International Journal of Software Innovation
Volume 11 • Issue 1

A Novel Approach to Parkinson's Disease Progression Evaluation Using Convolutional Neural Networks

Mhamed Zineddine, INSA Euromed, Euromed Research Center, Euromed University of Fes, Morocco*

ABSTRACT

Parkinson's disease (PD) is a devastating disorder with serious impacts on the health and quality of life for a wide group of patients. While the early diagnosis of PD is a critical step in managing its symptoms, measuring its progression would be the cornerstone for the development of treatment protocols suitable for each patient. This paper proposes a novel approach to digital PPMI measures and its combination with spirals drawings to increase the accuracy rate of a neural network to the maximum possible. The results show a well performing CNN model with an accuracy of 1(100%). Thus, the end-users of the proposed approach could be more confident when evaluating the progression of PD. The trained, validated, and tested model was able to classify the PD's progression as High, Medium, or Low, with high sureness.

KEYWORDS

CNN, Neural Network, Parkinson, Parkinson Progression, PPMI

1. INTRODUCTION

Individuals in their 60s are increasingly affected by neurological conditions. Parkinson's disease (PD) is the second most common neurological syndrome in the central nervous system (Benba et al. 2016a). There is a consensus among neurologists and researchers that Parkinson's is caused by aberrations in dopamine signaling in the brain; that is, the dopaminergic neurons fail to release enough signaling substance (dopamine) due to the demise of a significant percentage of them. Bradykinesia (slowness of movement), dysphonia (voice impairment), rigidity, tremor, and poor balance are common symptoms of PD that raise alerts regarding the loss of dopaminergic neurons in the substantia nigra of the brain (Diaz et al. 2022, Louis et.al 2015, Duffy, 2013), that is, the etiology of PD (Kouli et al., 2018). Multiple techniques have been used to detect PD at early stages. For instance, the accuracy of Magnetic Resonance Imaging (MRI) of the brain has increased; thus, MRI has been a part of PD diagnosis (Heim, et al. 2017, Fioravanti, et al. 2015). Physicians have visually and quantitatively interpreted MR images based on changes in brain structure and different types of tissues to identify

DOI: 10.4018/IJSI.315655 *Corresponding Author

PD (Tolosa et al.2021, Pyatigorskaya et al., 2018, Heim, 2017, Sterling et al. 2016, Fioravanti, 2015). Moreover, acoustic analysis of patients could help detect voice impairments, which is one of the signs of early PD (Fernández-García et al. 2021, Al-Fatlawi et al., 2016).

Certainly, each PD patient experiences dissimilar progression, cadence, occurrence, evolution and severity of symptoms (Raket et al., 2022), therefore, monitoring these manifestations is of great medical value in order to identify patients with a higher risk of rapid disease progression at early or at advanced stages of the disorder. The benefit of gauging the disease progression is to reassess the level of attention and monitoring required for these patients to cope with the aggressive evolution of their PD symptomatology. Consequently, this helps to ensure an acceptable level of quality of life, which requires careful adjustment of the PD's management plans.

Isolating potential markers that accelerate PD's progression could provide insights into the intricate mechanisms behind the manifestation and evolution of symptoms. Studying patients who experience faster worsening of PD, offers a golden opportunity to understand the disease and the factors contributing to such rapid progression. These factors might be amplified and thus easier to detect (e.g., brain connectivity networks, enzymes, proteins, and variations in metabolic paths). Cerebrospinal fluid (CSF) is often used to detect changes in the homeostasis of the central nervous system. Furthermore, the most accurate approach for determining the progression rate of PD is to estimate the deterioration status of specific brain networks using imagery (Sterling et al. 2016). However, motor and non-motor symptoms evaluation is easier and more cost effective. For instance, Unified Parkinson's Disease Rating Scale (UPDRS) offers accurate alternatives for rating PD progression; however, computer-assisted technologies can be used to monitor patients and their motor impairments, regardless of the probable uncertainty in the clinical ratings (Lu et al. 2021). Furthermore, motor symptoms progression differs from patient to patient; thus, it can be used as a metric to monitor the haste (rapid or slow) of the progression of the disorder. Roede et al. (2013) reported considerable variations in the annual worsening rate between the rapid (5.95) and slower (1.45) groups using UPDRS-III. Moreover, Fereshtehnejad et al. (2015) suggested that orthostatic hypotension, a rapid eye movement sleep behavior disorder, and signs of cognitive impairment could signal the decline of motor symptoms in patients. Undoubtedly, non-motor symptoms attributed to PD have an influence on patients and must be monitored and treated. For instance, cognitive impairment can be predicted based on the patient's age, presence of olfactory disturbances, sleep disturbances, and CSF biomarkers, as suggested by Schrag et al. (2017). In recent years, much attention has been given to the development of diagnostic systems based on artificial intelligence concepts, such as machine learning, and data mining to detect PD. A variety of data sources have been available to provide valuable datasets, such as handwritten patterns, speech recordings, voice signals, physiological signals, and collected data from wearable sensors for gait and others. These datasets can be used to predict PD using neural networks (Liaqat et al. 2019a; Liaqat et al. 2019b; Al-Fatlawi et al., 2019; Parisi et al., 2018 ; Naranjo et al.,2017a; Guruler et al. 2017 ; Naranjo et al., 2016; Sakar et al., 2013 ; Tsanas et al., 2012; Rigas et al. 2012; Das et al., 2010; Little et al. 2009). For instance, speech recordings, as a dataset, have been the target of multiple techniques used to detect PD. Little et al. (2009) applied a support vector machine (SVM) to replicate data from 31 subjects, which, resulted in 91.4% detection accuracy. Furthermore, complex-valued artificial neural networks with k-means clustering and feature weighting approaches had been applied to the same dataset, which increased the classification accuracy to 99.52% (Orozco-Arroyave et al. 2016). Despite the promising reported results, the issue is that the dataset is imbalanced; thus, most machine learning algorithms, which are very sensitive to imbalanced classes, led to flawed outcomes. Moreover, most studies used the conventional k-fold cross-validation approach. This ensured that all observations from the original dataset had the chance to appear in the training and test sets. However, this causes a subject overlap, which is a major issue. Subsequently, new balanced datasets had been created, such as the training and testing databases collected by Sarkar et al. (2013). The authors applied the k-nearest neighbor model and SVM classifiers to these new datasets, which surprisingly resulted in a 55% accuracy rate.

To improve accuracy, a variety of feature selection approaches had been tested (Guruler et al. 2017, Parisi et al. 2018, Naranjo et al. 2017b, Benba et al., 2016b, Canturk and F. Karabiber, 2016, Li, 2017, Benba, 2017, Ozcift, 2012, Al-Fatlawi et al., 2019). For instance, Canturk and Karabiber explored a variety of feature selection and classification algorithms and achieved an accuracy of 57.5% under leave-one-subject-out (LOSO) cross-validation (CV) (Canturk and F. Karabiber, 2016). Moreover, hybrid feature learning and SVM adopted by Li et al. achieved 82.5% accuracy (Li et al., 2017); further, mel-frequency cepstral coefficients (MFCCs) and SVM by Benba et al. (2016b) also achieved an accuracy of 82.5% for LOSO and CV. Other authors had sought to improve the accuracy rate by enhancing the datasets. For instance, Benba et al. (2017) achieved 87.5% accuracy using vowel samples and a subset of human factor cepstral coefficients (HFCCs), which is a transformation in temporal domains. Furthermore, voice recordings of PD patients and healthy subjects were used in a deep-learning setup aimed at transition detection by Vasquez-Correa et al. (2018). The use of AI techniques to detect and predict various diseases shows tremendous potential, thus, they have flourished. In addition to PD detection, more diseases had benefited from these AI techniques. The classification of hepatitis disease with an accuracy of 94.16% (Dogantekin et al., 2009), the classification of liver images using ResNet50 with an accuracy of 86.4% (Pasyar et al. 2021), the automatic detection of epilepsy (Subasi and Gursoy, 2010), and the classification of diabetes (Asgari et al., 2021) are a few successful applications of AI concepts.

There is no doubt that the PD progression rate is of significant medical value after diagnosis, and the use of AI concepts to predict PD is promising (Oludare et al. 2018). However, many challenges have to be mitigated to reach the set promises. For instance, while gait disturbances are major factors involved in the motor manifestation of PD, traditional motor deficit tests, such as the Unified Parkinson's Disease Rating Scale (UPDRS), fail to detect minor changes in people at risk or have PD. Therefore, more sensitive motor function tests are required to improve the likelihood of detecting pre-diagnosis motor alterations. Thus, ideas such as combining quantitative gait and mobility measurements and spiral drawings might be a sound technique to monitor pre-diagnosis alterations, as well as illness progression.

In line with efforts to ease these challenges and motivated by the significance and importance of PD progression monitoring to patients and their caregivers, the author of this paper proposes a novel approach to create a dataset of graphical profiles based on data received from the Parkinson's Progression Markers Initiative (PPMI) and apply convolutional neural networks to assess PD progression.

The remainder of this paper is organized as follows: Section (2) presents the methodology and tools; section (3) exhibits and discusses the results of the study; section (4) discusses the limitations and recommendations; and the last section concludes this research article.

2. METHODOLOGY AND TOOLS

Within the framework of the Parkinson's PPMI, a gait study has been conducted to acquire quantitative and objective motor measurements that might provide information on pre-clinical symptoms, progression indicators, and dynamic changes in functions during the disorder, as well as potential modifiers and mediators of motor symptoms. The gait study is part of the PPMI genetic cohort project to supplement the present PPMI procedure and dataset with additional data. The system used to collect the data consisted of three lightweight wireless wearable sensors with three axial sensors (accelerometer, gyroscope, and magnetometer). The system measures the acceleration of movement on three orthogonal axes, in addition to gait measurements as a function of time. The protocol adopted included an assessment of each of the participants for every visit. Data were collected using six tests and validated in a large cohort. The first test consisted of swaying for 30s with eyes open, which was used to capture the mass displacement during normal conditions; the second test was the same as the first test, but with closed eyes, which is used to capture the mass displacement during challenging

conditions; the third test includes Timed Up and Go (TUG 1), which is designed to assess movement, transfers, and turns; the fourth test, Timed Up and Go (TUG 2), which consists of repeated tasks; the fifth and sixth tests include usual walking for 1 min with preferred walking speed, and dual task walk for 1 min while computing some basic operations such as subtractions.

