy	training	wave
177	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V05PE01	parkinson	testing	spiral
166	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V15PE02	parkinson	training	spiral
115	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V55HE01	healthy	training	spiral
136	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V55HE08	healthy	training	spiral
156	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V14PE01	parkinson	training	spiral
191	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V55HE13	healthy	testing	spiral
100	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V01HO01	healthy	testing	wave
25	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V03PO05	parkinson	training	wave
102	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V10HE02	healthy	training	spiral
146	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V03PE03	parkinson	training	spiral

Fig. 3 Tabular view of the Image dataset from the Directory

The grayscale images had varying dimensions, ranging from 168 to 512 pixels. These images were resized to a standard dimension using the NumPy array() function. A size of 256 by 256 pixels (256 × 256) was chosen. Fig. 4 shows a sample display of the dataset.

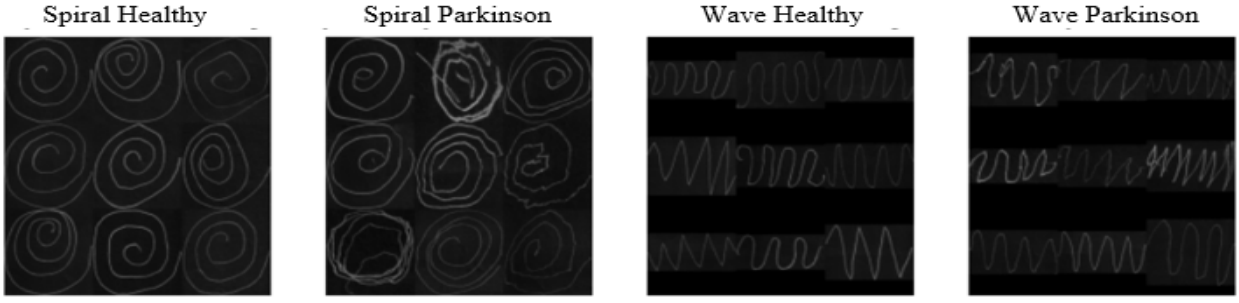

Fig. 4 Resized Image dataset to 256 x 256

The dataset was preprocessed further to generate data points, making it easier to differentiate the contours and irregularities in the images. Fig. 5 shows the data points for some of the images on the x and y axes.

All the image plots were then converted into a data frame, where each row represents a single data point for each image pixel.

	path	img_id	disease	validation	activity	x	y
310278	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V03PE01	parkinson	testing	spiral	0.667969	0.894531
134551	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V11PO01	parkinson	testing	wave	0.558594	0.231047
91033	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V55HO06	healthy	training	wave	0.429688	0.217241
26265	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V06PO03	parkinson	training	wave	0.871094	0.705357
48232	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V03PO05	parkinson	training	wave	0.855469	0.634146
80053	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V55HO02	healthy	training	wave	0.945312	0.275556
46026	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V15PO02	parkinson	training	wave	0.855469	0.500000
287877	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V08PE03	parkinson	training	spiral	0.644531	0.710938
178200	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V06HO01	healthy	testing	wave	0.376953	0.479310
275370	/content/drive/MyDrive/Colab Notebooks/Yinka-P...	V13PE02	parkinson	training	spiral	0.730469	0.796875

Fig. 5 Image dataset data points

1. Extracted Features

After pre-processing the image dataset, the following features were extracted and refined using the K-NN algorithm:

a. *Mean_thickness:* In the context of images, "mean thickness" refers to the average thickness of specific characteristics or structures within the images. Image analysis often involves measuring various aspects of objects or patterns, with thickness being one such feature.
b. *Std_thickness:* "std_thickness" refers to the standard deviation of thickness. The standard deviation measures the variability or dispersion of thickness values from the mean. In image analysis, it indicates how much individual thickness measurements deviate from the average thickness.
c. *Number_pixels:* The number of pixels refers to the total number of visual elements, or pixels, in an image. Each pixel represents the smallest unit of information in a digital image, and this feature is often used to describe the image's resolution.
d. *Number_edgepoints:* This refers to the number of points along the edges or borders of objects in an image. Edges represent regions of significant intensity changes, which are crucial for many computer vision and image processing tasks. Edge detection and analysis help identify and characterize shapes and structures within the image.
e. *Number_intersections:* This refers to the number of points in an image where lines or contours intersect. These intersections can serve as important features in computer vision and image processing applications.

These features were normalized and standardized using the StandardScaler() method. The two classes were encoded as 0 and 1, where 0 denotes healthy patients and 1 denotes patients with Parkinson's Disease.

Figs. 6 and 7 were obtained during the training process of the spiral image dataset. The figures show the original image dimensions, the number of training images, the number of testing images, and the total number of images used.

