Assessment of Parkinson's Disease Severity From Videos Using Deep Architectures

Yin, Z.; Geraedts, V.J.; Wang, Z.Q.; Contarino, M.F.; Dibeklioglu, H.; Gemert, J. van

Assessment of Parkinson's Disease Severity From Videos Using Deep Architectures

Zhao Yin, Victor J. Geraedts, Ziqi Wang, Maria Fiorella Contarino, Hamdi Dibeklioglu, and Jan van Gemert

***Abstract*—Parkinson's disease (*PD*) diagnosis is based on clinical criteria, i.e., bradykinesia, rest tremor, rigidity, etc. Assessment of the severity of PD symptoms with clinical rating scales, however, is subject to inter-rater variability. In this paper, we propose a deep learning based automatic PD diagnosis method using videos to assist the diagnosis in clinical practices. We deploy a 3D Convolutional Neural Network (*CNN*) as the baseline approach for the PD severity classification and show the effectiveness. Due to the lack of data in clinical field, we explore the possibility of transfer learning from non-medical dataset and show that PD severity classification can benefit from it. To bridge the domain discrepancy between medical and non-medical datasets, we let the network focus more on the subtle temporal visual cues, i.e., the frequency of tremors, by designing a Temporal Self-Attention (*TSA*) mechanism. Seven tasks from the Movement Disorders Society - Unified PD rating scale (*MDS-UPDRS*) part III are investigated, which reveal the symptoms of bradykinesia and postural tremors. Furthermore, we propose a multi-domain learning method to predict the patient-level PD severity through task-assembling. We show the effectiveness of TSA and task-assembling method on our PD video dataset empirically. We achieve the best MCC of 0.55 on binary task-level and 0.39 on three-class patient-level classification.**

***Index Terms*—Parkinson's disease (PD), severity classification, deep learning, transfer learning, self-attention, multi-domain learning.**

I. Introduction

PARKINSON'S disease (PD) is a chronic, progressive neurological disorder, affecting over 10 million people around the world according to the American Parkinson Disease Association (APDA) [45]. Individuals with Parkinson's disease typically present with characteristic motor symptoms, including bradykinesia (i.e. slowness of movement), rigidity (stiffness), and rest tremor [48]. These symptoms are progressive over time, subsequently leading to an increase in their severity.

At present, the Movement Disorder Society - Unified Parkinson's Disease Rating Scale (MDS-UPDRS), containing four parts: I for non-motor experiences of daily living, II for motor experiences of daily living, III for motor examination and IV for motor complications, has been widely used as a validated tool to quantify PD severity [20], [33]. MDS-UPDRS is the revised and more comprehensive version of the original UPDRS [17] and they are highly correlated on the motor sections [34]. This study uses the MDS-UPDRS part III (MDS-UPDRS-III) as the measurement for analysis, which contains 18 tasks and 33 scores, with some tasks pertaining to either left or right extremities. Each task, tied to a symptom, has five responses linked to symptom-severity: 0-normal, 1-slight, 2-mild, 3-moderate, and 4-severe, providing consistency across tasks. The clinical scores are assessed by a single examinator, that is either a nurse specialized in Parkinson's Disease or a physician. Both have the certification to rate the MDS-UPDRS III. Collapsing all the scores to provide the patient with a composite total score is not recommended by [20] but can still be applicable given the minimal clinically important difference threshold values [32] and is often used in clinical practice to monitor disease progression. Although MDS-UPDRS-III is currently the gold standard to quantify the severity, it still has the potential to cause less reliable ratings due to the intrinsic inter-rater variability caused by the non-identical inter-rater protocols and inexperienced examiners [16], [51]. Besides, the presence of the specialist is mandatory when giving the rating decisions. These difficulties make the manual rating inefficient and urge for automatic quantification method. In this work, we propose a deep learning based PD severity quantification approach using videos. Fig. 1 shows the overall pipeline.