The dataset, as described in Appendix (A), has been subjected to a rigorous process of cleanup, normalization, and missing values generation using stochastic regression imputation. All measured values were normalized to make them suitable for constructing meaningful profiles. Using SPSS 22, the cleaned data were imported and a regression model was built to predict the observed values based on other variables when tolerated.

The generated model was used to produce the values (fitted values) for cases with missing measurements. The choice of stochastic regression is supported by its error handling as it adds the average regression variance to the regression imputations. Moreover, this technique has less bias than the mean and regression imputations (Zhang et al. 2022; López et al 2021). The resulting dataset was filtered to select only subjects that had been assessed at least twice, in order to create subject's PD progression profiles. The operation of creating profiles requires at least two values measured at two different times (months or years). The records of each participant were used to compute the degree of change in PD symptoms (i.e., PD progression) during the assessment period. However, a set of coefficients totaling to one was used to rheostat the effect of each variable. Three classes were identified to categorize the PD progress: a class named High (i.e., fast), which is used to label profiles with fast progression, a class named Medium (i.e., typical), and a class named Low (i.e., slow), each of which labels medium and low PD progression respectively. To construct the profile images, Python programming language was used to manipulate the data. The values for each subject are plotted using different symbols and colors (Figure (1), Figure (6)). The same symbols and colors were preserved for each variable of each participant's record. The spirals drawings dataset has to be converted to images, since the dataset is delimited as CSV values in the form of X; Y; Z; Pressure; Grip Angle; Timestamp; and Test ID. The latter is used to identify the test type, 0 for the static spiral test (draw on the given spiral pattern), 1 for Dynamic Spiral Test (spiral pattern will blink in a certain time, so subjects need to continue on their draw), and 2 for the circular motion test (subject draws circles around a red point). A Python script was used to generate spirals for both periods using the data collected during two different visits, as shown in Figure (2). The plotted values are superimposed on the two generated spirals of the same subject with respect to the same visits (Figure (9)); that is, the spirals are used as a background (see Figure (13)). The three datasets (dataset 1: profile using measurement values provided by PPMI, dataset 2: spirals drawings generated from the data collected, dataset 3: profiles and spirals drawings are superimposed and used to train, validate, and verify the generated model, and eventually check the performance indicators such as accuracy and loss. The resulting model must be tuned to improve its performance. A variety of techniques for performance improvement have been gaining ground, such as hyper-parameter tuning, data redesigning, data augmentation, and model optimization. In this study, data augmentation was adopted, and the profile images were rotated to change their orientation. Moreover, the most used CNN optimizers (sgd, rmsprop, adagrad, adam) are used to identify the best. The number of epochs reflects the number of times the weights of the network were adapted. The validation and training error behaviors are deciding factors to choose the epoch number; as long as both validation and training errors are dropping, training should continue; however, it should stop otherwise or overfitting will be an issue (see Figure (3)).

2.1. VALIDATION METHODS AND EVALUATION METRICS

2.1.1. Validation Methods

Various validation methods have been suggested in the literature to enhance and evaluate the performance of a learning model (Maleki et al. 2020). These include k-fold, train-test holdout, and

Figure 1. Profile from PPMI data collected

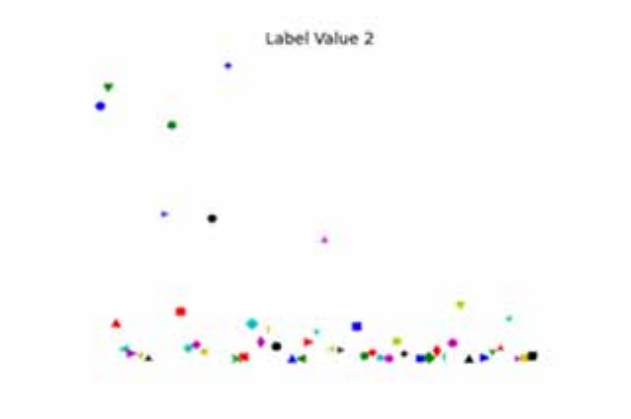


Figure 2. Generated spiral drawing

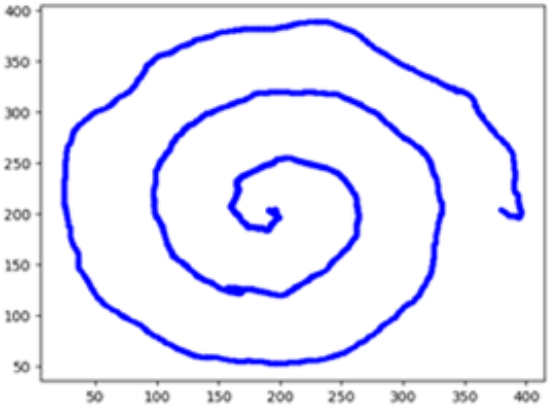
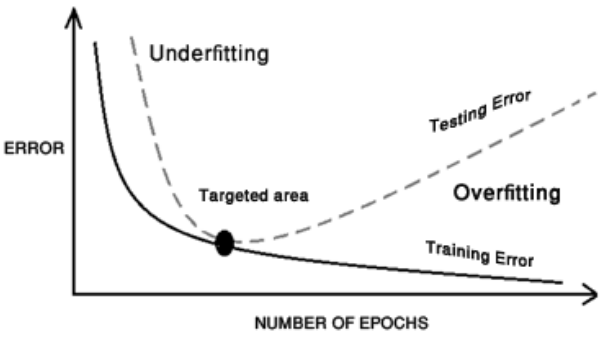


Figure 3. Targeted area to avoid overfitting or underfitting



leave-one-out cross-validations. The issue with the first and second methods is that they are subject to overlap when subjects have multiple related samples. To mitigate this problem, the leave-one-subject-out cross-validation method is appropriate.

2.1.2. Evaluation/Performance Metrics

In addition to the accuracy and loss indicators, to further evaluate the performance of the CNNs, precision, recall, F-measure, and support were computed. Precision instinctively reflects the ability of the model to avoid misclassifying negative samples as positive. This is computed using equation (1):

$$precision = \frac{tp}{\left(tp + fp\right)} \tag{1}$$

where true positives is noted by tp, and false positives is noted by tp.

Recall is the ability to identify all positive samples. This is computed using equation (2):

$$recall = \frac{tp}{\left(tp + fn\right)} \tag{2}$$

where the number of true positives is noted by tp and the number of false negatives is noted by fn.

The F1 score is the harmonic average of recall and precision. This is computed using equation (3):

$$F1score = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = 2 \times \frac{recall \times precision}{recall + precision} = \frac{tp}{tp + \frac{1}{2}\left(fp + fn\right)} \tag{3}$$

Support is the number of occurrences or frequency of each class.

While the macro average is the typical average, the weighted average reflects the number of times each class is involved in its computation; thus, the less the participation of a class, the less impact its precision/recall/F1 scores have on the weighted average.

2.1.3. Precision for Multi-Class Classification

Precision is not limited to binary classification problems, but can be applied to multiclass classification problems, such as this study. The dataset used in this research was balanced (see Figure (4), Table (1)) and the precision was computed using equation (4):

$$Precision = \frac{\sum_{i=1}^{n} TruePositives_i}{\left(\sum_{i=1}^{n} TruePositives_i + \sum_{i=1}^{n} FalsePositives_i\right)} \tag{4}$$

where n is the number of classes.

2.1.4. Recall (Sensitivity) for Multi-Class Classification

Recall is not limited to binary classification problems, it is calculated using equation (5):

$$Recall = \frac{\sum_{i=1}^{n} TruePositives_i}{\left(\sum_{i=1}^{n} TruePositives_i + \sum_{i=1}^{n} FalseNegatives_i\right)} \tag{5}$$

where n is the number of classes.

2.1.5. F-Measure for Imbalanced Classification

Precision and recall are important indicators; however, they can be combined into a single measure that captures both properties. It is the harmonic mean of precision and recall, computed as in Equation (6) (Berger and Guda, 2020); the scores corresponding to every class represent the accuracy of the classifier in classifying the data points in that particular class compared with all other classes, it is calculated using equation (6):

$$Recall = \frac{2 \times Precision \times Recall}{\left(Precision + Recall\right)} \tag{6}$$

2.1.6. Support Value for Multi-Class Classification

The number of actual instances of the class in the provided dataset is known as support. Imbalanced support in the training data may reveal fundamental problems in the classifier's reported scores, necessitating stratified sampling or rebalancing. Support does not change depending on the model, but rather diagnoses the evaluation process.

2.1.7. Macro Averaged F1 Score

Among the several averaging approaches, macro-averaging is the most basic. The arithmetic mean (unweighted mean) of all per-class F1 scores is used to obtain the macro-averaged F1 score (or macro F1 score). Moreover, this method is applied to all classes, regardless of their support values.

2.1.8. Weighted Average

The weighted-average F1 score is generated by taking the mean of all per-class F1 scores and factoring in the support for each class.

2.1.9. Support

The number of actual instances of the class in the dataset is referred to as support. For example, the support value of 1 in the High Class indicates that there is only one observation with the actual label of the High Class.

2.2. Prediction Techniques

Currently, artificial intelligence (AI) is applied across most research disciplines. One of the core objectives of AI is prediction. Various mechanisms have been developed; their computational techniques are briefly described below.

