Evolutionary Intelligence
https://doi.org/10.1007/s12065-020-00470-0

SPECIAL ISSUE

Detection of Parkinson's disease from handwriting using deep learning: a comparative study

Catherine Taleb[1] · Laurence Likforman-Sulem[2] · Chafic Mokbel[1] · Maha Khachab[1]

Received: 21 March 2020 / Revised: 30 June 2020 / Accepted: 1 August 2020

Abstract
Degenerative disorders such as Parkinson's disease (PD) have an influence on daily activities due to rigidity of muscles, tremor or cognitive impairment. Micrographia, speech intensity, and deficient generation of voluntary saccadic eye movements (Pretegiani and Optican in Front Neurol 8:592, 2017) are manifestations of PD that can be used to devise noninvasive and low cost clinical tests. In this context, we have collected a multimodal dataset that we call Parkinson's disease Multi-Modal Collection (PDMultiMC), which includes online handwriting, speech signals, and eye movements recordings. We present here the handwriting dataset that we call HandPDMultiMC that will be made publicly available. The HandPDMultiMC dataset includes handwriting samples from 42 subjects (21 PD and 21 controls). In this work we investigate the application of various Deep learning architectures, namely the CNN and the CNN-BLSTM, to PD detection through time series classification. Various approaches such as Spectrograms have been applied to encode pen-based signals into images for the CNN model, while the raw time series are directly used in the CNN-BLSTM. In order to train these models for PD detection on large scale data, various data augmentation approaches for pen-based signals are proposed. Experimental results on our dataset show that the best performance for early PD detection (97.62% accuracy) is reached by a combination of CNN-BLSTM models trained with Jittering and Synthetic data augmentation approaches. We also illustrate that deep architectures can surpass the models trained on pre-engineered features even though the available data is small.

Keywords HandPDMultiMC dataset · Parkinson's disease (PD) · CNN · CNN-BLSTM · Handwriting · Data augmentation · Transfer learning

1 Introduction

Parkinson's disease (PD) is a neurological disorder caused by a decreased dopamine level in the brain. This disease is characterized by motor and non-motor symptoms. The motor symptoms consist of tremor, rigidity, slowness of movement or bradykinesia, micrographia (the fact that the writing size decreases along the text-line), gait and posture disturbances, and speech and swallowing difficulties [23]. Non-motor symptoms include disturbances in sleep, sensation, mood (depression, apathy, anxiety, obsessive-compulsive, psychotic), as well as autonomic and cognitive (memory, attention,...) disturbances. These symptoms may manifest in varying degrees and combinations in different individuals [18].

Neurological tests such as the unified Parkinson's disease rating scale (UPDRS) in addition to brain scans are routinely applied in the diagnosis [23]. These methods are expensive and need a high level of professional expertise. Hence, there is a need to define a reliable computer-based system that assists in the decision-making process for the diagnosis of PD. With the spread of digitizing devices, it's now possible to record sequences of measurements from handwriting tasks, provided by tablet and pen devices. We focus in this work on the definition of a deep computer-based PD detection system from handwriting. Machine learning are popular approaches for signal and image classification. They have been applied to the classification of sequences such as the speech signal or handwriting textline images. Deep learning has also been applied to the classification of

✉ Catherine Taleb
catherine.taleb@std.balamand.edu.lb

Laurence Likforman-Sulem
likforman@telecom-paris.fr

[1] University of Balamand, El-Koura, Lebanon

[2] LTCI, Telecom Paris, Institut Polytechnique de Paris, Paris, France

Published online: 10 September 2020

subjects into PD and healthy controls. In [22] Pereira et al. propose to encode the whole set of measurements of a given task into a 2D representation. The image corresponding to the 6 time series captured by a smartpen (microphone, fingergrip, axial pressure, tilt and acceleration in X, Y, and Z directions) is analyzed by a convolutional neural network (CNN). Moetesum et al. [14] exploited the static visual attributes of handwriting (a visual image of the drawing), to predict PD using CNNs. They extract separate discriminating visual features from the three representations of the input data (the raw image, median filter residual image, and edge image). The features extracted by 3 CNNs are combined and fed into and single SVM for classification. Khatamino et al. [12] applied a CNN model for PD classification from handwriting drawings where both dynamic features and static visual attributes are used. Gallicchio et al. [6] have proposed a novel approach for diagnosis of PD based on Deep Echo State Networks models where the deep recurrent model is fed by the whole time-series captured from a tablet.

When dealing with online handwriting time series, the variation over the time axis challenges models requiring a fixed dimension input. One approach would consist in normalizing the time series leading to a fixed dimension visual representation [22]. An alternative approach explicitly considers the time dimension, especially that the variation over this dimension is nonlinear. This paper proposes two deep learning classes of models in order to tackle the time variation in time series classification. In the first one 2D representations of time series are fed in a CNN. In the second one the time variation is integrated within a CNN Bidirectional Long-Short-Term Memory Networks (CNN-BLSTM) model which is directly applied on the time series. The paper is organized as follows. In Sect. 2, we describe the handwriting dataset we have collected at Saint George Hospital (Lebanon). In Sect. 3 we describe the deep learning systems including the CNN and CNN-BLSTM architectures (Sects. 3.2, 3.3). We investigate in Sect. 4 how to train these models from small data sets, using data augmentation and transfer learning approaches. We conducted several experiments that are described in Sect. 6. The experiments outcomes permit to benchmark the encodings, the architectures and show the advantage of the CNN-BLSTM over an SVM and a CNN. Conclusions and perspectives are drawn in Sect. 7.

2 HandPDMultiMC dataset

PDMultiMC is a multimodal database that includes handwriting tasks, speech samples, and eye movements recordings collected from PD patients in two phases (medication on and medication off) and from control subjects [24]. PD patients were selected from those attending an experienced neurologist at Saint George Hospital (Lebanon). The control group is composed of healthy subjects matched for age, years of education, and hand dominance.

In this work, we focus on the handwriting modality for the detection of PD. Handwriting samples are taken from the HandPDMultiMC subset (a part of PDMultiMC database that will be released on the IAPR TC11). Seven handwriting tasks were recorded for each of the 21 HC controls and 21 PD patients in their “on- state” (with dopaminergic medication). Participants were required to write on a sheet of paper laid on the tablet. The handwriting tasks are separated into two parts. Part I includes the free writing tasks. In the copying tasks (Part II), participants are asked to copy patterns and words which were preprinted into 3 different languages on the left of the sheet paper placed on the tablet. The seven tasks are displayed in Fig. 1 and are:

- Task 1: Drawing repetitive cursive letter ℓ
- Task 2: Drawing a triangular wave
- Task 3: Drawing a rectangular wave. For tasks 1–3, subjects were asked to proceed copying the patterns from left to right until 10 cycles.
- Task 4: Repetitive writing of word ‘Monday’ within the word sequence Monday–Tuesday. These words may be written in the subject’s familiar language. Subjects were asked to write this sequence 5 times.
- Task 5: Repetitive writing of word ‘Tuesday’. (See Task 4 remarks).
- Task 6: Repetitive writing of first name.
- Task 7: Repetitive writing of last name. For Tasks 6 and 7, subjects were asked to write their full name 5 times, each time on a different line.