The goal of PD severity quantification is that, given an individual patient's video performing a specific task, the corresponding severity level can be predicted by the machine learning algorithm to assist ratings of examiners. As the task performed by the patient in the video is a kind of action, we naturally think of the human action recognition method to solve the identification of Parkinson's severity. Recently, many action recognition

Manuscript received October 7, 2020; revised February 1, 2021, April 3, 2021, and July 3, 2021; accepted July 16, 2021. Date of publication July 26, 2021; date of current version March 7, 2022. This work was supported by NWO and LIACS. *(Zhao Yin and Victor J. Geraedts contributed equally to this work.) (Corresponding author: Zhao Yin.)*

Zhao Yin, Ziqi Wang, and Jan van Gemert are with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2600 Delft, The Netherlands (e-mail: joy_yin@outlook.com; z.wang-8@tudelft.nl; j.c.vangemert@tudelft.nl).

Victor J. Geraedts is with the Departments of Neurology and Epidemiology, Leiden University Medical Centre, Leiden, The Netherlands (e-mail: v.j.geraedts@lumc.nl).

Maria Fiorella Contarino is with the Department of Neurology, Leiden University Medical Centre, Leiden, The Netherlands and also with the Department of Neurology, Haga Teaching Hospital, The Netherlands (e-mail: m.f.contarino@lumc.nl).

Hamdi Dibeklioglu is with the Department of Computer Engineering, Bilkent University, Ankara, Turkey (e-mail: dibeklioglu@cs.bilkent.edu.tr).

Digital Object Identifier 10.1109/JBHI.2021.3099816

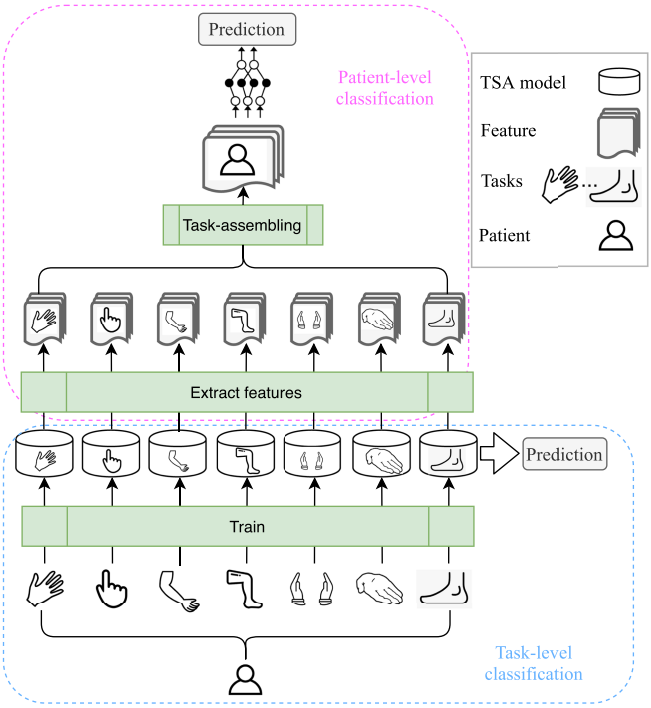


Fig. 1. The flowchart of the automatic PD severity quantification. The task symbols from left to right denote task *finger tapping*, *hand movements*, *kinetic tremor*, *leg agility*, *postural tremor*, *pronation*, and *toe tapping*.

architectures [5], [18], [21] achieved promising performance on public human action datasets and one of the mostly used architecture is the inflated 3D CNN (I3D) [5], which is a 3D CNN with 3D kernels inflated from a 2D CNN with an additional temporal dimension. Therefore, we opt to use I3D as the base model for this work.

Due to the small size of our PD dataset, directly training I3D from scratch is inefficient and prone to overfitting; thus, we use transfer learning to pre-train the network on large datasets to make the training process more stable. However, public datasets we pre-train on have noticeable motion differences while the motion difference in our PD dataset is subtle. Such large domain discrepancy makes it difficult to transfer knowledge between domains, so we need a solution to focus on exploring the temporal motion changes. Besides, the video in our dataset is a repeating task with periodic actions, where the model should learn the repeating frequency or the starting and ending point. Thus, we need another solution to assign different weights for the frames of the video. Additionally, as stated in [39], [41], not all frames are equally crucial for action recognition, so we propose to use temporal self-attention to assign the weights for frames as well as solve the domain discrepancy issue. The benefit is not only for such a repeating dataset but also for other datasets because it holds for other datasets as well that not all frames are equally important.