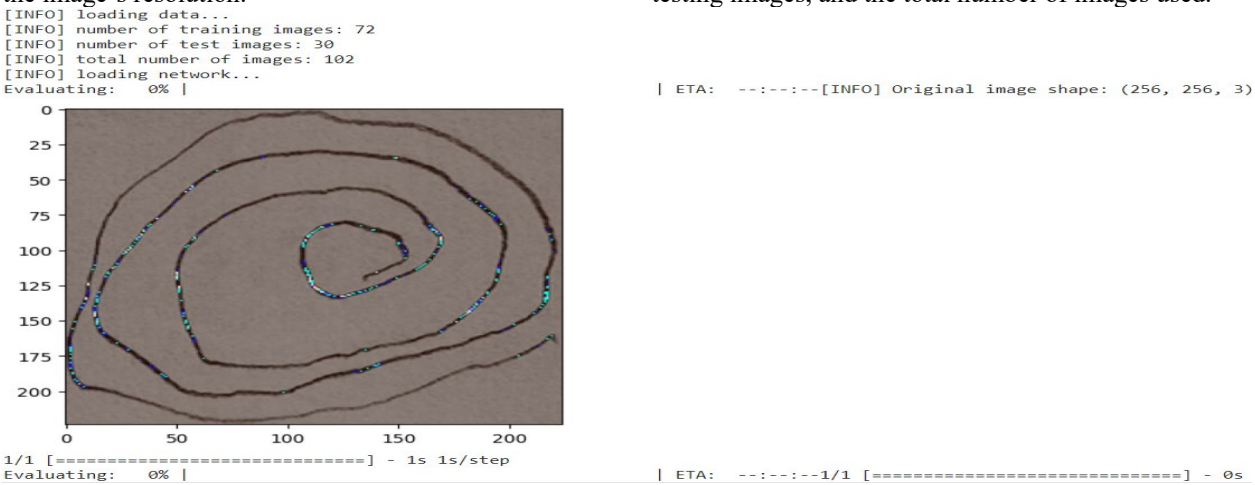

Fig. 6 Spiral image dataset feature training process

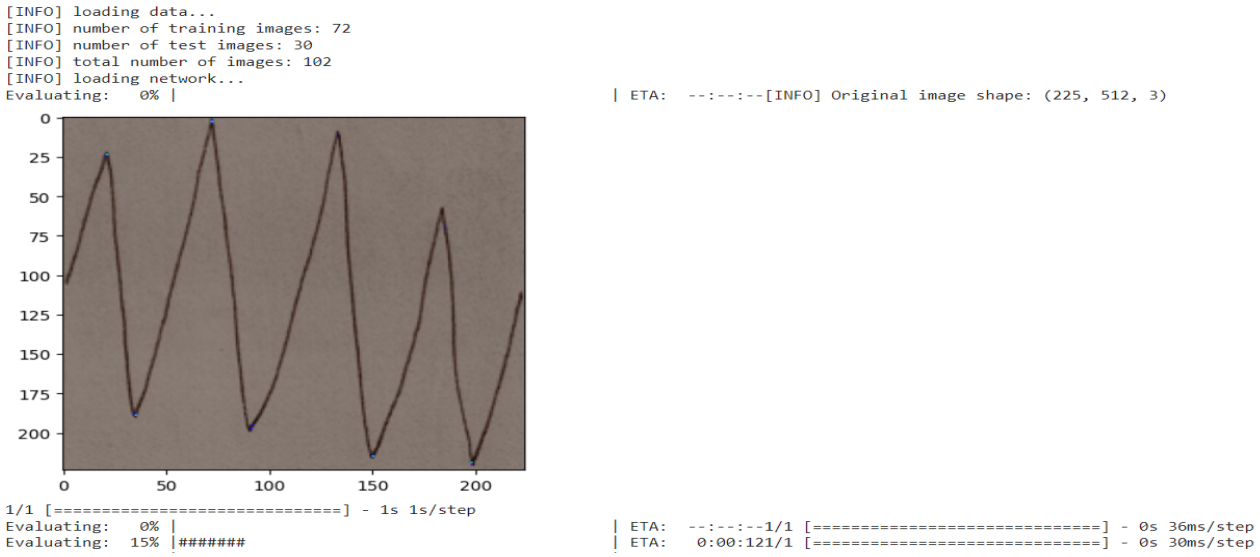

Fig. 7 Wave image dataset feature training process

B. CNN Model Training

In designing the CNN model, we considered important parameters and hyper parameters required for effective model training. In addition to the identified features, deeper CNN features are automatically detected based on the number of layers required for training. Table 1 shows the CNN parameters used and their values. The step-by-step classification system within the convolutional neural network (CNN) architecture is described below.

1. Model Architecture

a. The architecture of the CNN model is built using the Sequential API provided by Keras.
b. The layers include Convolution layers (Conv2D), ReLU layers, MaxPooling2D layers, Dropout layers, and Fully Connected (Dense) layers.
c. The final layer utilizes the Sigmoid activation function to perform binary classification, scaling between zero and one.