Data have been registered using a Wacom Intuos 5 tablet and a special pen device, with a sample rate of 197 points/s and high spatial and pressure accuracies. The following measurements are collected per sample point:

- Pen tip position in X-axis, Y-axis, and Z-axis. The Z coordinate is registered when the pen tip is within 0–1 cm above the tablet. When Z equals 0, the pen is on tablet and when $Z > 0$ an in-air point is registered.
- Pen tip pressure on the surface of the tablet.
- Altitude and Azimuth angle of the pen with respect to the tablet.
- Time stamp.

Tasks are chosen in a manner to highlight as much as possible the differences between PD and control. The first 3 tasks demand continuous pen movement; which emphasis hypokinesia and tremor. Micrographia needs long writing task to manifest, that’s why we chose the words repetition

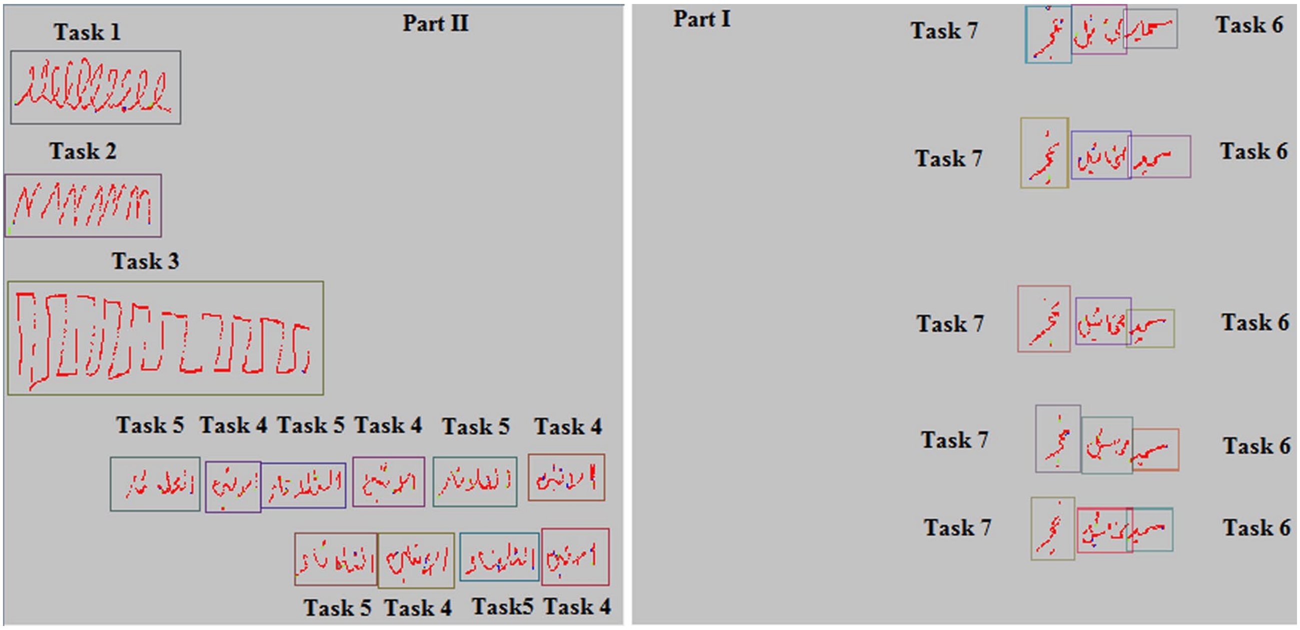
Task 1
Part II
Part I
Task 7
Task 6
Task 2
Task 7
Task 6
Task 3
Task 7
Task 6
Task 5
Task 4
Task 5
Task 4
Task 5
Task 4
Task 7
Task 6
Task 7
Task 6
Task 5
Task 4
Task5
Task 4

Fig. 1 The seven tasks segmented from the sheet filled by a PD subject

in the other 4 tasks. These tasks were verified by an experienced neurologist at Saint George Hospital-Lebanon. The data can be represented as time series. The outputs of Task 1 from a healthy subject and a PD patient are depicted in Fig. 2. The differences between drawings are not intuitively recognizable, where the signals extracted from PD patient are noisier than those of the control subject.

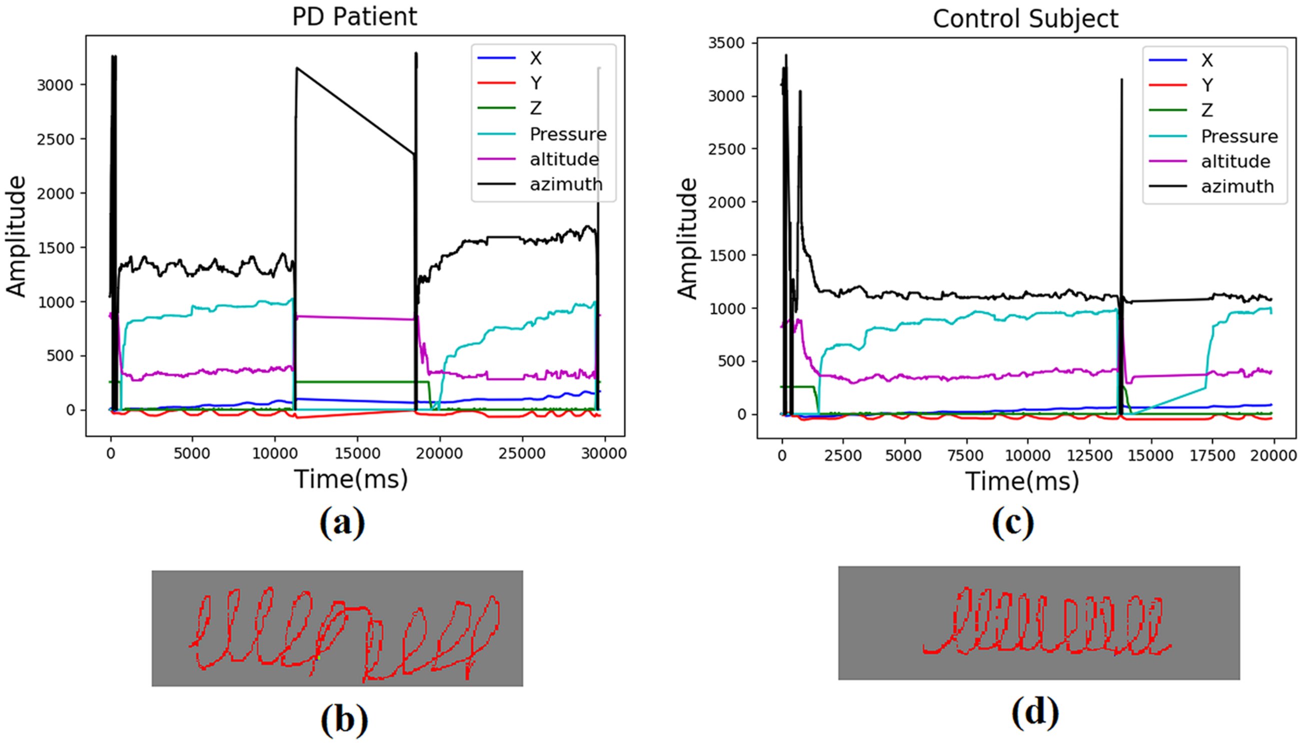

Fig. 2 Repetitive ℓ from PD (**b**) and control (**d**) subjects, and respective signals (**a**) and (**c**)