2.2.1. Artificial Neural Network (ANN)

Inspired by the behavior of biological neurons, ANN has been adopted as a computing technique for prediction. A set of layers of artificial neurons, herein referred to as nodes are interconnected in such that the associations between the input and the output are modeled by activating each node and

transmitting its response to the output nodes. The conventional ANN architecture consists of three main types of layers: input, hidden (multiple layers), and output layers of neurons wired together. An activation function was used to determine the state (fire or not to fire) of the neurons. Each layer acts as both an input and an output layer, enabling the ANN to understand more complex objects. A variety of parameters are used to define an ANN, such as the connection mode, learning technique, and activation function (Wang and Srinivasan 2017). Backpropagation is a paramount concept in ANN. It consists of sending backward error information during the supervised training phase to adjust the weights according to their responsibility towards the error. This allows the ANN to adjust the output while considering errors. Therefore, the error due to the disparity between the desired and predicted outcomes is used to recalibrate the weights of the ANN. The aim of this recalibration is to learn how to minimize the chances of errors and undesired outcomes. ANN has been applied to a variety of problems, including image processing, speech recognition, robotic control, and data mining (Abiodun et al., 2018). There is no systematic approach to determine the optimal size of the hidden layers for an ANN; thus, trials and errors during network training are still required to reduce the weight errors as much as possible (Uzair and Jamil, 2020). However, ANN has its own limitations, such as model overfitting, random weight optimization sensitivity, and probability of local optima convergence (Chu et al., 2021, Novak et al. 2018, Zhang et al. 2019).

2.2.2. Deep Belief Network (DBN)

DBNs have been applied successfully in the field of AI, such as image recognition, classification, and filtering. DBN might be seen as an unsupervised network of Restricted Boltzmann machine (RBM), and logistic regression layer (Wang et al. 2016). The first is used to extract features, whereas the logistic regression layer is used for predictions. Furthermore, a DBN might consists of a set of layers of multiple RBMs, where hidden layer for each subnet is the visible layer for the subsequent layers (Wang et al. 2016).

2.2.3. Recurrent Neural Network (RNN)

An RNN is characterized by its connections along time series, which enables temporal dynamic behavior; thus, it might be termed as a dynamic artificial neural network (Ashour, and Abbas, 2018). RNNs use their memory to process input sequences and model temporal dependencies (Dupond, 2019, Abiodun, et al. 2018, Tealab, 2018). They used feedback connections to recall previous time step values as in equation (7):

$$F_{j,i}m = w_i\psi\left(F_{j,i}^{\ m}\right) + w_2\psi\left(F_{j,i}^{\ m-1}\right) \tag{7}$$

where $F_{j,i}^{\ m}$ is the activation function, j represents the output mode, m represents the layer and t represents time. In a Recurrent Neural Network, dependencies are related to a given time steps (Sun et al., 2022). Moreover, various strategies are used in RNNs designs, such as long short-term memory (LSTM) (Yu et al., 2019) and gated recurrent units (GRU) (Chung et al., 2014). Examples of RNNs applications include handwriting recognition (Chen et al. 2020), and speech recognition (Shivakumar et al. 2022, Li, et al. 2014).

2.2.4. Nonlinear Autoregressive Neural Network (NAR and NARX).

Each type of neural network is efficient in some specific areas. The nonlinear autoregressive network with exogenous inputs (NARX) is a variant network of RNNs, which is a recurrent and dynamic network enclosed by several layers with feedback connections, making it' suitable for time-series forecasts (Wang, et al. 2016, Cadenas, et al. 2016). In line with recurrent systems behavior, the NARX model can decrease the output errors with self-learning, thus enhancing the estimation of the ANN.

According to Chan et al. (2015), subsequent NARX values are forecasted based on previous data as in equation (8).

$$\hat{y}(t) = f\left[u(t), u(t-1), \cdots, u(t-nu), y(t-1), \cdots, y(t-ny)\right] + e(t) \tag{8}$$

where y(t) is the target variable; y^(t) is the predicted variable; u(t) is the input variable of the network; n_u and n_y are the time delays of the input and output variables respectively; and e(t) is the model variance between the target and prediction (Liu et al.2020).

2.2.5. Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs), also called ConvNets were introduced by Yann LeCun in the 1990's, one of the main achievements of early versions of CNNs was successfully recognizing handwritten digits. CNNs have found fertile areas for applications such as postal services and banking. However, despite their success in reading zip codes and digits on checks, they could not flourish because of serious issues when scaled. Larger size or high-resolution images have hindered the spread of CNNs, because they require higher computing resources and larger datasets, which were not available at early times. However, in the last couple of decades, the social/mobile Web (Web 2.0), semantic Web (Web 3.0), and upcoming intelligent Web (Web 4.0) have been supporting the creation of a mammoth of content, including images and videos. Coupled with the power of today's computing systems, available datasets such as ImageNet, which includes millions of labeled pictures, could be used to train complex CNN designs for computer vision tasks that were previously unthinkable. Various applications have proven the success of CNNs in many areas, such as natural language processing (NLP), speech recognition, and image classification, including handwritten digit recognition (LeCun et al., 1998), which represents the first successful implementation. Since then, various improvements have been introduced to CNNs' architecture, such as the innovation of new layers and integration of different computer vision techniques (Srivastava et al. 2014). CNNs have provided tremendous opportunities through ImageNet challenges using various datasets (Wang et al. 2022). In the past decade, CNN has been less performant than humans in terms of the detection of image datasets. As claimed by Yang and Hospedales (2015), the human accuracy rate on a typical dataset is 73.1%, whereas the accuracy rate achieved by trained CNNs on the same dataset is limited to 64%. However, after many enhancement and optimization attempts, CNNs outperformed humans (73.1% accuracy rate) on the same datasets, as the CNNs reached a 74.9% accuracy rate. Since then, further improvements have been adopted such as stroke order capturing, innovative feature detectors, descriptor advanced techniques, and optimized and sound designed classification algorithms (Ballester, and Araújo, 2016).

CNN architecture is quite special as it includes:

2.2.5.1. The Convolution Layer

It is the workhorse of the complete network, as it handles the main computational load. A dot product between kernels (i.e., a set of learnable parameters or filters) and the matrix representing the restricted area of the receptive field is used to perform operations such as edge detection, box blur, Gaussian blur, and sharpening. Assuming that WxWxD is the size of the input, and Dout the number of kernels with a special size f with strides (i.e., number of pixels shift), and p is the amount of padding (i.e., number of 0's to add to the image, if the last shifts of the kernels do not fit), then the size of the output volume can be computed as in equation (9) and equation (10):

$$V_{output} = W_{out} \times W_{out} \times D_{out} \tag{9}$$

where

$$W_{out} = \frac{W - f + 2p}{s} + 1 \tag{10}$$

Owing to its effectiveness, the rectified linear unit (ReLU) activation function was adopted for nonlinear operations.

The output is: $f\left(x\right) = \max\left(0, x\right)$ (11)

The transformation performed by this layer involves both activations in the input volume and the parameters (weights and biases) of the neurons.

2.2.5.2. The Pooling Layer

The main role of the pooling layer is to down-sample the images by reducing the dimensionality of each map; however, this requires the preservation of important information. Moreover, different approaches are available such as maximum, average, and sum pooling. For instance, max pooling takes the maximum value from a rectified feature map (Zhao et al. 2017). The main goal of a pooling layer is to minimize the number of parameters in the input tensor, which helps to reduce overfitting, extract representative features from the input tensor, and aid efficiency by reducing computation.

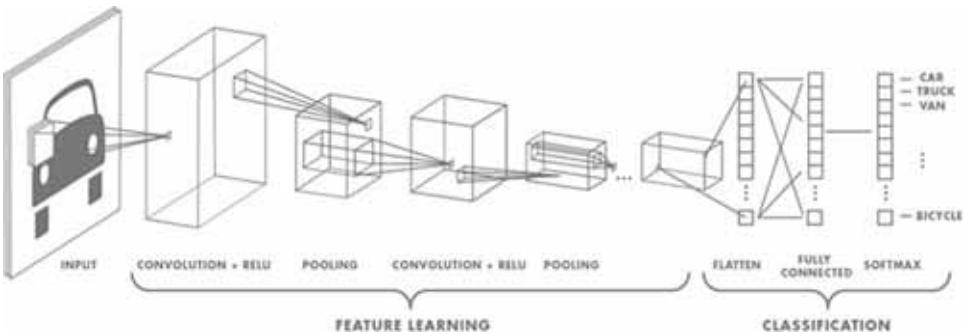
2.2.5.3. Fully Connected (FC) Layer

The main task of this layer is to determine class scores and identify the category of the input image. The structure of this layer is the same as that of a normal Neural Network. The input of the FC is the flattened output from the final pooling or convolutional layers. The transformation conducted by the FC layer involves both activations in the input and the parameters (weights and biases) of the neurons. In contrast, the pooling layers implement a fixed activation function.

The aim of CNNs is to learn abstract features, by assembling and interchanging the pooling and convolution layers (see Figure (4)). The convolution layers convolve the input data through a variety of kernels to transform local features, and the pooling reduces the dimensionality of each map (Zhao, et al. 2017).

In this study, a conventional neural network (CNN) was designed, trained, validated, and used to predict the Parkinson's disease progression category based on the subject's constructed profiles.

Figure 4. Neural network architecture (Prabh, 2018).



3. RESULTS AND DISCUSSION

After the creation of each profile and the generation of the equivalent images in each phase (Phase 1: Data profiles only, Phase 2: Spirals drawings only, Phase 3: Both combined); the data set is subdivided on training, validation, and verification subsets. The designed CNNs with multiple optimizers are trained and validated using the data subsets. Multiple optimizers are used for comparison reasons. The numbers of epochs used are one hundred and two hundreds; however, the accuracy did not change significantly, therefore, the one hundred epochs' case is discussed in details. A small set of profiles' images is reserved for testing the generated models. Certainly, classifiers should be trained and validated by using k-fold cross validation. That is, the training and validation sets are portioned into k subsets of folds $\{s_1,s_2,\ldots,s_k\}$,while the k-1 folds are used to train the classifier, the remaining fold is used to assess the classification performance. Each fold is used for validation for k times when performance metrics are computed; the average of the metrics for each fold is used in computing the global performance. Due to issues such as subjects overlap as mentioned earlier, only training and validation datasets approach is adopted.