2. Data Loading

a. Image data is loaded from the specified directory using the Path class from the pathlib module.
b. The training and testing datasets are organized into DataFrames using Pandas, with paths and corresponding labels.

3. Data Augmentation: The ImageDataGenerator from Keras was used for data augmentation, including rotation, shifting, brightness adjustment, horizontal and vertical flipping, and a pre-processing function (contrast stretching).

4. Data Splitting: Data is split using the validation split attribute, which divides the data into training and validation sets.

5. Model Compilation: The model is compiled using the binary cross-entropy loss function and the Adam optimization algorithm. The model's performance is evaluated using the accuracy metric.

6. Model Training

a. The model is trained using the fit_generator function for both training and validation set generators. During training, steps per epoch and validation steps are specified.
b. Callbacks such as TensorBoard, learning rate reduction, early stopping, and model check pointing are employed during training.

7. Hyperparameter Tuning

a. Hyperparameter tuning is performed using Grid Search with logistic regression.
b. The parameter grid includes the learning rate "C" with values of 0.00001, 0.0001, 0.001, and 1.0.

TABLE I MODEL PARAMETERS AND VALUES

Parameters	Type	Value
15 Layers	Conv2D, activation, Maxpooling2D, Dense, Dropout, Flaten	
Activation Function		Relu, sigmoid
Pool_size		(2,2)
Strides		(2,2)
Dropout		0.25
Epoch		100
Validation_steps		400
Batch_size		32

C. Model Architecture Description

1. *Layers:* The model consists of a total of 15 layers, including Conv2D, Activation, MaxPooling2D, Dropout, Flatten, and Dense layers.
2. *Conv2D Layers*: The model employs five convolutional layers with an increasing number of filters (32, 64, 64, 128, 128). These layers progressively extract more

complex features from the input data as it passes through them.

3. *Activation Functions:* The ReLU activation function is used in various layers of the model; except for the final output layer, all layers convert outputs using this activation function.
4. *Pooling Layers:* MaxPooling2D layers are applied after every two convolutional layers, down sampling the feature maps by a factor of 2 in both dimensions. This reduces the complexity of the data and helps control overfitting.
5. *Dropout Layers:* Regularization through dropout layers randomly removes a certain percentage of neurons to prevent overfitting, allowing the model to learn more abstract concepts.
6. *Dense Layers:* There are two dense layers in this architecture: one consists of 128 neurons, and the other consists of 1 neuron. The final output layer, with a single neuron, uses a sigmoid activation function for binary classification.
7. *Flatten Layer:* This layer converts the multi-dimensional feature maps produced by the convolutional layers into a one-dimensional structure, which is then fed to the dense layers for processing.
8. The CNN model utilized the parameters listed in Table I to generate additional features from the image dataset. The total number of parameters generated for the spiral image model and wave image model are 4,480,993 and 8,675,297, respectively. All parameters are trainable. Fig. 8 shows the summary of the developed model.

Layer (type)	Output Shape	Param #
conv2d_24 (Conv2D)	(None, 128, 128, 32)	320
activation_32 (Activation)	(None, 128, 128, 32)	0
conv2d_25 (Conv2D)	(None, 128, 128, 32)	9248
activation_33 (Activation)	(None, 128, 128, 32)	0
max_pooling2d_12 (MaxPooling2D)	(None, 64, 64, 32)	0
dropout_16 (Dropout)	(None, 64, 64, 32)	0
conv2d_26 (Conv2D)	(None, 64, 64, 64)	18496
activation_34 (Activation)	(None, 64, 64, 64)	0
conv2d_27 (Conv2D)	(None, 64, 64, 64)	36928
activation_35 (Activation)	(None, 64, 64, 64)	0
max_pooling2d_13 (MaxPooling2D)	(None, 32, 32, 64)	0
dropout_17 (Dropout)	(None, 32, 32, 64)	0
conv2d_28 (Conv2D)	(None, 32, 32, 128)	73856
activation_36 (Activation)	(None, 32, 32, 128)	0
conv2d_29 (Conv2D)	(None, 32, 32, 128)	147584
activation_37 (Activation)	(None, 32, 32, 128)	0
max_pooling2d_14 (MaxPooling2D)	(None, 16, 16, 128)	0
dropout_18 (Dropout)	(None, 16, 16, 128)	0
flatten_4 (Flatten)	(None, 32768)	0
dense_8 (Dense)	(None, 128)	4194432
activation_38 (Activation)	(None, 128)	0
dropout_19 (Dropout)	(None, 128)	0
dense_9 (Dense)	(None, 1)	129
activation_39 (Activation)	(None, 1)	0

Fig. 8 Model Summary

D. Result Summary

For validation purposes, the spiral and wave image sets were used separately, with the model's performance assessed using various metrics, including accuracy, precision, recall, and F1-score. These metrics were considered for classifying healthy spiral images and those of Parkinson's Disease patients, as described in Tables II and III. A similar result was obtained for the wave image sets; however, there was a slight change in the accuracy readings between the two scenarios. The accuracy measures of the two CNN models are sufficient for predicting Parkinson's disease from given sketch diagrams. These accuracies are dependent on the learning rates set during model training.