3 Deep learning models for times series classification

In recent years, convolutional neural networks have shown excellent performance on image classification tasks [7]. In order to benefit from this, some researchers proposed to transform a time series into an image and provide it as an input to CNN, which learn features that are used to distinguish healthy individuals from PD patients [22]. In parallel, LSTM Networks are a family of recurrent neural networks that excels in learning from sequential data and can cope with variable length time series [1], e.g. handwriting signals [9]. Time series can be represented as 1D (time series) or 2D arrays (images) and provided at the input of CNN-based models. This work explores the two representations.

Pre-processing HandPDMultiMC database includes Arabic, French and English samples. In order to have the same writing direction, the X coordinates of the Arabic samples are flipped. After that, the X and Y coordinates are normalized to achieve a uniform range across all subjects by subtracting the minimum and the mean calculated for each sequence, respectively. For all our models, all images and raw time series are normalized to the range (0, 1) to achieve a uniform contrast and intensity range.

3.1 2D representations of time series

Each handwriting task is composed of n rows (time) and 7 columns (X, Y, Z, pressure, altitude, azimuth, and time stamp). An optimal selection of time series features to be used is performed and a hyper-parameter k between 1 and 7 must be determined. The first approach for 2D representation consists in concatenating the k time series and reshaping the result as an image. The second approach computes spectrograms and use them as 2D representations.

Concatenation approach (time series-based) This representation is inspired from [22]. It consists in transforming k time series of length n into a single image. The whole data ($n \times k$ matrix) is transformed into one image by concatenating the n rows into one vector and then reshaping it into a square matrix of size $(\sqrt{n \times k}, \sqrt{n \times k})$. This squared matrix is resized to 64 × 64 pixels resolution using Lanczos resampling method [10]. In contrast to [22] we search for the best k measurements to include in the 2D representation. Observing the time series-based image of a patient and a healthy subject for the 7 tasks (Fig. 3) show a variation across the tasks; which tends to indicate that each task may capture distinct information.

Spectrogram (image-based) Handwritten dynamics signals are considered as nonstationary. Time-frequency representations method is specified to analyze such signals [11] where Short Time Fourier Transforms (STFT) are computed on sliding windows of the signal. The time-frequency resolution depends on the window size and type [19]. Based on our experiments blackman windowing with window length 256 and overlapping rate 50% provide the best spectrogram resolution. When treating a spectrogram like an image, the number of frequencies and the number of time bins in the spectrogram refer to the height and the width of the image in pixels. The numerical "brightness" value of each pixel is then equal to the output value of the spectrogram [17]. These values are converted to a logarithmic scale then normalized to [0, 1] generating a grayscale image [10]. The width of the image depends on the length of the signal. To keep the number of input feature maps identical, the area of spectrogram should be the same for all subjects. We apply Lanczos technique to resize the spectrogram images to size 64 × 64 [10]. Figure 4 provides an illustration.

3.2 CNN architecture

The CNN model architecture is represented in Fig. 5. The overall architecture consists of 2 main parts, the feature extractor and the classifier.

The feature extractor layers consist of two convolution layers, each followed by Relu activation function, and two pooling layers.The convolutional layers employ kernels of size 5 × 5 with stride of 1 pixel, and the maxpooling operations are applied on regions of size 2 × 2, with stride 2. The convolutional layers convert the 64 × 64 pixel input image into 64 feature maps of size 16 × 16. It can be noted that different convolution (Relu unit) and pooling operations are applied on each image in order to learn different features

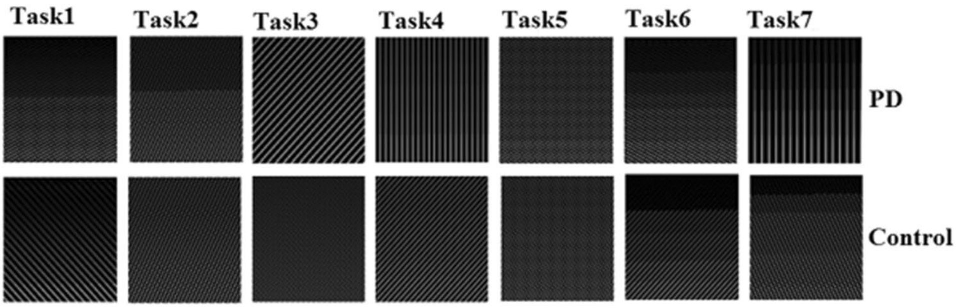

Fig. 3 Time series-based images of PD (top) and control (bottom) subjects for the 7 tasks