Once we can predict each task's severity, each patient will have a separate severity score for each task. However, it is more clinically interesting to give a summary severity for the patient rather than multiple ones, so we propose to apply a novel task-assembling method to combine the predictions of different tasks from the patient to predict a single score.

The contributions of this work are:

1) we perform automatic task-level PD severity classification using I3D from videos of our PD dataset, based on seven tasks in MDS-UPDRS-III;
2) we show that I3D can benefit from non-medical datasets with transfer learning;
3) we propose TSA to focus on the temporal visual clues and overcome the large discrepancy of motion difference between non-medical datasets and our PD dataset during transfer learning;
4) we propose a task-assembling method to combine models of different tasks to produce a single concluding severity score for a patient.

II. Related Work

A. Machine/Deep Learning Based Approaches

Machine/deep learning based PD motor assessment and analysis has been intensively researched in recent years. For instance, the K-nearest neighbors (KNN) AdaBoost classifier and support vector machines (SVM) with RBF kernel were used to classify between PD patients and controls based on the features extracted from individual handwriting [15]. Butt *et al.* [4] applied machine learning based methods to investigate the significance of PD motor features. For signal-based analysis, signals acquired from the gyroscope attached to the subject's finger were extracted to feed into multiple classifiers [49]. In [2], glottal flow features were used as input for SVM classifier to detect PD with an accuracy of 75.3%. Ferraris *et al.* [19] used data from optial RGB-Depth devices, which tracks hands and body movements, to train classifiers for PD motor severity rating. Apart from the signal-based analysis, the video was also used as an input data type for PD quantification [54], [59]. Lu *et al.* [30] designed a pose-based estimation system for assessing Parkinson's disease motor severity. However, to the best of our knowledge, apart from [46] in which freezing of gait videos were used to feed the 3D network, most researchers extracted the feature from videos as the final input for classifiers without fully utilizing the video resource. Based on machine/deep learning approaches, our work applies action recognition method to quantify PD severity using RGB video data.

B. Transfer Learning

Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [58]. It is widely used as a pre-training approach to offer the model a better starting point instead of training from scratch. In the work of [36], CNN layers trained from ImageNet is reused to transfer visual recognition tasks to learn mid-level representations for small datasets. In action recognition, researchers apply transfer learning to pre-train the model on a large dataset to make the training process faster, more efficient, and less prone to overfitting with a significant performance improvement [5], [21]. Most

related research shows that transfer learning can be a useful tool to make the network work on small datasets, and thus we use transfer learning in this work to help improve the performance on our PD dataset.

C. Capture Temporal Information

1) For General Video Dataset: In action recognition, researchers apply various methods to capture the temporal information crucial in video data. In the work of [50] (*C3D*), 3D CNN is used as a spatiotemporal feature extractor for videos, and the extracted features are used as inputs for simple linear classifiers. Based on the 3D CNN, an I3D is introduced to take advantage of pre-trained 2D models [5]. Similar to 3D CNN, I3D performs 3D convolution on both spatial and temporal dimensions simultaneously. However, in I3D, pre-trained 2D filters are repeated or inflated multiple times to form 3D filters. Therefore, I3D can benefit from successful image (2D) classification models trained on large datasets such as ImageNet [12]. Besides 3D CNN, a combination of a stack of CNNs and Long Short-Term Memory (LSTM [23]) networks is applied to exploit the temporal information [1], [13] as well. These methods apply either 3D CNN or 2D CNN with fusion methods such as LSTM on the video data to capture the temporal information. We use I3D as our base model because of its decent performance on public datasets, including Kinetics-400 experimented in [21].

2) For Periodic- and Subtle-Motion Video Dataset: The spatiotemporal template of motion features is used to recognize and segment the repetitive motion by template matching [38]. In [10], CNN is used to count the number of repetitions, and circle length in periodic-motion videos. Besides the task of action recognition, the estimation of repeating frequency is studied in [37], using a Lagrangian approach and an Eulerian approach as the frequency estimators. In periodic-motion videos, we need to focus on the repeating frequency, starting, and ending points to make the model work.