A balanced dataset of profiles was generated that included 187 profiles (some classes were balanced by augmentation). Eighty percent of the cleaned and balanced data, that is, 153 profiles, were used for training (figure (5)), and the remaining 34 profiles were split as a set of 30 used for validation, and four were used for verification (Table (1)). The designed CNNs with multiple optimizers (for performance comparison) were trained and validated using these data subsets.

3.1. Selected PPMI Data Profiles Dataset

Accuracy, which is the proportion of correctly predicted observations to the total number of observations, is the most intuitive measure of the performance of CNNs. Higher accuracy does not guarantee that the model is best, only if the datasets used are symmetric, that is, with balanced false-positive and false-negative results. Therefore, other parameters such as precision, recall, and f1-score must be investigated to elucidate the model's performance (see Table (2)). Applying the model to the

Figure 5. Count of training dataset

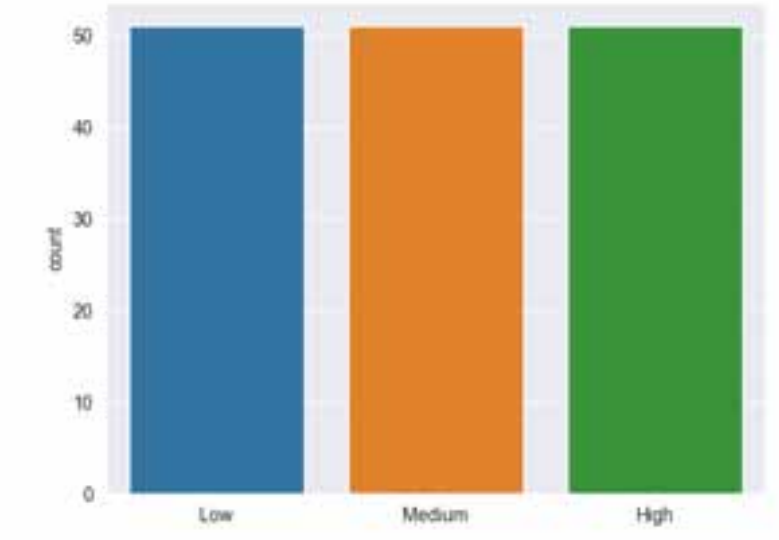


Table 1. Data set distribution

Data distribution			
Label	Training	Validation	Verification
High	51	10	2
Medium	51	10	1
Low	51	10	1

Figure 6. Generated profile from PPMI measures

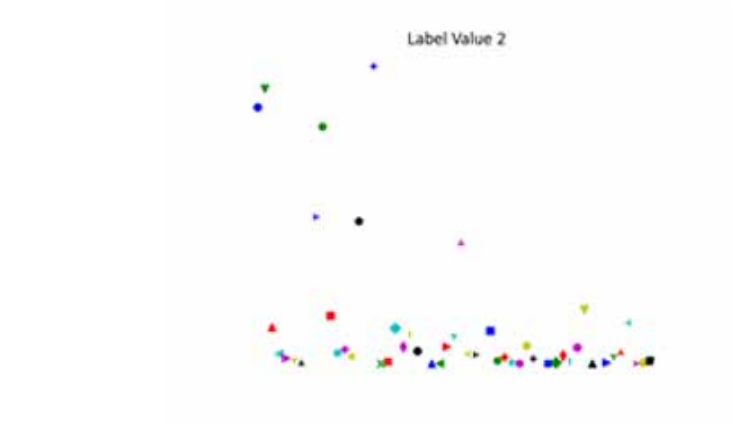


Figure 7. Training and validation loss curve for sgc, rmsprop, adagrad, and adam algorithms

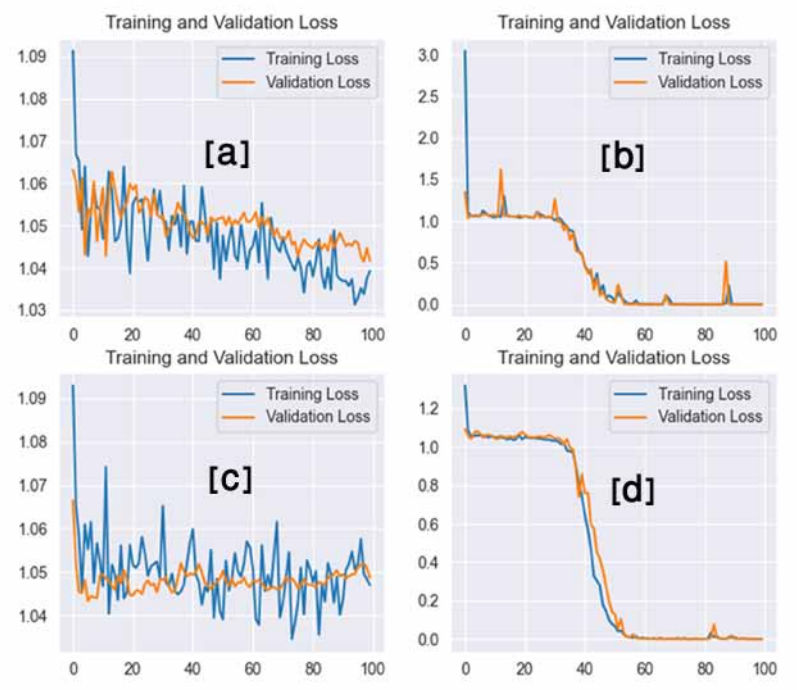


Figure 8. Training and validation accuraccy curve for sgc, rmsprop, adagrad, and adam algorithms

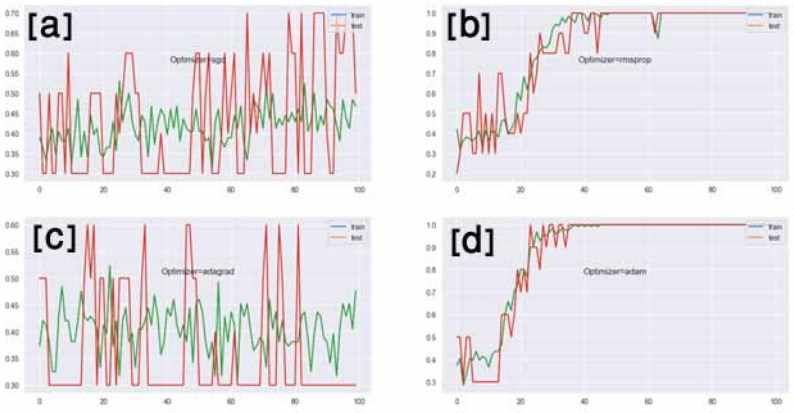


Table 2. Adam optimizer results for PPMI data profiles

Probability: [[8.89760554e-01 1.10230878e-01 8.64232516e-06]
[5.03172696e-01 4.96825606e-01 1.74236004e-06]
[4.44767778e-08 1.04522645e-04 9.99895453e-01]
[6.20392839e-08 8.53807308e-07 9.99999046e-01]
[2.09857058e-06 8.47504707e-05 9.99913096e-01]]
To predict y set: [0 1 2 2 2]
Predicted Classes for y: [0 0 2 2 2]
precision recall f1-score support
Class Low 0.50 1.00 0.67 1
Class Medium 0.00 0.00 0.00 1
Class High 1.00 1.00 1.00 3
accuracy 0.80 5
macro avg 0.00 0.00 0.00 1
weighted avg 0.70 0.80 0.73 5

images generated from the profiles selected from the PPMI dataset achieved an accuracy of 80%, and the precision for class Low was 0.5, which is barely acceptable; however, it is unacceptable for the class Medium as the precision value is 0.00. Moreover, the precision for class High is 1.0, which is perfect for this class, because the higher the precision, the lower the false positive rate. Furthermore, in this section of the article, the recall values 1.00, 0.00, and 1.00 suggest that the first recall value of class Low and the third recall value for class High are very satisfactory; however, the second value of 0 is unsatisfactory for class Medium, as it is less than 0.5. The F1 score is not as easy to understand as accuracy when it has an uneven class distribution. In this paper, the author has been cautious about this issue; thus, the data were balanced by using selectivity and augmentation. F1 scores are 0.67, 0.00, and 1.00 for Class Low, Class Medium, and Class High, which means relatively acceptable for the first class, and perfect for the last class; however, for Class Medium the F1score is 0, because both precision and recall are 0s.

3.2. Profile Using Spirals Drawings Dataset

As mentioned earlier, a higher accuracy does not systematically mean that the model is best; therefore, other parameters such as precision, recall, and f1-score (Table (3)) must be investigated to shed light on the model's performance. Applying the model to the spiral drawing dataset led to an accuracy of 83%. In this section of the article, the precision for class Low is 1.00, which is a perfect precision;

Figure 9. Spirals drawings for one subject

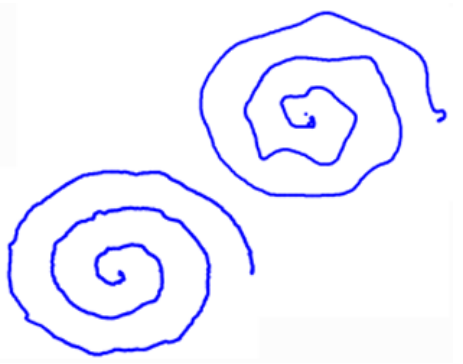
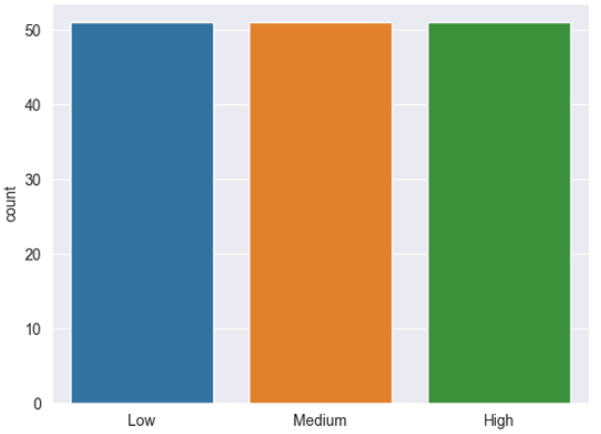


Figure 10. Count of training dataset



however, it is acceptable for the class medium as the precision value is 0.67, and the precision for class High is 1.0, which is perfect for this class, as the higher the precision, the lower the false positive rate. Furthermore, the recall values of 1.00, 1.00, and 0.50 suggested that the first recall value of class Low and the second recall value of class Medium are perfect; however, the third recall value of 0.50 is barely acceptable for class High. F1 scores are 1.00, 0.80, and 0.67 for Class Low, Class Medium, and Class High, which is perfect for the first class, and very satisfactory for the second class. For Class High, the F1score is 0.67, which is acceptable as it is greater than 0.5.