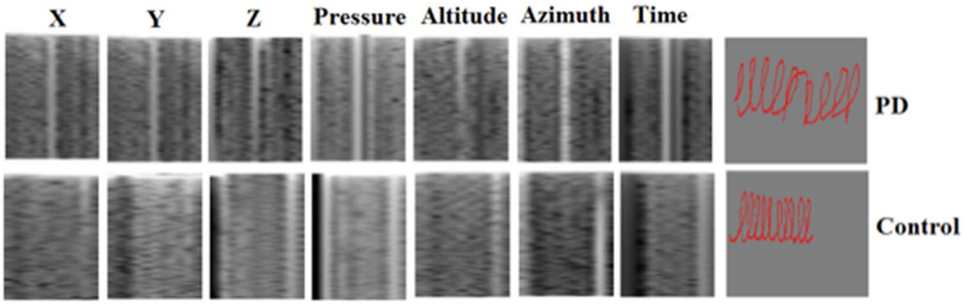

Fig. 4 The spectrograms of PD and control subjects in Task 1 for the 7 signals

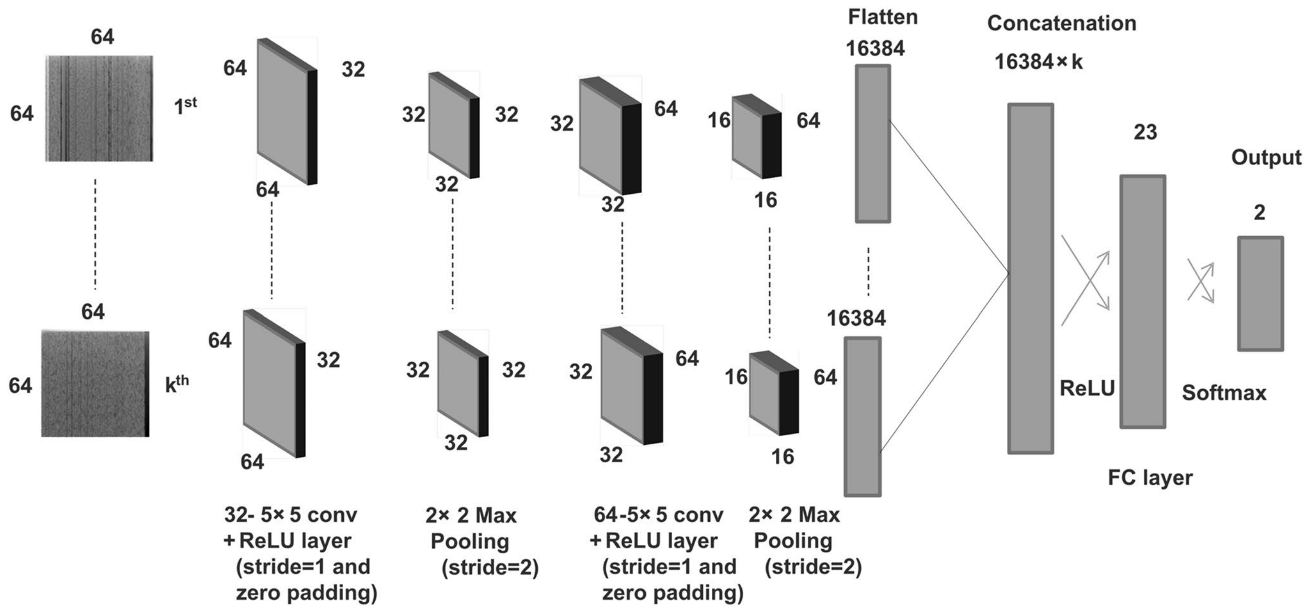

Fig. 5 Single-task CNN architecture with input k 2D representations of 1D time series

independently. At a later stage, the outputs of the convolution layers are flattened, then concatenated and fed into densely connected layer to make a prediction. The number of input images is a hyper-parameter k, $(1 \leq k \leq 7)$. This CNN model can be used for classification from a single image including k measurements (time-series based), or classification from k measurements, (i.e. k images).

3.3 CNN-BLSTM architecture

The CNN-BLSTM architecture (Fig. 6) consists in using CNN layers for feature extraction combined with BLSTMs to support sequence prediction. Instead of converting the time series into images, the entire raw time series are used here as input to the model. The convolutional layers are constructed using one-dimensional kernels that move through the sequence.The output of the CNN is a sequence of length n/4 of vectors of size 32, where n represents the time series length. This sequence is then used as input to a BLSTM. The number of input time series k is a hyper-parameter $(1 \leq k \leq 7)$.

3.4 All tasks combination approach

To enhance recognition, seven single-task systems can be combined into one all-task system. The combination approach considered in this part is the majority voting. In order to get the best time series features combination (the hyper-parameter k) a suboptimal incremental approach has been used. The feature providing the highest overall classification accuracy is first selected. Then, features are added incrementally by selecting, at every iteration, the one yielding the highest overall classification accuracy. The iterations stop when no more increase in performance is observed. Two strategies will be applied to solve the overfitting problem due to the small dataset that we are working on: transfer learning and data augmentation.

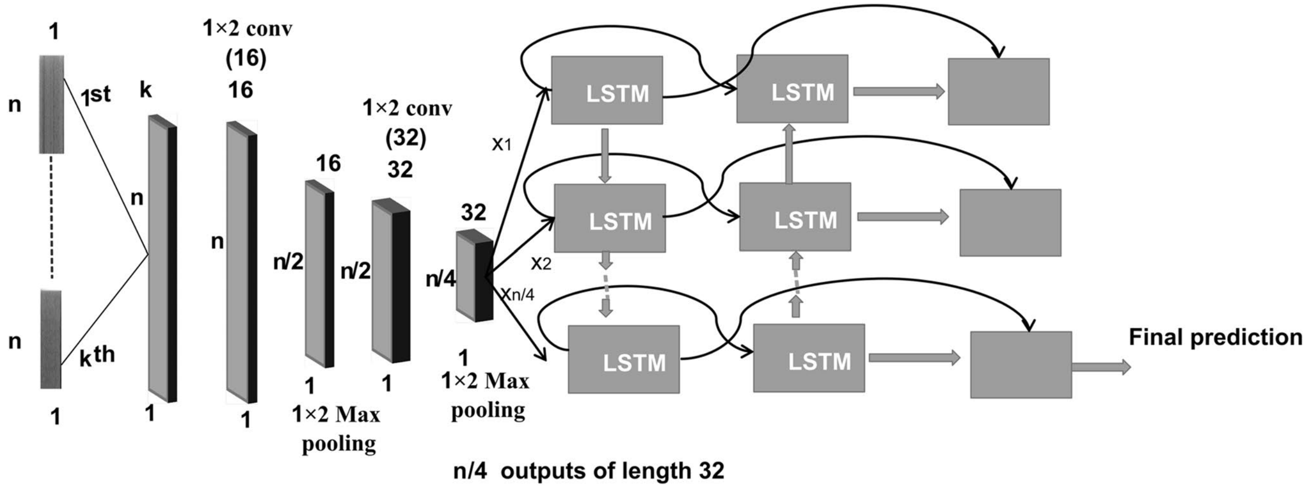

Fig. 6 Single-task CNN-BLSTM architecture on multivariate time series

4 Strategies to avoid overfitting

Deep learning has shown excellent performance where large datasets are available. It becomes challenging to apply deep learning to problems where only limited size datasets are available like medical data [26]. Training a neural network with a small dataset can cause the network to memorize all training examples, in turn leading to poor performance on a holdout dataset [2]. To solve this problem, also known as overfitting, different techniques can be applied such as collecting more data (which can be hard to obtain), using transfer learning [15], or using data augmentation.

4.1 Transfer learning process

Pre-trained models which have been previously trained on large datasets, can bootstrap the training on our limited dataset [13]. Here we train the CNN model (Fig. 5) on a larger handwriting dataset, namely the PaHaw dataset [4]. PaHaW and HandPD are two available handwriting datasets slightly larger than HandPDMultiMC. HandPD includes spiral and meander tasks collected from 92 subjects (74 PD and 18 Controls) [22], and where time series signals were captured by a biosensor smart pen (BiSP). In comparison, handwriting tasks and time series signals in PaHaW are the closest to HandPDMultiMC dataset (loops and time series). Thus, PaHaw was selected [4]. The differences between the 2 sets are that the Z coordinate feature is missing in PaHaW, and the number of tasks is 8 instead of 7. To match the two datasets we eliminate Task 8 in PaHaW and the Z coordinate feature in HandPDMultiMC.