TABLE II SPIRAL CNN MODEL RESULT ON LEARNING RATES

Learning Rate	Patient Type	Accuracy	Precision	Recall	F1-Score
0.00001	Healthy Patient	97.93	74	93	82
	PD Patient		91	67	77
0.0001	Healthy Patient	97.93	82	93	87
	PD Patient		92	80	86
0.001	Healthy Patient	97.93	82	93	87
	PD Patient		92	80	86
0.01	Healthy Patient	97.93	81	87	84
	PD Patient		86	80	83
1.0	Healthy Patient	97.93	87	87	87
	PD Patient		87	87	87

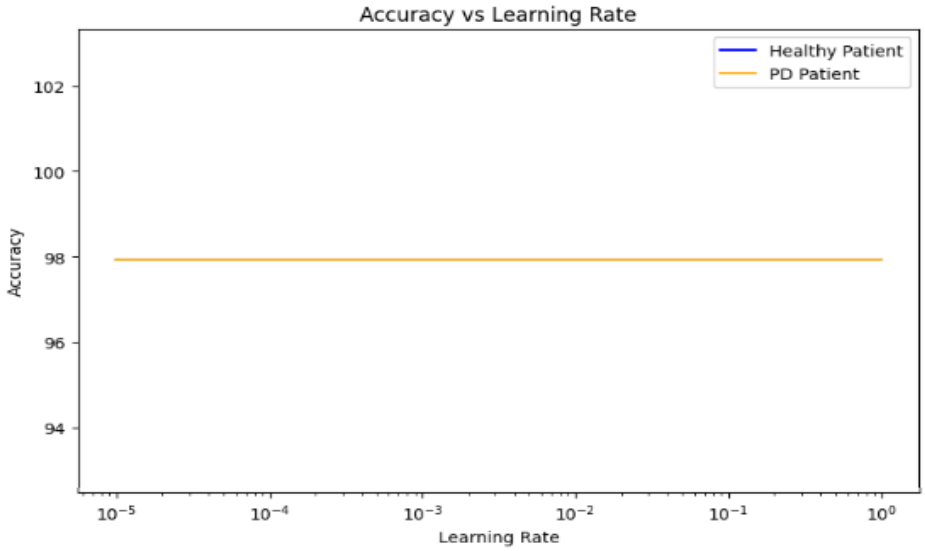

Fig. 9 Accuracy Vs Learning Rate of Spiral CNN Model

Fig. 10 Precision Vs Learning Rate of Spiral CNN Model

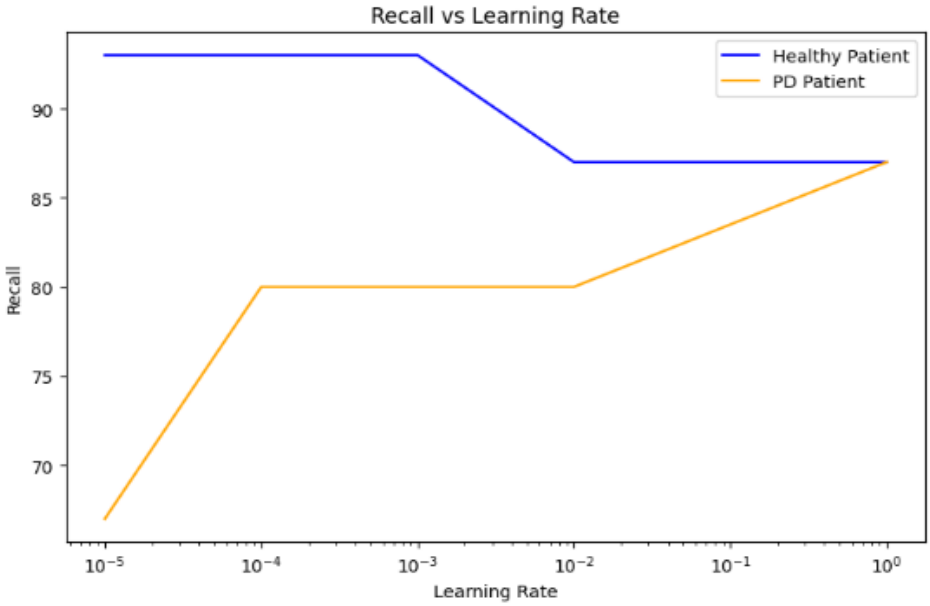

Fig. 11 Recall Vs Learning Rate of Spiral CNN Model

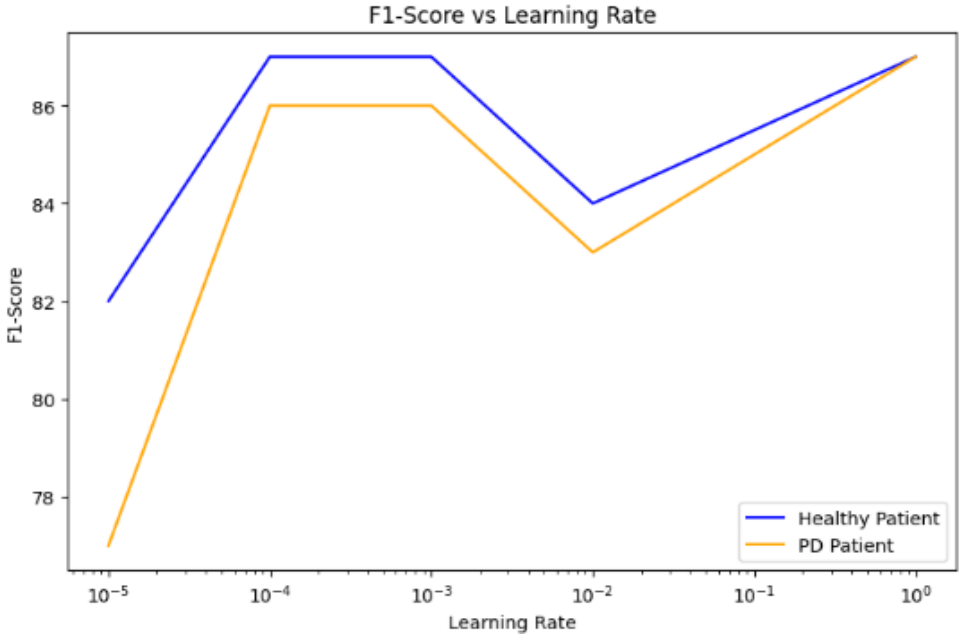

Fig. 12 F1-Score Vs Learning Rate of Spiral CNN Model

TABLE III WAVE CNN MODEL RESULT ON A LEARNING RATE OF 0.00001

Learning Rate	Patient Type	Accuracy	Precision	Recall	F1-Score
0.00001	Healthy Patient	97.93	92	73	81
	PD Patient		78	93	85
0.0001	Healthy Patient	97.93	92	80	86
	PD Patient		82	93	87
0.001	Healthy Patient	97.93	92	80	86
	PD Patient		82	93	87
1.0	Healthy Patient	97.93	92	80	86
	PD Patient		82	93	87