3.3. Profiles Combining Numerical Data and Spirals

The analysis of the generated figures and collected data concerning all phases suggests that the Adam optimizer is the best performing algorithm of the tested set of optimizers (Figures (7)-(17); Tables (4)-(7)). Thus, the focus is on the results produced by the model using the adam optimizer.

Despites the slight ups and downs, and the loss decreases over time, the model keeps learning, as shown in the figure of training and validation loss for the adam optimizer (Figure (14)(d), figure (15)(d)). As the author observes (Figure (16), Figure (17)), 100 epochs is sufficient for the adam optimizer, therefore, the execution of model with 100 epochs was adopted in this study.

Figure 11. Training and validation loss curve for sgc, rmsprop, adagrad, and adam algorithm

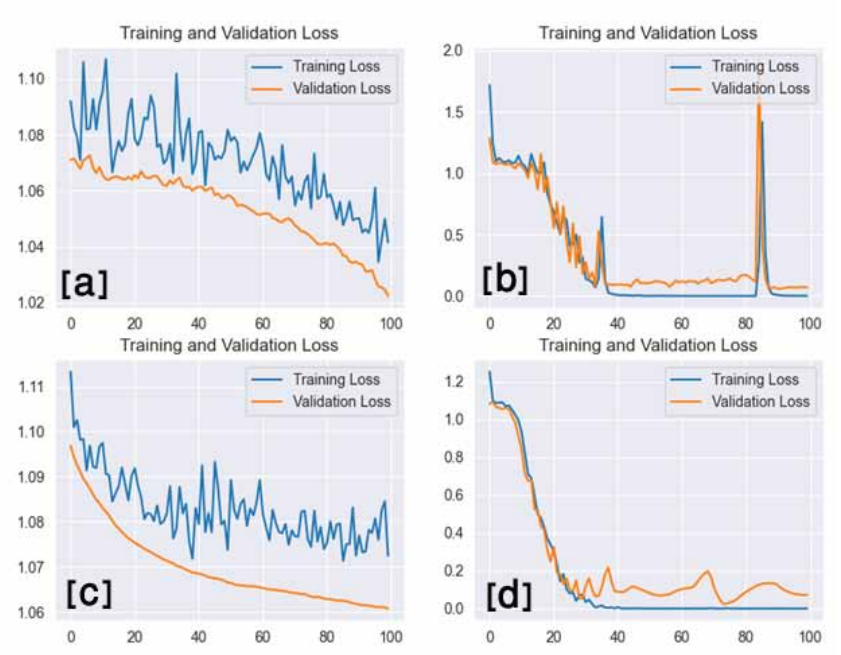
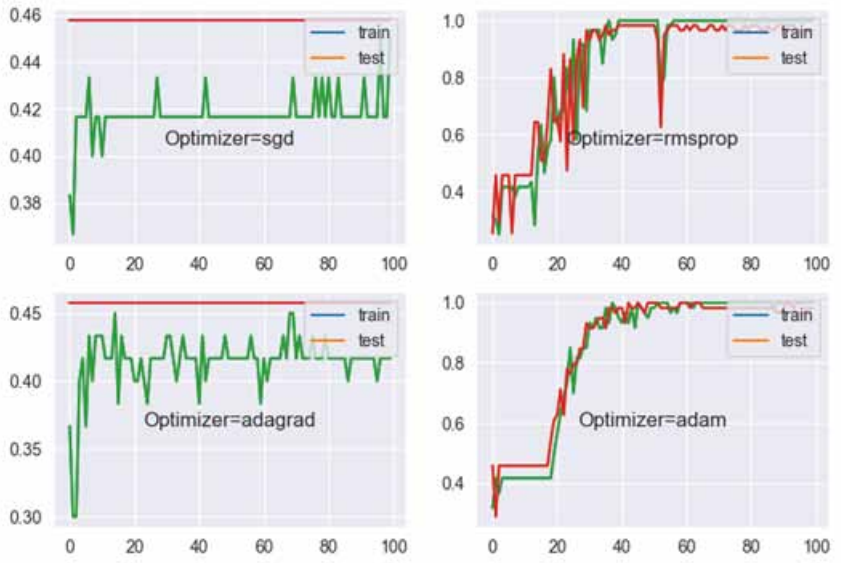


Figure 12. Training and validation accuraccy curve for sgc, rmsprop, adagrad, and adam algorithms



In line with the training and validation loss behavior using the adam optimizer, the model performance increases as the number of iterations progresses, which means that the model is improving with experience, that is, it is learning. However, it is clear that before 100 epochs, the model reaches a plateau, meaning it is no longer able to learn. This validates the earlier observations concerning the training and validation losses in figure (15 adam) and figure (16d).

Table 3. adam optimizer results for spirals drawings

Probability: [[9.9998939e-01 5.2209281e-10 1.0561393e-05] [9.9998939e-01 5.2209281e-10 1.0561393e-05] [4.3724486e-07 9.9985337e-01 1.4608701e-04] [4.3724486e-07 9.9985337e-01 1.4608701e-04] [1.6452427e-08 7.6973373e-03 9.9230266e-01] [5.7641308e-05 9.3937582e-01 6.0566530e-02]] To predict y set: [0 0 1 1 2 2] Predicted Classes for y: [0 0 1 1 2 1] precision recall f1-score support Class Low 1.00 1.00 1.00 2 Class Medium 0.67 1.00 0.80 2 Class High 1.00 0.50 0.67 2 accuracy 0.83 6 macro avg 0.89 0.83 0.82 6 weighted avg 0.89 0.83 0.82 6

Figure 13. Progression profiles samples(Combined)

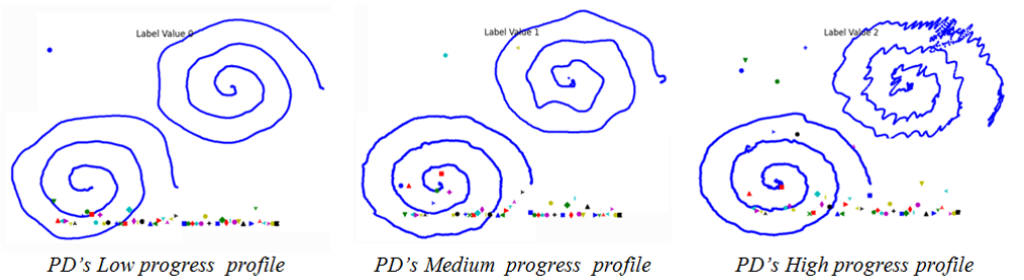


Figure 14. Training and validation loss curve for sgc, rmsprop, adagrad, and adam algorithms (200 Epochs)

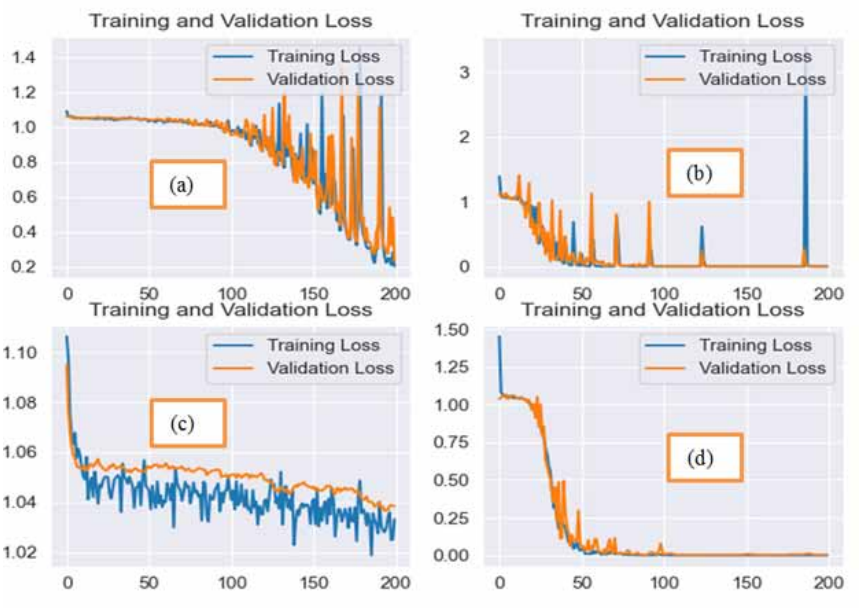


Figure 15. Training and validation accuracy curve for sgc, rmsprop, adagrad, and adam algorithms (200 Epochs)