In medical datasets such as movement disorder dataset, videos usually have subtle motion changes, which are hard for architectures to work because subtle motion information is difficult to capture and can not even be seen with bare eyes. The subtle motions can be magnified using a steerable pyramid [28], [53]. In the work of [11], motion frequency is used to estimate material properties. Similarly, signal analysis in the Fourier domain is employed to estimate the tremor frequency of subtle motions [37]. In subtle-motion videos, we need to focus on magnifying the subtle motion or directly estimating the frequency.

D. Self-Attention

Attention module is widely used in natural language processing [7] and computer vision [43], [56] fields by allowing the network to focus on key words or pixels. Self-attention mechanism is proposed to capture the relative relationship between words or pixels. Self-attention is extensively explored since the Transformer network is introduced for machine translation [52] where the self-attention is used to compute the interactions between words. In recent work, the QANet [60] architecture uses self-attention in cooperation with convolutions for machine-reading and question answering tasks, where the convolution computes local interactions and self-attention computes global interactions. In image tasks, self-attention with relative positional embeddings is usually used to compute the interactions among pixels in the same image and allows the model to learn which part of the image is of more importance [3]. In the non-local network [55], self-attention can be used in convolutional architectures to learn the long-range interactions among pixels in images or videos for object detection and video classification. In general, self-attention is used in architectures for modeling sequences as it can capture long-distance interactions. In this paper, we propose a new method, temporal self-attention model, for PD quantification, which involves I3D and the self-attention mechanism, attempting to detect the periodic and subtle motion in the video data.

E. Multi-Domain Learning

Different non-i.i.d. Parkinson tasks can be treated in a multi-domain setting [14], [29], [57] with each task being one domain. Multiple similar domains can be learned to let the model work on a new target domain using parameter combination from multiple classifiers [26]. In [6], perceptron-based algorithms are employed for multi-task binary classification problem with the similarity estimation among tasks. Multi-domain learning aims at exploring the relationship between tasks or domains and integrating them to solve a common task. In this work, we combine the features from multiple domains (i.e., tasks from MDS-UPDRS-III) to predict patient-level PD severity classification.

III. Methods

The overall flow of the algorithm is described as follows. Initially, each video is preprocessed to have the same spatial and temporal size. At the same time, we use network-based transfer learning to transfer knowledge from non-medical datasets to the medical one, i.e., reusing the network trained on large datasets as the pre-trained model to replace model initialization. Then, the pre-trained model is fine-tuned on the collected Parkinson's dataset to learn the underlying patterns. After fine-tuning, the model can be used as the classifier for task-level classification. By combining the features extracted by the deep models from different tasks and training a shallow neural network using those features, patient-level analysis can be further made.

A. Inflated 3D Convolutional Neural Network (I3D)

In this paper, we use I3D as the base network with Residual Networks (ResNet) as the backbone (currently 18, 34, 50, 101, 152-layer variations are available) and its pre-trained models are already available [21]. Furthermore, rather than using two streams (RGB frames and optical flow), we use RGB frames as the only input because computing optical flow is time-consuming, which is not feasible if the real-time prediction is required.

The model is optimized using gradient descent by minimizing the empirical loss with class-balanced focal loss [9]:

$$J(\omega) = \frac{1}{N}\sum_{i=1}^{N}\left(-\frac{1-\beta}{1-\beta^{n_y}}\sum_{c=1}^{C}(1-p_{i,c}^{t})^{\gamma}\log(p_{i,c}^{t})\right) + \lambda\left\|\omega\right\|_2^2, \tag{1}$$

where C, N, ω and γ denote the number of classes, number of samples, learned parameters and *focusing* parameter, and $\beta = (N-1)/N$. n_y stands for the number of samples in the ground-truth class y and p^t is defined as

$$p^t = \begin{cases} p & \text{if } y = c \\ 1-p & \text{otherwise.} \end{cases} \tag{2}$$

B. Self-Attention Replacing Convolution

We describe the proposed temporal self-attention block for video classification following the symbol styles of [3].