Different transfer learning strategies were studied and are summarized in Fig. 7. The first one freezes all the PaHaW-trained model layers and a new softmax classifier is trained on HandPDMultiMC dataset. The softmax contains relatively few parameters and can be trained from a relatively small number of examples [28]. The second strategy freezes only a part of the PaHaW-trained model. Actually, lower convolutional layers refer to general features, while higher layers refer to specific features [13]. We studied 2 partial freeze strategies; freezing the whole

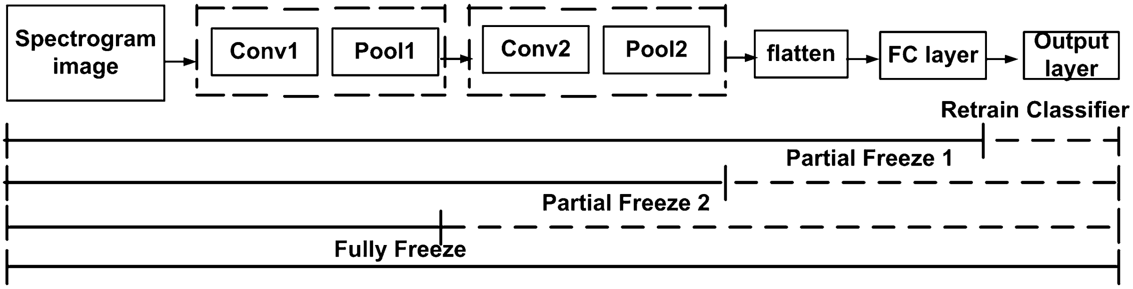

Fig. 7 Different transfer learning strategies: dashed lines indicate that blocks are retrained

convolutional base, and freezing only the first layers of the convolutional base. Last, we consider full freezing of all layers.

4.2 Data augmentation applied to time series

Another way to prevent overfitting is to enlarge the training sets by generating synthetic examples. It is important to find the proper data augmentation method that will increase the recognition accuracy. There are two basic data augmentation approaches used in image processing: geometric transformation (shift, scale, rotation/reflection, time-wraping, etc.) and noise addition [26]. Minor changes will not alter the data labels because they are likely to happen in real world (imperfect pen sensors) [26]. Jittering, scaling, time-warping, and synthetic data generation techniques are used to generate new time series samples. These methods do not crop time series into shorter subsequences.

Jittering It is considered as a way of simulating additive sensor noise. We focus on adding Gaussian noise to each feature time series [27]. In order to ensure that the average amplitude will not be changed, we generate the Gaussian noise with $\mu = 0$ [27]. In order to explore the effect of noise intensity (standard deviation (STD)) and the augmented multiple (m) different values of STD and m are studied. Actually, too little noise has no effect, whereas too much noise makes the mapping function too challenging to learn or may alter the labels because it introduces rapid fluctuations similar to tremor [26].

Scaling Scaling changes the magnitude of the data in a window by multiplying by a random scalar [26]. It is considered as a way of simulating multiplicative sensor noise. We also focus on multiplying Gaussian noise (with a non-zero mean) to each time series [26]. Different values of m and STD are studied.

Time-warping Time-warping is a way to perturb the temporal location by smoothly distorting the time intervals between samples [26].

Generating synthetic data To create the synthetic time series, some authors propose to average a set of time series and to use the averaged time series as a newly created example [5]. In this work, we are working with time series of variable lengths. First of all the training data are separated into subsets of the same class label. The next step assigns weights for each subset separately based on the following [5]: a random initial time series chosen from the subset is initially assigned with a weight equal to 0.5, then the 5 nearest neighbors using the Dynamic Time Warping (DTW) distance are found and 2 out of these 5 are randomly selected and assigned with a weight equal to 0.15 each, finally the remaining time series will share the rest of the weight 0.2.

5 Combination approach

The experiments in Sect. 6 are divided into two rounds: single assessment and combined assessment [22]. In the single assessment, we analyze each task separately, while in the combined assessment we combine the outputs of the 7 models (one per task) in order to find the final label and obtain the overall performance of the all-task system [25]. Two combination schemes are considered: majority voting and MLP-based combinations. Majority voting consists on choosing the class label which has the maximum number of vote by each classifier. MLP-based combination consists in combining the probability vectors of size 2 (probability of a Parkinsonian or a control subject given the time series) provided by each of the 7 models. The MLP model is composed of an input layer of $2 \times 7 = 14$ nodes, a single hidden layer of 40 nodes with Rectified Linear Unit (ReLU) activation function, and 2 output nodes (corresponding to PD and control) with softmax activation function. Majority voting is used as a baseline combination scheme when systems are trained using transfer learning, while MLP combination is an enhanced combination scheme used in conjunction with data augmentation.

In addition, several all-task systems can be trained from our architecture. For instance by using distinct data augmentation approaches. For combining these all-task systems, a meta-MLP approach is used that combines the outputs of the previous MLPs, 2 per all-task system.

6 Experiments and discussions

The performance was figured out in term of accuracy, sensitivity, and specificity. The models described in Sect. 3 are tested; where threefolds cross validation was applied where stratified sampling method was used in order to insure the same class distribution in all the subsamples. The performances in Table 1 represent the average of 3 runs (the highest performance accuracy in bold). According to the results, both the CNN model with spectrogram images as input and the CNN-BLSTM model yield the best accuracy (83.33%), where the best feature sets selected do not include the time stamp feature. This finding is obvious since the multivariate time series are generated with a fixed synchronized sampling.

After selecting the best deep learning model for time series classification, transfer learning and data augmentation techniques (Sect. 4) are applied on the best time series combination (X, Y, Z, pressure and altitude for the CNN and X, Y, Z, pressure, altitude, and azimuth for the CNN-BLSTM). Different parameter values (STD and the augmented multiple m) are experimented. For jittering STD=0.3 is chosen. For scaling, a random scalar is sampled from a Gaussian distribution with a mean of 1 and 0.1 STD (to avoid negative values). For time-warping, random sinusoidal curves are generated using arbitrary amplitude frequency and phase values [26]. We achieved the best accuracy when the training data is augmented twice. The new times series are either used directly with the CNN-BLSTM model, or converted into spectrogram images with the CNN model (Sect. 3).