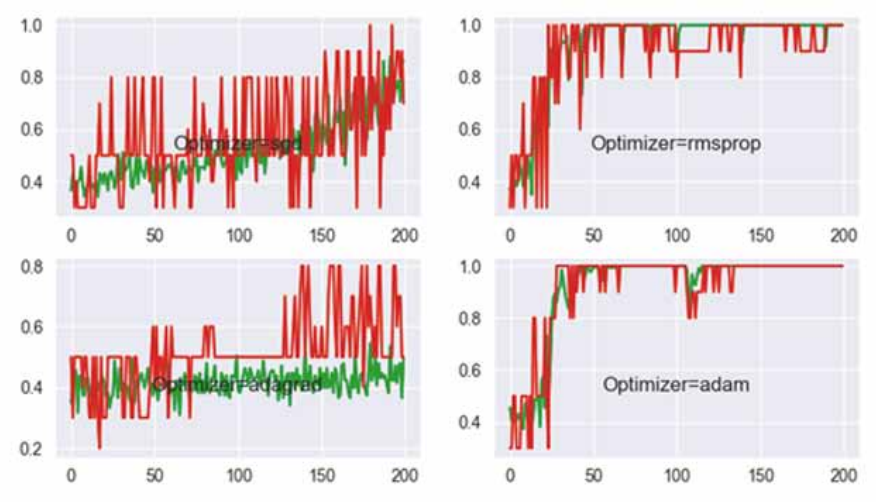
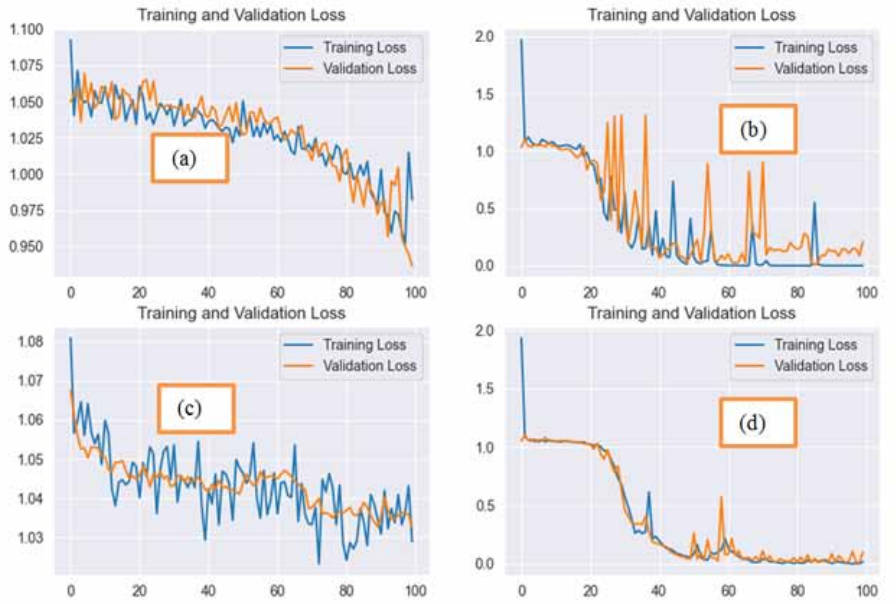


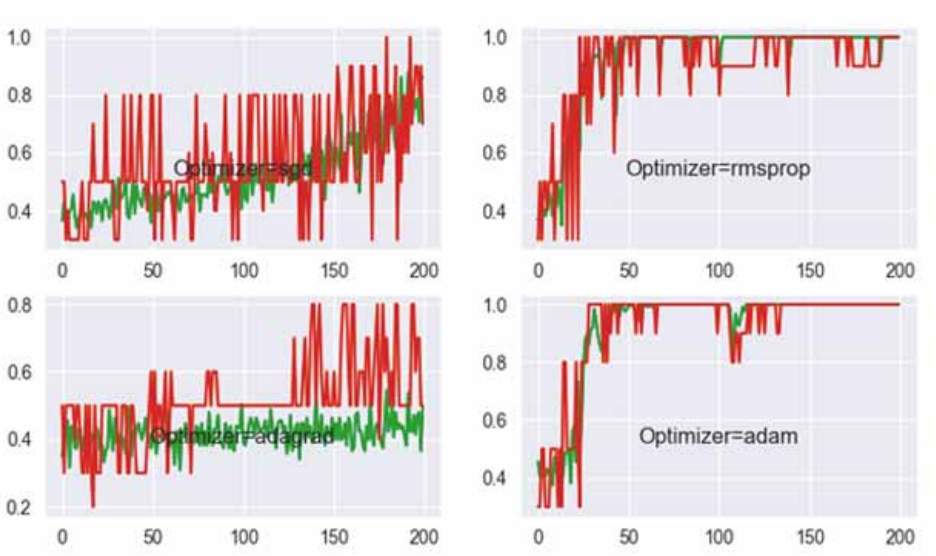
Figure 16. Training and validation loss curve for sgc, rmsprop, adagrad, and adam algorithms (200 Epochs)



Likewise, in figure (17) for the adam optimizer, the model performance (i.e., accuracy) improves as the number of iterations progresses, which means that the model is learning with experience. However, it is clear that a 75 epochs might be sufficient for this model, because after this epoch it is no longer learning.

After execution, the CNN produces a probability distribution of possible input classes. Accordingly, the most probable class was selected, as shown in Tables (4)-(7). To predict the class of the first input (i.e., the image of a profile), the model computes the probabilities suggesting the target class. The generated image of the first profile is classified in the Parkinson's low progression

Figure 17. Training and validation accuracy curve for sgc, rmsprop, adagrad, and adam algorithms (100 Epochs)



category with a probability value of 9.9998236e-01 for class 0, the probability value of 1.7691382e-05 for class 1, and the probability value of 1.5363933e-14 for class 2. Furthermore, the precision is 1.0, 1.0, and 1.0, for the low, medium, and high classes, respectively, which means that the inputs classified as a class low, a class medium, or a class high, are all (100%) perfectly placed where they really should be. Moreover, the macro-average precision is 1.0, which is the same as the weighted average. The weighted average is higher across all classes for this model because the precision did not decrease for all classes, as all were equally represented in the weighted average.

In this study, the author evaluated the progression of Parkinson's disease by converting indicators' measures and spirals drawings to an image that reflects the progression of the disease in a more complex manner that neither the indicators' measures alone nor the drawings alone might reflect. In other words, CNN tools offer an opportunity to detect hidden patterns in a profile (Figure 13) that might not be detected otherwise. As the results show, this novel approach increased the accuracy to 100% for the dataset under study, which is an important achievement. Subsequently, the accuracy has improved from 80% for the numerical data profile images and from 83% for spirals drawings to 100% when both combining to a single profile.

4. LIMITATIONS AND RECOMMENDATIONS

It is clear that the size of the dataset is somewhat limited, because of the exclusion of subjects that do not have the required data to build their profiles; however, data augmentation used in this research has improved the performance of the generated CNN models. Future research should evaluate the performance of such models using different datasets.

5. CONCLUSION

Certainly, the fact that PD is the second most frequent neurological syndrome of the central nervous system' deserves substantial attention from researchers of different fields. Currently, there is no cure for this devastating neurological disease, which is a dilemma. Consequently, tracking its progression is

a major step towards symptoms management and medication protocol adaptation. This research paper has emerged from the need to mitigate this dilemma. AI solutions have flourished in many scientific and engineering fields. Therefore, a CNN is designed and used to categorize PD patients according the disease progression rate. The novel idea of combining numerical measurements of a set of indicators of the disease and spiral drawings of patients in a periodic manner has been successful in gaging its progression. The constructed graphical profiles helped to reach a very high accuracy (100%) when classifying the progression of PD using the designed CNN. The diversity of the measured indicators has been a major contributor to the convergence of an identifying disease progression profile. Neither the profiles without spiral drawings (accuracy 80%) nor the spirals drawings by themselves (accuracy 83%) reached the high accuracy rate of (100%) achieved by the same model when both profiles are combined by superimposition. Thus, the proposed novel approach is a major contribution to the evaluation of PD's progression. However, the approach might be scaled to a more granular scale than the used one with the limited three classes (high speed, medium speed, low speed). A five-point Likert scale (very slow, slow, average, fast, very fast) or 7 Likert scale (extremely slow, very slow, slow, average, fast, very fast, extremely fast) could help capture a more accurate and refined evaluation of PD progression.

REFERENCES

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Nachaat, A. M., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon, 4*(11). 10.1016/j.heliyon.2018.e00938

Al-Fatlawi, A. H., Jabardi, M. H., & Ling, S. H. (2016). Efficient diagnosis system for Parkinson s disease using deep belief network. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE.

Asgari, S., Khalili, D., Zayeri, F., Azizi, F., & Hadaegh, F. (2021). Dynamic prediction models improved the risk classification of type 2 diabetes compared with classical static models. *Journal of Clinical Epidemiology, 140*, 33-43. 10.1016/j.jclinepi.2021.08.026

Ashour, M. A. H., & Abbas, R. A. (2018). Improving Time Series Forecast Errors by Using Recurrent Neural Networks. In *Proceedings of the 2018 7th International Conference on Software and Computer Applications*. ACM.

Ballester, P., & Araújo, R. M. (2016). On the Performance of GoogLeNet and AlexNet Applied to Sketches. *AAAI*.

Benba, A., Jilbab, A., & Hammouch, A. (2016a). Analysis of multiple types of voice recordings in cepstral domain Using MFCC for discriminating between patients with Parkinson s disease and healthy people. *International Journal of Speech Technology*, *19*(3), 449–456.

Benba, A., Jilbab, A., & Hammouch, A. (2016b). Voice assessments for detecting patients with Parkinson s diseases using PCA and NPCA. *International Journal of Speech Technology*, *19*(4), 743–754.

Benba, A., Jilbab, A., & Hammouch, A. (2017). Using human factor cepstral coefficient on multiple types of voice recordings for detecting patients with Parkinson s disease. *IRBM*, *38*(6), 346–351.

Berger, A., & Guda, S. (2020). Threshold optimization for F measure of macro-averaged precision and recall. *Pattern Recognition, 102*. 10.1016/j.patcog.2020.107250

Cadenas, E., Rivera, W., Campos-Amezcua, R., & Cadenas, R. (2016). Wind speed forecasting using the NARX model, case: La Mata, Oaxaca, México. *Neural Computing & Applications*, *27*(8), 2417–2428. https://doi.org/10.1007/s00521-015-2012-y

Canturk, I., & Karabiber, F. (2016). A machine learning system for the diagnosis of Parkinson s disease from speech signals and its application to multiple speech signal types. *Arabian Journal for Science and Engineering*, *41*(12), 5049–5059.