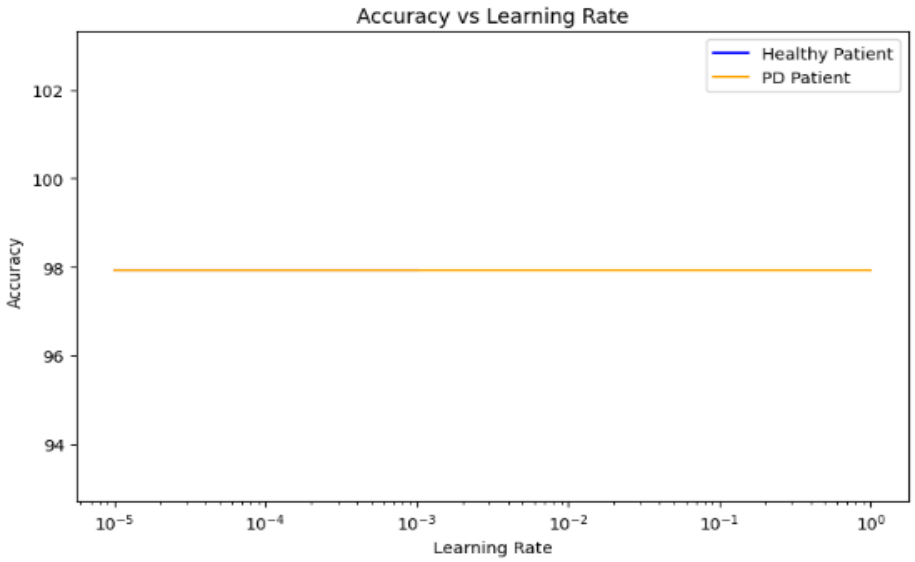

Fig. 13 Accuracy Vs Learning Rate of Wave CNN Model

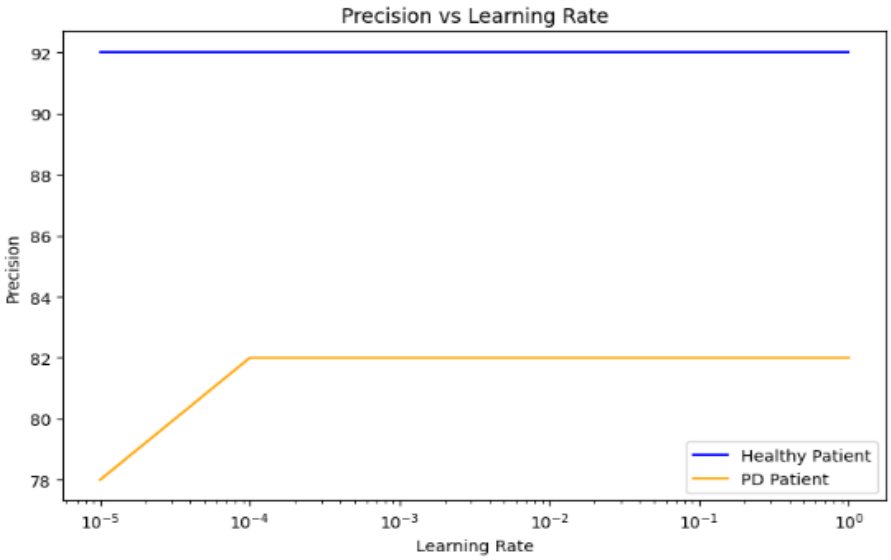

Fig. 14 Precision vs Learning Rate of Wave CNN Model

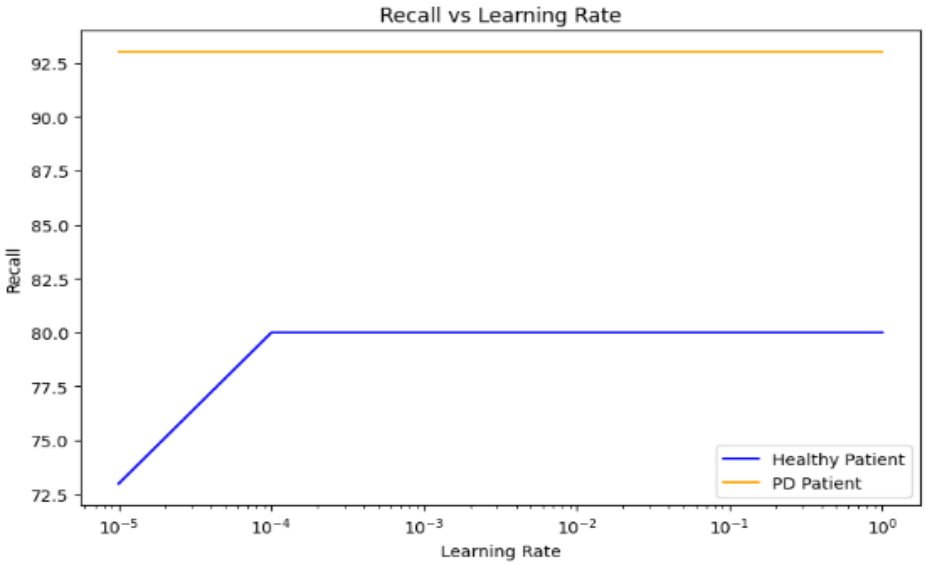

Fig. 15 Recall Vs Learning Rate of Wave CNN Model

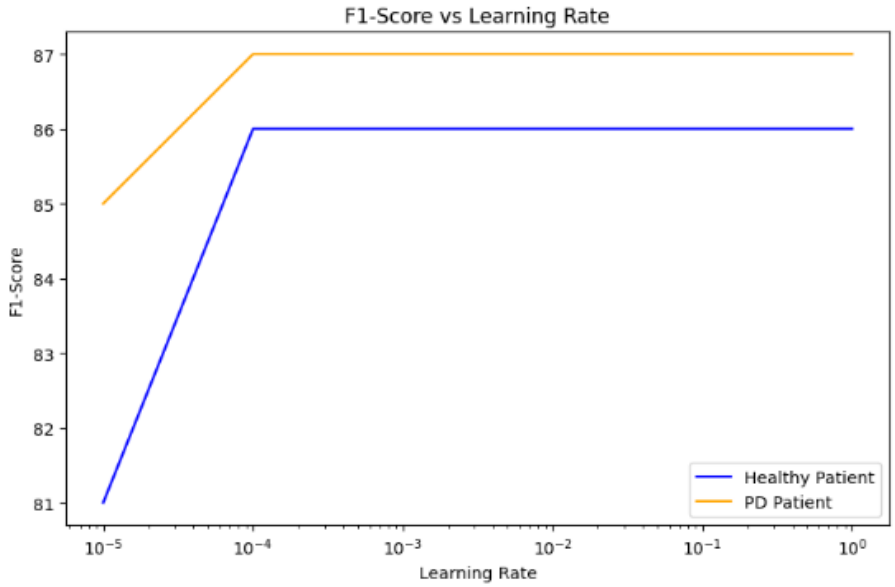

Fig. 16 Accuracy vs Learning Rate of Wave CNN Model

1. Overall Accuracy

a. *Spiral CNN:* The accuracy for both healthy and Parkinson's Disease patients remains almost constant (around 97.93%) across all learning rates, except for 0.01, which shows a slight drop for Parkinson's Disease patients (86%).

b. *Wave CNN:* The accuracy for both healthy and Parkinson's Disease patients shows a wider range, with the highest accuracy at 0.00001 (92% for healthy patients and 78% for Parkinson's Disease patients) and the lowest at 1.0 (92% for healthy patients and 82% for Parkinson's Disease patients).