The different transfer learning strategies (Sect. 4) are first compared with the CNN model; where majority voting is applied to merge the results provided by the 7 models (referring to the 7 tasks). Based on the results in Table 2 (three-folds CV), fully freezing performs worse than other strategies; where incremental performance may be observed when more convolutional layers are included in fine-tuning (partial freeze 1 and partial freeze 2). In definitive, using PaHaW database to pre-train the CNN model performs worse than training from scratch. This may be due to the absence of Z coordinate feature in this database, and while this feature seems to play a role in classification.

Moving to data augmentation strategy, the results of the proposed techniques are presented in Table 3. The CNN and the CNN-BLSTM all-task models are used here, and a Multi-Layer Perceptron (MLP) model is applied to combine the probability vectors provided by the 7 models. Scaling fails to improve the CNN-BLSTM performance because changing in the intensity of the signal may alter the labels [26]. On the other hand, jittering, Time-Warping, and creating synthetic time series by averaging a set of time series used with CNN-BLSTM model improve the accuracy by 7.15%. Data augmentation improves the CNN-BLSTM performance and fails to improve the CNN performance because the CNN-BLSTM deals with time series directly without encoding them into spectrogram images (where it mostly adds a bias).

Task-wise system accuracies for the developed models in this study and the SVM model developed in our previous study [24] are represented in Table 4; where D1, D2, D3,

Table 1 Threefold CV performance measures of all-task system considering the majority voting

Model	Data input	Overall perf. (%) Acc (Sens, Spec)	Best features combination
CNN	Time series-based images	80.95 (85.71, 76.19)	Pressure
CNN	Spectrogram images	**83.33** (85.71, 80.95)	X + Y + Z + Pressure + Altitude
CNN-BLSTM	Raw time series	**83.33** (71.43, 95.24)	X + Y + Z + Pressure + Altitude + Azimuth

The bold values refer to the highest performance

Table 2 Transfer learning strategies for the spectrogram CNN model with majority voting

Transfer learning strategy	Best features combination	Overall per (%) Acc (Sens, Spec)
No transfer learning	X + Y + Z + Pressure + Altitude	**83.33** (85.71, 80.95)
Retrain classification layer	X + Y + Pressure + Altitude	54.76 (28.57, 80.95)
Partial freeze 1	X + Y + Pressure + Altitude	66.67 (66.67, 66.67)
Partial freeze 2	X + Y + Pressure + Altitude	66.67 (66.67, 66.67)
Fully freeze images	X + Y + Pressure + Altitude	45.24 (71.43, 19.05)

The bold value refer to the highest performance

Table 3 Performance with data augmentation and MLP combination

Model	Data input	Augmentation technique	Best features combination	Overall per (%) Acc (Sens, Spec)
CNN	Spectrogram images	Jitter	X + Y + Z + Pressure + Altitude	83.33 (85.71, 80.95)
CNN-BLSTM	Raw time series	Jitter	X + Y + Pressure + Altitude + Azimuth	**90.48** (95.24, 85.71)
CNN-BLSTM	Raw time series	Scaling	X + Y + Pressure + Altitude + Azimuth	59.51 (19.05, 100)
CNN-BLSTM	Raw time series	Time-warping	X + Y + Pressure + Altitude + Azimuth	**90.48** (90.48, 90.48)
CNN-BLSTM	Raw time series	Synthetic data	X + Y + Pressure + Altitude + Azimuth	**90.48** (85.71, 95.24)

The bold values refer to the highest performance

Table 4 Task-wise system and all-tasks system accuracies (in %)

Task	D1	D2	D3	D4
Repetitive cursive letter ℓ	**87.5**	59.52	57.14	47.62
Triangular wave	**93.75**	80.95	83.33	78.57
Rectangular wave	**90.63**	71.43	69.05	76.19
Repetitive “Monday”	**87.5**	78.57	66.67	76.19
Repetitive “Tuesday”	**87.5**	57.14	47.62	59.52
Repetitive “Name”	**84.38**	57.14	42.86	50
Repetitive “Family Name”	**84.38**	69.05	71.43	64.29
All tasks (MLP combination)	**96.87**	**90.48**	**90.48**	**90.48**

The bold values refer to the highest performance

D1: SVM, D2: CNN-BLSTM/Jitter, D3: CNN-BLSTM/Time-Warping, D4: CNN-BLSTM/Synthetic data

Table 5 Performance (in %) obtained by combining all-task CNN-BLSTM models

Performance	D2 + D3	**D2 + D4**	D3 + D4	D2 + D3 + D4
Accuracy	92.86	**97.62**	92.86	92.86
Sensitivity	90.48	**95.24**	95.24	95.24
Specificity	95.24	**100**	90.48	90.48

The bold values refer to the highest performance

and D4 models refer to SVM, CNN-BLSTM with jittering augmentation, CNN-BLSTM with time-warping augmentation, and CNN-BLSTM with synthetic augmentation models respectively. It can be observed from Table 4 that “all tasks” reports highest accuracies across all the 4 models. Additionally, Tasks 2 and 3 (triangular and rectangular waves) report highest accuracies across all the 4 models. These two tasks are considered long and somehow complex. Copying these cursive tasks involves higher cognitive force and thus explains the effect of disease on handwriting. This conclusion coroborates with what is found in [24].

Experiments have also been carried out by combining the results obtained from the CNN-BLSTM all-task model with different data augmentation methods (D2, D3, and D4 in Table 4) using an MLP composed of an input layer of L nodes, a single hidden layer of H nodes with Rectified Linear Unit (ReLU) activation function, and 2 output nodes (corresponding to PD and control) with softmax activation function. (L, H) = (4, 30) and (L, H) = (6, 35) when 2 or 3 data augmentation methods are combined respectively. The performance measures are summarized in Table 5 where it can be seen that combining the results of various data augmentation methods show better performance than that of a single data augmentation. The highest accuracy is 97.62% obtained when combining Jittering and Synthetic data augmentation methods. Time-warping augmentation method fails to improve the performance when included. From a clinical point of view, inter-samples timing disturbances occur due to the neuro-motor dysfunctions affecting wrist and finger movement of PD patients [4, 8]. As mentioned before, the temporal locations of the samples are changed by time-warping; which will look similar to inter-samples time disturbances.

The best final model that classifies people between PD and healthy controls with an accuracy of 97.62% is summarized in Fig. 8, where Jittering and Synthetic data augmentation techniques are applied separately on “All-tasks” system. Two MLPs models (MLP1 and MLP2) are applied, where each one is used to combine the probability vectors (each of size 2) obtained by the 7 CNN-BLSTM models (with the best features set see Table 1) trained with distinct data augmentation approach. At a later stage, another MLP model (MLP3) combines the probability vectors provided by each of MLP1 and MLP2 providing the final prediction. This 204,060 parameters model was trained with Nvidia GTX 1080 GPU of 8 GB memory. The time required for the training is around 1 day.