Chan, R. W. K., Yuen, J. K. K., Lee, E. W. M., & Arashpour, M. (2015). Application of Non-linear- Autoregressive-Exogenous model to predict the hysteretic behavior of passive control systems. *Engineering Structures*, *85*, 1–10. doi:10.1016/j.engstruct.2014.12.007

Chen, Z., Yin, F., Zhang, X. Y., Yang, Q., & Liu, C. L. (2020). MuLTReNets: Multilingual text recognition networks for simultaneous script identification and handwriting recognition. *Pattern Recognition, 108*. 10.1016/j.patcog.2020.107555

Chu, J., Liu, X., Zhang, Z., Zhang, Y., & He, M. (2021). A novel method overcoming overfitting of artificial neural network for accurate prediction: Application on thermophysical property of natural gas, Case Studies. *Thermal Engineering, 28*. 10.1016/j.csite.2021.101406

Diaz, K., Kohut, M. L., Russell, D. W., & Stegemöller, E. L. (2022). Peripheral inflammatory cytokines and motor symptoms in persons with Parkinson s disease, Brain, Behavior, & Immunity. *Health, 21*. 10.1016/j.bbih.2022.100442

Duffy J. R. (2013). Motor Speech Disorders: Substrates, Differential Diagnosis, and Management. *Elsevier Health Sciences*.

Dupond, S. (2019). A thorough review on the current advance of neural network structures. *Annual Reviews in Control*, *14*, 200–230.

Fereshtehnejad, S. M., Romenets, S. R., Anang, J. B., Latreille, V., Gagnon, J. F., & Postuma, R. B. (2015). New clinical subtypes of Parkinson disease and their longitudinal progression: A prospective cohort comparison with other phenotypes. *JAMA Neurology*, *2015*(72), 863–873.

Fernández-García, S., Dumitrache, C. G., & González-López, J. A. (2021). Acoustic analysis of the voice in patients with Parkinson s disease and hypokinetic dysarthria. *Revista de Logopedia, Foniatría y Audiología, 41*(3), 142-150. 10.1016/j.rlfa.2020.04.002

Fioravanti, V., Benuzzi, F., Codeluppi, L., Contardi, S., Cavallieri, F., Nichelli, P., & Valzania, F. (2015). MRI correlates of Parkinson s disease progression: A voxel-based morphometry study. *Parkinson's Disease*, 1–8. doi:10.1155/2015/378032

Guruler, H. (2017). A novel diagnosis system for Parkinson s disease using ¨complex-valued artificial neural network with k-means clustering feature weighting method. *Neural Computing & Applications*, *28*(7), 1657–1666.

Heim, B., Krismer, F., De Marzi, R., & Sppi, K. (2017). Magnetic resonance imaging for the diagnosis of Parkinson s Disease. *Journal of Neural Transmission (Vienna, Austria)*, *124*(8), 915–964. doi:10.1007/s00702-017-1717-8

Kouli, A., Torsney, M. K., & Kuan, W. (2018). Parkinson's Disease: Etiology, Neuropathology, and Pathogenesis. In *Parkinson s Disease Pathogenesis and Clinical Aspects. Codon Publications*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

Li, X., & Wu, X. (2014). *Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition*. arXiv:1410.4281 [cs.CL].

Li, Y., Zhang, C., Jia, Y., Wang, P., Zhang, X., & Xie, T. (2017). Simultaneous learning of speech feature and segment for classification of Parkinson disease. In *e-Health Networking, Applications and Services (Healthcom), IEEE 19th International Conference on*. IEEE.

Liaqat, A., Zhu, C., Golilarz, N. A., Javeed, A., Zhou, M., & Liu, Y. (2019b). 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: Practical Innovations, Open Solutions*, 1–1.

Liaqat, A., Zhu, C., Zhou, M., & Liu, Y. (2019a). Early diagnosis of Parkinson s disease from multiple voice recordings by simultaneous sample and feature selection. *Expert Systems with Applications*, *137*, 22–28.

Little, M. A., McSharry, P. E., Hunter, E. J., Spielman, J., & Ramig, L. O. (2009). Suitability of dysphonia measurements for telemonitoring of Parkinson s disease. *IEEE Transactions on Biomedical Engineering*, *56*(4), 1015–1022.

Liu, Q., Chen, W., Hu, H., Zhu, Q., & Xie, Z. (2020). An Optimal NARX Neural Network Identification Model for a Magnetorheological Damper With Force-Distortion Behavior. *Frontier in Materials*. 10.3389/fmats.2020.00010

López, J. L., Hernández, S., Urrutia, A., López-Cortés, X. A., Araya, H., & Morales-Salinas, L. (2021). Effect of missing data on short time series and their application in the characterization of surface temperature by detrended fluctuation analysis. *Computers & Geosciences, 153*. 10.1016/j.cageo.2021.104794

Louis, E.D., Mayer, S.A., & Rowland, L.P. (2015). *Merritt s Neurology* (13th ed.). Lippincott Williams & Wilkins.

Lu, M., Zhao, Q., Poston, K. L., Sullivan, E. V., Pfefferbaum, A., Shahid, M., Katz, M., Kouhsari, L. M., Schulman, K., Milstein, A., Niebles, J. C., Henderson, W., Fei-Fei, L., Pohl, K. M., & Adeli, E. (2021). Quantifying Parkinson s disease motor severity under uncertainty using MDS-UPDRS videos. *Medical Image Analysis, 73*. 10.1016/j.media.2021.102179

Maleki,, F., & Muthukrishnan,, N., Ovens, K., Reinhold, C., & Forghani, R. (2020). Machine Learning Algorithm Validation. *Neuroimaging Clinics of North America*, *30*(4), 433–445.

Naranjo, L., Perez, C. J., Campos-Roca, Y., & Martın, J. (2016). Addressing voice recording replications for Parkinson s disease detection. *Expert Systems with Applications*, *46*, 286–292.

Naranjo, L., Perez, C. J., & Martin, J. (2017a). Addressing voice recording replications for tracking Parkinson s disease progression. *Medical & Biological Engineering & Computing*, *55*(3), 365–373.

Naranjo, L., Perez, C. J., Martın, J., & Campos-Roca, Y. (2017b). A two-stage variable selection and classification approach for Parkinson s disease detection by using voice recording replications. *Computer Methods and Programs in Biomedicine*, *142*, 147–156.

Novak, R., Bahri, Y., Abolafia, A. D., Pennington, J., & Sohl-Dickstein, J. (2018). Sensitivity and Generalization in Neural Networks: an Empirical Study. In 6th ICLR 2018, Vancouver, Canada.

Oludare, I. A., Jantan, A., Omolara, A.E., Dada, K.V., Nachaat AbdElatif, M., & Humaira, A. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon, 4*(11). doi:10.1016/j.heliyon.2018.e00938

Orozco-Arroyave, J., Honig, F., Arias-Londono, J., & Vargas-Bonilla, J. (2016). Automatic detection of Parkinson s disease in running speech spoken in three different languages. *The Journal of the Acoustical Society of America*, *139*(1), 481–500.

Parisi, L. (2018). Feature-driven machine learning to improve early diagnosis of Parkinson's disease. *Expert Systems with Applications*, *110*, 182–190.

Pasyar, P., Mahmoudi, T., Kouzehkanan, S. Z. M., Ahmadian, A., Arabalibeik, H., Soltanian, N., & Radmard, A. R. (2021). Hybrid classification of diffuse liver diseases in ultrasound images using deep convolutional neural networks. *Informatics in Medicine Unlocked, 22*. 10.1016/j.imu.2020.100496

Prabh, R. (2018). *Understanding of Convolutional Neural Network (CNN).* Deep Learning. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f

Pyatigorskaya, N., Magnin, B., Mongin, M., Yahia-Cherif, L., Valabregue, R., Arnaldi, D., Ewenczyk, C., Poupon, C., Vidailhet, M., & Lehéricy, S. (2018). Comparative study of MRI biomarkers in the substantia nigra to discriminate idiopathic Parkinson disease. *AJNR. American Journal of Neuroradiology*, *39*(8), 1460–1467. doi:10.3174/ajnr.A5702

Raket, L. L., Oudin Åström, D., Norlin, J. M., Kellerborg, K., Martinez-Martin, P., & Odin, P. (2022). Impact of age at onset on symptom profiles, treatment characteristics and health-related quality of life in Parkinson s disease. *Scientific Reports*, *12*, 526. https://doi.org/10.1038/s41598-021-04356-8

Rigas, G., Tzallas, A. T., Tsipouras, M. G., Bougia, P., Tripoliti, E. E., Baga, D., Fotiadis, D. I., Tsouli, S. G., & Konitsiotis, S. (2012). Assessment Of tremor activity in the Parkinson s disease using a set of wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, *16*(3), 478–487.

Roede, , Uppal, , Park, , Lee, , & Tran, , Walker, Strobel, Rhodes, Beate, & Jones. (2013). Serum Metabolomics of Slow vs. Rapid Motor Progression Parkinson's Disease: A Pilot Study. *PLoS One*, *8*(10), e77629.

Sakar, B. E., Isenkul, M. E., Sakar, C. O., Sertbas, A., Gurgen, F., Delil, S., Apaydin, H., & Kursun, O. (2013). Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings. *IEEE Journal of Biomedical and Health Informatics*, *17*(4), 828–834.

Schrag, A., Siddiqui, U. F., Anastasiou, Z., Weintraub, D., & Schott, M. J. (2017). Clinical variables and biomarkers in prediction of cognitive impairment in patients with newly diagnosed Parkinson s disease: a cohort study. *The Lancet Neurology, 16*(1), 66-75.

Shivakumar, P. G., & Narayanan, S. (2022). End-to-end neural systems for automatic children speech recognition: An empirical study. *Computer Speech & Language, 72*. 10.1016/j.csl.2021.101289

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sterling, N. W., Lewis, M. M., Du, G., & Huang, X. (2016). Structural Imaging and Parkinson s disease: Moving Toward Quantitative Markers of Disease Progression. *Journal of Parkinson's Disease*, *6*(3), 557–567.