2. Precision and Recall

a. *Spiral CNN:* Both precision and recall remain relatively stable for healthy patients across all learning rates. However, for Parkinson's Disease patients, there are some fluctuations. At 0.001 and 0.01, both precision and recall drop slightly compared to other learning rates.

b. *Wave CNN:* Similar to accuracy, precision and recall for both healthy and Parkinson's Disease patients vary more broadly across learning rates. Generally, higher learning rates lead to higher precision for healthy patients but lower precision for Parkinson's Disease patients. Recall tends to be more stable for healthy patients but shows some variation for Parkinson's Disease patients.

3. F1-Score

a. *Spiral CNN:* Similar to precision and recall, the F1-score remains stable for healthy patients but shows some variation for Parkinson's Disease patients, with the highest score at 0.0001 (87) and the lowest at 0.01 (84).

b. *Wave CNN:* The trend for the F1-score mirrors that of precision and recall, with higher learning rates leading to a higher F1-score for healthy patients but a lower F1-score for Parkinson's Disease patients.

4. Comparison between Spiral CNN and Wave CNN

Overall, the Spiral CNN appears to be more robust to changes in learning rate, maintaining consistent performance across all values. The Wave CNN seems to be more sensitive to the learning rate, with both higher and lower performance depending on the chosen value. In terms of accuracy, both datasets perform well; however, the Spiral CNN achieves slightly higher accuracy for healthy patients, while the Wave CNN achieves slightly higher accuracy for Parkinson's Disease patients at the lowest learning rate.

V. CONCLUSION

This study implements a CNN model for grade classification of the Parkinson's disease database and focuses on the effect of changing the learning rate on the model's performance. However, research has shown that CNNs can predict the incidence of the disease; further investigation is required to understand how variations in the learning rate affect the effectiveness and stability of the model. The study utilized a dataset containing hand-drawn sketches from both Parkinson's disease patients and healthy individuals, employing KNN for feature extraction and convolutional layers for feature generation. The CNN design incorporated dense layers for classification. During training, different learning rates were carefully examined, and pre-processing techniques such as normalization and augmentation were applied to enhance model generalization. The study found that higher learning rates initially accelerated convergence but degraded accuracy, while lower rates exhibited more consistent convergence but required longer training epochs. The optimal learning rate significantly impacts model convergence and accuracy, providing valuable insights for future model development. The findings have practical implications for improving the predictability of Parkinson's disease models. Understanding the influence of learning rates aids in designing more robust and efficient diagnostic tools,

potentially facilitating early intervention and personalized treatment regimens. One limitation is the reliance on a single dataset, which may contain biases. Furthermore, the study focused on a specific CNN architecture; alternative hyper parameters warrant further investigation in future research.

REFERENCES

[1] N. Boualoulou, M. Miyara, B. Nsiri, and T. Belhoussine Drissi, "A novel Parkinson's disease detection algorithm combined EMD, BFCC, and SVM classifier," *Diagnostyka*, vol. 24, no. 4, pp. 1-10, Oct. 2023, doi: 10.29354/diag/171712.

[2] S. Latif *et al*., "Dopamine in Parkinson's disease," *Clin. Chim. Acta*, vol. 522, pp. 114-126, Nov. 2021, doi: 10.1016/j.cca.2021.08.009.

[3] E. I. Abdukodirov, K. M. Khalimova, and R. J. Matmurodov, "Hereditary-genealogical features of Parkinson's disease and their early detection of the disease," *Int. J. Health Sci. (Qassim)*, Apr. 2022, doi: 10.53730/ijhs.v6nS1.5802.

[4] S. S, A. S, G. V. V. Rao, P. V, K. Mohanraj, and R. Azhagumurugan, "Parkinson's disease prediction using machine learning algorithm," in *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, IEEE, Dec. 2022, pp. 1-5, doi: 10.1109/ICPECTS56089.2022.10047447.

[5] L. di Biase *et al*., "Gait analysis in Parkinson's disease: An overview of the most accurate markers for diagnosis and symptoms monitoring," *Sensors*, vol. 20, no. 12, p. 3529, Jun. 2020, doi: 10.3390/s20123529.

[6] D. Palacios-Alonso, G. Melendez-Morales, A. Lopez-Arribas, C. Lazaro-Carrascosa, A. Gomez-Rodellar, and P. Gomez-Vilda, "MonParLoc: A speech-based system for Parkinson's disease analysis and monitoring," *IEEE Access*, vol. 8, pp. 188243-188255, 2020, doi: 10.1109/ACCESS.2020.3031646.