An overview of the existing models applied for PD detection through handwriting analysis are summarized in Table 6. In [3, 16, 24] the authors have applied SVM models that are trained on temporal, pressure, and intrinsic features yielding a classification accuracy of 89.09% [3], where [24] reports a higher accuracy of 96.87% when features related to the correlation between kinematic and pressure are used. Mucha et al. [16] report an accuracy of 97.14% for a combination of kinematic and temporal features that are extracted

Table 6 Performance comparison between our best proposed model and previous works

Refs.	Database	Model	Features	Perf (%) Acc (Sens, Spec)
Drotar et al. [3]	PaHaW	SVM	Kinematic, temporal, spatial, entropy, EMD, pressure (on-paper)	89.09 (n/a, n/a)
Taleb et al. [24]	HandPDMultiMC	SVM	Kinematic, stroke, entropy, EMD, pressure (on-paper)	**96.87** (93.75, 100)
Mucha et al. [16]	PaHaW	XGBoost	Kinematic, temporal (on-paper and in-air)	**97.14** (95.5, 100)
Pereira et al. [22]	HandPD	CNN-ImageNet	CNN-based features (on-paper and in-air)	93.42 (97.84, 89.0)
Proposed model	HandPDMultiMC	CNN-BLSTM	CNN-based features (on-paper and in-air)	**97.62** (95.24, 100)

The bold values refer to the highest performance

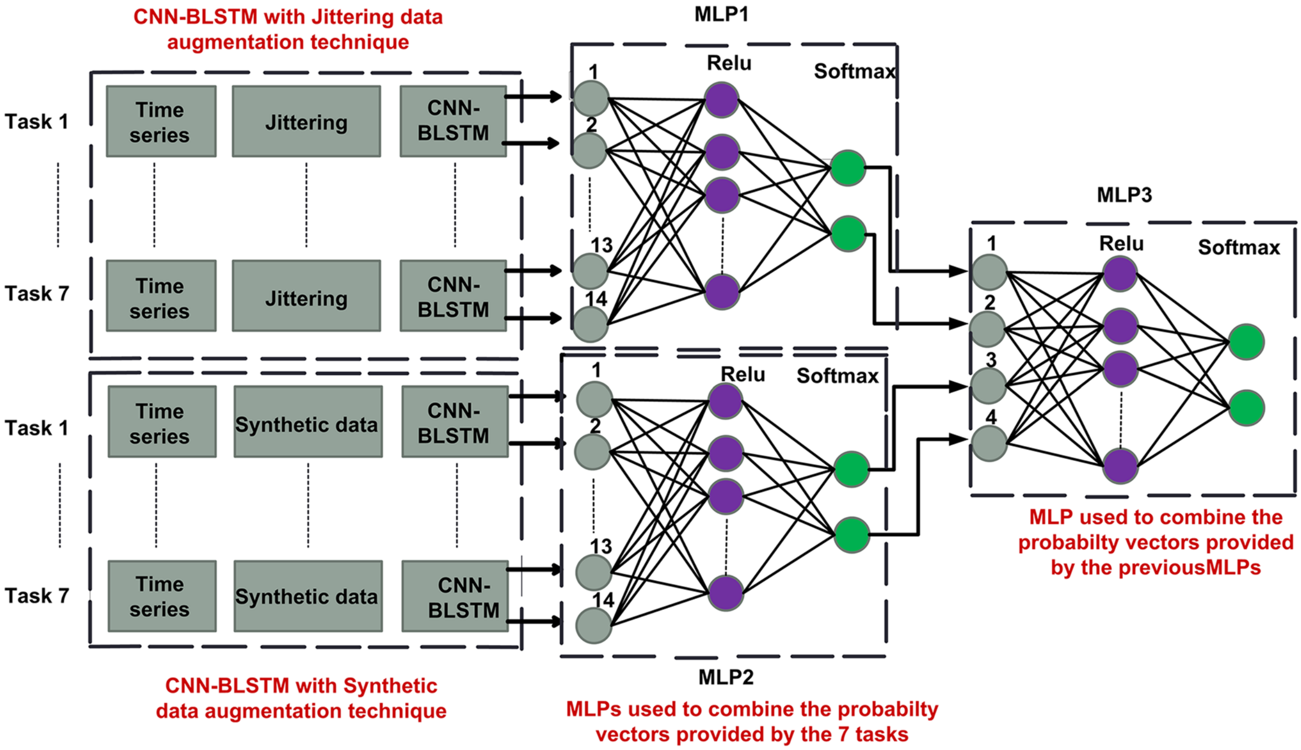
CNN-BLSTM with Jittering data augmentation technique
Task 1
Task 7
Time series
Jittering
CNN-BLSTM
MLP1
Relu
Softmax
1
2
13
14
MLP3
Relu
Softmax
1
2
3
4
Task 1
Task 7
Time series
Synthetic data
CNN-BLSTM
Relu
Softmax
MLP2
MLP used to combine the probabilty vectors provided by the previousMLPs
CNN-BLSTM with Synthetic data augmentation technique
MLPs used to combine the probabilty vectors provided by the 7 tasks

Fig. 8 Best final model combining using MLPs the outputs of 14 CNN-BLSTM

for both "on-paper" and "in-air". In [22] the handwriting signals are transformed into images and fed into a deep CNN model yielding an accuracy of 93.42%.

Finally, in this work a CNN-BLSTM model with the combination of 2 different data augmentation techniques (Jittering and Synthetic data) returns the best accuracy.Accuracy along with 95% confidence intervals for this system is: 97.62% (93.01–100%). This model with all-tasks SVM yield significantly (at 95%) better performance than all experimented models.

Table 6 shows that our deep learning model reports highest performance (although results are not always measured on the same database), especially when compared to [24]. When checking if data augmentation is effective for SVM model, no improvement has been obtained. For the sake of comparison, we have conducted 2 more experiments. First, our best model has been trained and tested on the PaHaW database. In this database there is no Z coordinate so that training was performed with X, Y, pressure, altitude and azimuth time series. Performance obtained for this system is: accuracy = 88.10%, sensitivity = 85.71% and specificity 90.48%. The second experiment consists in training and test our best system on the HandPDMultiMC dataset using X, Y, pressure, altitude and azimuth time series, and removing the Z coordinates. This yields an accuracy equal to 90.48%. When our best model is trained and tested using HandPDMultiMC dataset with Z coordinate, the accuracy obtained is 97.62%, whereas the accuracy obtained when eliminating the Z coordinate is 90.48%. Moreover, when the system is trained and tested on PaHaW dataset, the accuracy obtained (88.1%) is close to the one obtained with HandPDMultiMC, without Z feature. These results confirm the importance of Z feature and the relevance of the results obtained.