Sun, C., Zhang, Y., Huang, G., Liu, L., & Hao, X. (2022). A soft sensor model based on long & short-term memory dual pathways convolutional gated recurrent unit network for predicting cement specific surface area. *ISA Transactions*. 10.1016/j.isatra.2022.03.013

Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal, 3*(2), 334–340. doi:10.1016/j.fcij.2018.10.003

Tolosa, E., Garrido, A., Scholz, W. S., & Poewe, W. (2021). Challenges in the diagnosis of Parkinson s disease. *The Lancet Neurology, 20*(5), 385-397. 10.1016/S1474-4422(21)00030-2

Tsanas, A., Little, M. A., McSharry, P. E., Spielman, J., & Ramig, L. O. (2012). Novel speech signal processing algorithms for high-accuracy classification of Parkinson s disease. *IEEE Transactions on Biomedical Engineering*, *59*(5), 1264–1271.

Uzair, M., & Jamil, N. (2020). Effects of Hidden Layers on the Efficiency of Neural networks. *2020 IEEE 23rd International Multitopic Conference (INMIC),* 1-6. doi: doi:10.1109/INMIC50486.2020.9318195

Vasquez-Correa, J. C., Arias-Vergara, T., Orozco-Arroyave, J., Eskofier, B. M., Klucken, J., & Noth, E. (2018). Multimodal assessment of Parkinson s disease: A deep learning approach. *IEEE Journal of Biomedical and Health Informatics*.

Wang, H. Z., Wang, G. B., Li, G. Q., Peng, J. C., & Liu, Y. T. (2016). Deep belief network based deterministic and probabilistic wind speed forecasting approach. *Applied Energy, 182*, 80-93. 10.1016/j.apenergy.2016.08.108

Wang, N., Wang, Y., & Er, M. J. (2022). Review on deep learning techniques for marine object recognition: Architectures and algorithms. *Control Engineering Practice, 118*. 10.1016/j.conengprac.2020.104458

Wang, Z., & Srinivasan, R. S. (2017). A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. *Renewable & Sustainable Energy Reviews*, *75*, 796–808.

Yang, Y., & Hospedales, T. M. (2015). *Deep neural networks for sketch recognition*. CoRR abs/1501.07873.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTMcells and network Architectures. *Neural Computation*, *31*, 1235–1270. https://dx.doi.org/10.1162/neco_a_01199

Zhang, C., Ding, S., & Du, W. (2022). Broad stochastic configuration network for regression. *Knowledge-Based Systems, 243*. 10.1016/j.knosys.2022.108403

Zhang, G., Martens, J., & Grosse, R. B. (2019). Fast Convergence of Natural Gradient Descent for Over-Parameterized Neural Networks. Advances in Neural Information Processing Systems, 32.

Zhao, R., Yan, R., Wang, J., & Mao, K. (2017). Learning to Monitor Machine Health with convolutional Bi-Directional LSTM Networks. *Sensors (Basel)*, *17*(2), 273. https://doi.org/10.3390/s17020273

APPENDICES

Appendix Table A. Brief data description and computing methods.

Measure	Descriptive	Calculation method
Walk		
SP_U SP_DT	Speed base walking and Dual task walking (m/sec)	Calculated from the 10m walking time. Same procedure for usual walking and dual task walking
RA_AMP_U RA_AMP_DT	Right arm amplitude base and Dual task walking (deg)	Average Amplitude of the Right arm presented in degrees. Calculated from the right wrist accelerometers as the range from peak anterior movement to posterior movement (rotational movement included into calculation based on Euler angles)
LA_AMP_U LA_AMP_DT	Left arm amplitude base and Dual task walking (deg)	Average Amplitude of the Left arm degree (same as above for the left)
RA_STD_U RA_STD_DT	Right arm variability (standard deviation) in base walking and Right arm standard deviation in Dual task walking (%)	Standard deviation of the average amplitude of the Right arms presented in degrees. Calculated from the right wrist accelerometers as the range from peak anterior movement to posterior movement (rotational movement included into calculation based on Euler angles)
LA_STD_U LA_STD_DT	Left arm standard deviation in base walking Left arm standard deviation in Dual task walking (%)	Standard deviation of the Left arm degree (same as above)
SYM_U SYM_DT	Between arm symmetry base walking Between arm symmetry Dual task walking (%)	

Appendix Table B

Measure	Descriptive	Calculation method
Walk		
		$= \left\| \frac{45^{\circ} - \arctan(\frac{GreaterHandAmplitude}{SmallerHandAmplitude}) * 100\%}{90} \right\|$ The ASA is designed to represent asymmetry in arm swing magnitude between each arm. A value of 0.00 would indicate that both arms are moving exactly the same magnitude.
ASYM _IND_U ASYM _IND_DT	Asymmetry index calculated from arms in base and dual task walking (%)	Asymmetry index $= \left\| \frac{Av_{AmplitudeL} - AV\_Amplitude_R}{Av_{AmplitudeL} + AV\_Amplitude_R} \right\| * 100\%$:
TRA_U TRA_DT	Trunk Rotation asymmetry index in base walking and dual task (%)	Calculated from the acceleration signal from the lower back sensor, we quantified trunk rotation to the left and to the right as thetransverse plane angular rotation of the thorax. The magnitude of trunk rotation was quantified as the total side--to--side rotation of the thorax during a stride cycle. We then calculated the trunk rotation asymmetry (TRA) as follows TRA $= \left\| \frac{45^{\circ} - \arctan(\frac{Trunk_{more}magnitude}{Trunk_{less}magnitude}) * 100\%}{90} \right\|$
T_AMP_U T_AMP_DT	Average trunk amplitude of rotational movement in base and Dual task walking (deg)	Average Amplitude of rotational movement around the vertical axis of the trunk in degrees. Calculated from the Euler angles from the lower back sensor
CAD_U CAD_DT	Cadence in base and Dual task walking (step/min)	Number of gait cycles taken within one minute of walking. Calculated from the acceleration signal from the lower back sensor
STR_T_U STR_T_DT	Average stride time in base and dual task walking (sec)	Stride time average during straight-line walking (turns excluded). Calculated from the vertical acceleration signal from the lower back sensor.

Appendix Table C

Measure	Descriptive	Calculation method
Walk		
STR_CV_U STR_CV_DT	Stride variability (coefficient of variation-CV) in base and Dual task walking (%)	Average Stride CV from straight-line walking (turns excluded). Calculated from the vertical acceleration signal from the lower back sensor.
STEP_REG_U STEP_REG_DT	Step Regularity in base and Dual task walking (g^2)	Derived from the autocorrelation signal of the vertical acceleration from the lower back sensor. Steps of both legs are averaged to provide one measure. Perfect regularity (i.e., no variability) result in correlation coefficients of 1.
STEP_SYM_U STEP_SYM_DT	Step Symmetry in base and dual task walking	Step symmetry were derived from frequency analysis of vertical acceleration signals using autocorrelation.
JERK_T_U JERK_T_DT	Jerk of the acceleration movement of the legs during base and dual task walking (m/sec^3)	Jerk calculated as the time- derivative of trunk acceleration, to quantify smoothness.
SW_VEL_OP SW_VEL_CL	Sway velocity (m/sec) in Eyes open and Eyes closed conditions (30 sec each)	Mean velocity is obtained through the derivative of the displacement the anterior-posterior (AP) and medial- lateral (ML) acceleration of the sensor on the lower back
SW_PATH_OP SW_PATH_CL	Sway path (m/sec^2) in Eyes open and Eyes closed conditions (30 sec each)	Total length of CoP (Center of pressure)) trajectory, calculated from the anterior-posterior (AP) and medial- lateral (ML) acceleration of the sensor on the lower back

Appendix Table D

Measure	Descriptive	Calculation method
Walk		
SW_FREQ_OP SW_PATH_CL	Centroidal frequency (HZ) in Eyes open and Eyes closed conditions (30 sec each)	The frequency at which the spectral mass is concentrated. Calculated from the anteriorposterior (AP) acceleration using Fast Fourier Transformation (FFT).
SW_JERK_OP SW_JERK_CL	Jerk sway (m/sec^3) in Eyes open and Eyes closed conditions (30 sec each)	A measure of Sway smoothness. Calculated as the time-derivative of the anterior-posterior (AP) and medial- lateral (ML) acceleration from the sensor on the lower back.
TUG1_DUR TUG2_DUR	TUG 1 and 2 duration (sec)	The time from start of sit-to-stand (the minimum AP acceleration peak before the signal started to rise from steady state), until end of stand-to-sit (the minimum acceleration peak when the AP acceleration reached steady state).
TUG1_STEP_NUM TUG2_STEP_NUM	TUG 1 and 2 number of step (#number)	Number of steps throughout the duration of the task including turns.
TUG1_STRAIGHT_DUR TUG2_STRAIGHT_DUR	TUG 1 and 2 Average step duration during straight walking (sec)	Average time initial gait step (detected from back sensor vertical acceleration) until 1st turn (detected from back sensor gyroscope –yaw) + duration from end of 1st turn until beginning of turn to sit
TUG1_TURNS_DUR TUG2_TURNS_DUR	TUG 1 and 2 Average step duration during turns (sec)	Turning measures were derived from the yaw axis of the back gyroscope. The start and end points of each turn were determined as the points in the yaw signal in which it crossed 0.1 of the maximum yaw peak amplitude of the turn. Average step duration for these segments is calculated from the vertical acceleration signal
TUG1_STEP_REG TUG2_STEP_REG	TUG 1 and 2 Step regularity (g^2)	Step regularity was derived from the autocorrelation signal of the vertical acceleration.
TUG1_STEP_SYM TUG2_STEP_SYM	TUG 1 and 2 step symmetry	Step symmetry= $\frac{stepregularity}{strideregularity}$

Appendix CNN Model Architecture Code

```
model = Sequential()
model.add(Conv2D(32, 3, padding="same", activation="relu", input_
shape=(224, 224, 3)))
model.add(MaxPool2D())
model.add(Conv2D(32, 3, padding="same", activation="relu"))
model.add(MaxPool2D())
model.add(Conv2D(64, 3, padding="same", activation="relu"))
model.add(MaxPool2D())
model.add(Conv2D(64, 3, padding="same", activation="relu"))
model.add(MaxPool2D())
model.add(Dropout(0.4))
model.add(Flatten())
model.add(Dense(128, activation="relu"))
model.add(Dense(3, activation="softmax"))
```

Appendix Structure of the CNN Network

Figure 18. Flowchart of the CNN network