[7] M. Olson, T. E. Lockhart, and A. Lieberman, "Motor learning deficits in Parkinson's disease (PD) and their effect on training response in gait and balance: A narrative review," *Front. Neurol.*, vol. 10, Feb. 2019, doi: 10.3389/fneur.2019.00062.

[8] Y. Kumar, A. Koul, R. Singla, and M. F. Ijaz, "Artificial intelligence in disease diagnosis: A systematic literature review, synthesizing framework and future research agenda," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 7, pp. 8459-8486, Jul. 2023, doi: 10.1007/s12 652-021-03612-z.

[9] P. N. Srinivasu, N. Sandhya, R. H. Jhaveri, and R. Raut, "From blackbox to explainable AI in healthcare: Existing tools and case studies," *Mob. Inf. Syst.*, vol. 2022, pp. 1-20, Jun. 2022, doi: 10.1155/ 2022/8167821.

[10] M. Alissa *et al*., "Parkinson's disease diagnosis using convolutional neural networks and figure-copying tasks," *Neural Comput. Appl.*, vol. 34, no. 2, pp. 1433-1453, Jan. 2022, doi: 10.1007/s00521-021-06469-7.

[11] Y. Zhang and Y. Ma, "Application of supervised machine learning algorithms in the classification of sagittal gait patterns of cerebral palsy children with spastic diplegia," *Comput. Biol. Med.*, vol. 106, pp. 33-39, Mar. 2019, doi: 10.1016/j.compbiomed.2019.01.009.

[12] S. M. A. Asaduzzaman Sakib, A. F. M. Nazmus Shusmita, and S. A. Kabir, "Detection of Parkinson's disease from neuro-imagery using deep neural network with transfer learning," BRAC University, 2020. [Online]. Available: https://dspace.bracu.ac.bd/xmlui/handle/10361/ 14457.

[13] J. T. J. Goschenhofer, F. M. J. Pfister, K. A. Yuksel, B. Bischl, and U. Fietzek, "Wearable-based Parkinson's disease severity monitoring using deep learning," 2019. [Online]. Available: https://ecmlpkdd 2019.org/downloads/paper/575.pdf.

[14] M. B. Makarious *et al*., "Multi-modality machine learning predicting Parkinson's disease," *npj Park. Dis.*, vol. 8, no. 1, p. 35, Apr. 2022, doi: 10.1038/s41531-022-00288-w.

[15] N. Khoury, F. Attal, Y. Amirat, L. Oukhellou, and S. Mohammed, "Data-driven based approach to aid Parkinson's disease diagnosis," *Sensors*, vol. 19, no. 2, p. 242, Jan. 2019, doi: 10.3390/s19020242.

[16] A. Al Imran, A. Rahman, H. Kabir, and S. Rahim, "The impact of feature selection techniques on the performance of predicting Parkinson's disease," *Int. J. Inf. Technol. Comput. Sci.*, vol. 10, no. 11, pp. 14-29, Nov. 2018, doi: 10.5815/ijitcs.2018.11.02.

[17] K.-M. Giannakopoulou, I. Roussaki, and K. Demestichas, "Internet of Things technologies and machine learning methods for Parkinson's disease diagnosis, monitoring and management: A systematic review," *Sensors*, vol. 22, no. 5, p. 1799, Feb. 2022, doi: 10.3390/s22051799.

[18] P. Arora, A. Mishra, and A. Malhi, "Diagnosis of Parkinson's disease genes using LSTM and MLP-based multi-feature extraction methods," *Int. J. Data Min. Bioinform.*, vol. 27, no. 4, pp. 326-348, 2023, doi: 10.1504/IJDMB.2023.134301.

[19] L. Ali, C. Zhu, N. A. Golilarz, A. Javeed, M. Zhou, and Y. Liu, "Reliable Parkinson's disease detection by analyzing handwritten drawings: Construction of an unbiased cascaded learning system based on feature selection and adaptive boosting model," *IEEE Access*, vol. 7, pp. 116480-116489, 2019, doi: 10.1109/ACCESS.2019.293 2037.

[20] S. Chakraborty, S. Aich, J. S. Sim, E. Han, J. Park, and H. C. Kim, "Parkinson's disease detection from spiral and wave drawings using convolutional neural networks: A multistage classifier approach," in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, IEEE, Feb. 2020, pp. 298-303, doi: 10.23919/ ICACT48636.2020.9061497.

[21] P. Zham, D. K. Kumar, P. Dabnichki, S. P. Arjunan, and S. Raghav, "Distinguishing different stages of Parkinson's disease using composite index of speed and pen-pressure of sketching a spiral," *Front. Neurol.*, vol. 8, Sep. 2017, doi: 10.3389/fneur.2017.00435.