7 Conclusions

In this paper, an automatic classification system for PD detection is developed based on online handwriting. Two deep-learning models, trained end-to-end, have been proposed for time series classification, namely the CNN and the CNN-BLSTM. For the CNN model, two different approaches were proposed to encode time series into images: the concatenation approach and the spectrogram approach, where for the CNN-BLSTM model the raw time series are directly used. We have demonstrated the importance and specificity of each of the two models: a deep architecture based on the combination of 1D CNN and BLSTM, and a CNN model with spectrograms as input, since they both consider local short term information before normalization. In comparison, the concatenation approaches will normalize the time series into a fixed dimension image without extracting local information yielding lower performance. Moreover, the CNN-BLSTM model takes advantage of learning the temporal feature activation dynamics [20].

Deep learning models require larger number of training samples compared to SVM. This makes PD classification using deep learning a challenging task due to the limited data availability. To cope with this, two classes of approaches were reported in this paper. Firstly, multiple transfer learning strategies across the CNN model for time series classification were investigated and compared. It was found that the more convolutional layers included in the fine-tuning, the better performance we get. However, there are no gains of transfer learning over training our CNN from scratch. We believe this is due to the absence of Z coordinate feature in PaHaW database. Secondly, jittering, scaling, Time-Warping, and synthetic data generation techniques are used for data augmentation. The challenging PD task is successfully tackled using the CNN-BLSTM and the combination of jittering and synthetic data augmentation methods. The accuracy improves from 83.33% (no data augmentation) to 97.62%.

Main findings The local short term information allows the deep learning models to provide better classification results compared to a globally normalized fixed dimension visual representation. Z coordinates are important in Parkinson's disease classification. Data augmentation over transfer learning are effective at reducing error and decreasing overfitting. Time-warping technique fails to improve the performance of PD classification due to the distortion of time intervals similar to inter-samples time disturbances; which is one of the early marks of PD. Data augmentation methods are more effective when time series are used directly with deep learning. Finally, we have shown the superiority of deep architectures with data augmentation over an SVM model trained on pre-engineered features, even though the available data is small.

In future work our model shall be tested on other PD databases (PaHaW, HandPD) for further validation of the conclusions. We shall also study the combination of two bio-signals (handwriting and speech) for PD detection.

Acknowledgements This study has been approved by the institutional review board (IRB) of the University of Balamand and Saint George Hospital University Medical Center.

References

1. Atienza R (2017) LSTM by example using Tensor-flow. https://towardsdatascience.com/lstm-by-example-using-tensorflow-feb0c1968537. Accessed 3 July 2019
2. Brownlee J (2019) Train neural networks with noise to reduce overfitting. Machine learning mastery. https://machinelearningmastery.com/train-neural-networks-with-noise-to-reduce-overfitting. Accessed 11 Sept 2019
3. Drotar P et al (2015) Contribution of different handwriting modalities to differential diagnosis of Parkinson's disease. In: IEEE international symposium on medical measurements and applications (MeMeA), pp 344–348
4. Drotar P et al (2016) Evaluation of handwriting kinematics and pressure for differential diagnosis of Parkinson's disease. Artif Intell Med 67:39–46
5. Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller PA (2018) arxiv.org/abs/1808.02455
6. Gallicchio C, Micheli A, Pedrelli L (2018) Deep tree echo state networks. IJCNN. https://doi.org/10.1109/ijcnn.2018.8489464
7. Gamboa JCB (2017) Deep learning for time-series analysis. https://arxiv.org/pdf/1701.01887.pdf. Accessed 3 July 2019
8. Gómez-Vilda P et al (2017) Parkinson disease detection from speech articulation neuromechanics. Front Neuroinform 11:56
9. Himmetoglu B (2017) Time series classification with Tensor flow. https://burakhimmetoglu.com/2017/08/22/time-series-classification-with-tensorflow/. Accessed 3 July 2019
10. Huzaifah M (2017) Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. arxiv.org/abs/1706.07156
11. Khan NA, Jafri MN, Qazi SA (2011) Improved resolution short time Fourier transform. In: 7th International conference on emerging technologies
12. Khatamino P, Cantürk I, Özyılmaz L (2018) A deep learning—CNN based system for medical diagnosis: an application on Parkinson's disease handwriting drawings. In: Proceedings of the 6th international conference control engineering information technology, pp 1–6
13. Marcelino P (2019) Transfer learning from pre-trained models. https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751. Accessed 11 Sept 2019
14. Moetesum M, Siddiqi I, Vincent N, Cloppet F (2019) Assessing visual attributes of handwriting for prediction of neurological disorders: a case study on Parkinson's disease. Pattern Recognit Lett 121:19–27
15. Mormont R, Geurts P, Andmarée R (2018) Comparison of deep transfer learning strategies for digital pathology. In: IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 2262–2271
16. Mucha J et al (2018) Identification and monitoring of Parkinson's disease dysgraphia based on fractional-order derivatives of online handwriting. In: 41st International conference on telecommunications and signal processing, pp 1–4
17. Naumov E (2017) A convolutional network on EEG spectrograms for sleep staging. M.S.thesis, College of CS. McGill Univ., Montreal
18. Nilashi M, Ibrahim O, Ahani A (2016) Accuracy improvement for predicting Parkinson's. Dis Prog Sci Rep 6(1):1–18
19. Nisar S, Khan OU, Tariq M (2016) An efficient adaptive window size selection method for improving spectrogram visualization. Comput Intell Neurosci. https://doi.org/10.1155/2016/6172453
20. Ordóñez F, Roggen D (2016) Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1):115–115
21. Pretegiani E, Optican LM (2017) Eye movements in Parkinson's disease and inherited Parkinsonian syndromes. Front Neurol 8:592
22. Pereira CR et al (2018) Handwritten dynamics assessment through convolutional neural networks: an application to Parkinson's disease identification. Artif Intell Med 87:67–77
23. Sharma RK, Gupta AK (2015) Voice analysis for telediagnosis of Parkinson disease using artificial neural networks and support vector machines. Int J Intell Syst Appl 7(6):41–47
24. Taleb C, Likforman L, Khachab M, Mokbel C (2017) Feature selection for an improved Parkinson's disease identification based

on handwriting. In: 1st International workshop on arabic script analysis and recognition. ASAR, Nancy, France

25. Taleb C, Likforman L, Khachab M, Mokbel C (2019) Visual representation of online handwriting time series for deep learning Parkinson's disease detection. In: 3rd International workshop on arabic and derived script analysis and recognition, ASAR@ ICDAR 2019. Sydney, Australia
26. Um T et al (2017) Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. In: Proceedings of the 19th ACM international conference on multimodal interaction-ICMI
27. Wang F, Zhong S, Peng J, Jiang J, Liu Y (2018) Data augmentation for EEG-based emotion recognition with deep convolutional neural networks. MultiMedia Model Lect Notes Comput Sci. https://doi.org/10.1007/978-3-319-73600-6_8
28. Zeiler DM, Fergus R (2013) Visualizing and understanding convolutional networks. Comput Vision – ECCV 2014. Lect Notes Comput Sci. https://doi.org/10.1007/978-3-319-10590-1_53

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.