1) Temporal Self-Attention Over Video Volume: We first transpose and flatten the input of shape (C, T, H, W)[1] from the previous layer to the shape of $HW \times T \times C$ and then perform multi-head-attention on the temporal dimension

$$O_h = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k^h}}\right)V, \tag{3}$$

where queries $Q = XW_q$, keys $K = XW_k$ and values $V = XW_v$ and W_q, $W_k \in \mathbb{R}^{C\times d_k^h}$ and $W_v \in \mathbb{R}^{C\times d_v^h}$ are learned linear transformations.[2] d_k^h and d_v^h stand for the dimension of each head of K and V. Note that we transpose the last two dimensions of V to correctly multiply with Q. Concatenating the outputs from all heads we get

$$O = [O_1, \ldots, O_{N_h}]. \tag{4}$$

The shape of O is $(HW \times T \times d_k^h)$ and is transformed with $W^O \in \mathbb{R}^{d_v\times d_v}$ to

$$\text{MultiHead}(Q, K, V) = OW^O, \tag{5}$$

where $\text{MultiHead}(Q, K, V)$ is of shape $(HW \times T \times d_v^h)$. After reshaping back to the original spatial and temporal dimension, we have the final output $\text{MultiHead}(Q, K, V) \in \mathbb{R}^{T\times H\times W\times d_v}$ of our temporal self-attention block if relative postional embeddings [3] (see Section III-B2) not applied.

The novelty of our temporal self-attention block is applying the self-attention mechanism solely on the temporal dimension, leaving the spatial dimension untouched. The advantage is that self-attention can capture the long-range temporal changes while keeping standard CNN there, capturing the necessary visual patterns simultaneously. As such, the abilities of both self-attention and CNN be retained and incorporated in the temporal self-attention block, which effectively makes up the drawback of I3D.

Fig. 2 illustrates the temporal self-attention mechanism. The temporal sequence of feature points (red ones) that share the same spatial position is the atomic unit, on top of which the temporal self-attention applies. We have HW sequences/units located at all spatial positions, and each of them is independent of others when performing the temporal self-attention.

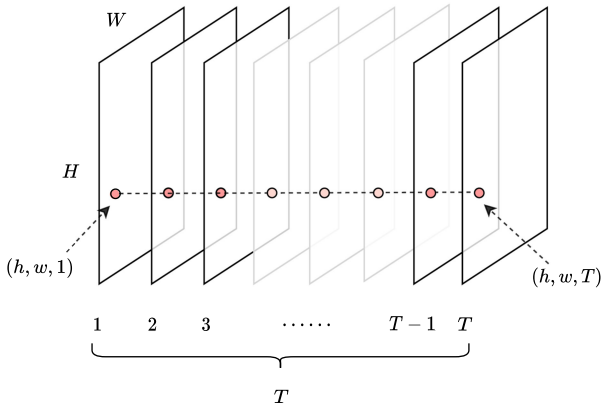


Fig. 2. An example of temporal self-attention. Assume the stack of those rectangles is a feature map (or more intuitively for 3D data, feature volume) from one channel. Each rectangle represents the spatial visual patterns at a specific temporal position. Our temporal self-attention is performed on the feature points colored in red, which share the same spatial position along the temporal dimension. It can be seen as self-attention through time.

2) Relative Positional Embeddings: The only difference between 1D and 2D relative positional embeddings is the dimensions involved in the algorithm. Thus we refer to [3] for the details of 2D relative positional embeddings, and we do not discuss the 1D variation anymore in this paper. To implement temporal relative self-attention, we add relative temporal information to the temporal self-attention block's output. The output is now changed from (3) to

$$O_h = \text{Softmax}\left(\frac{QK^T + S_T^{rel}}{\sqrt{d_k^h}}\right)V, \tag{6}$$

where $S_T^{rel} \in \mathbb{R}^{HW\times T\times T}$ is the matrix of relative position logits along the temporal dimension.

3) Temporal Relative Self-Attention: We combine temporal self-attention with 1D relative positional embeddings to form our new building block-temporal relative self-attention block. Fig. 3 describes the whole pipeline of the proposed block.

4) Temporal Relative Self-Attention Network (TSA): Once the temporal relative self-attention block is built up, the convolutional block in any architecture can be substituted. Take 3D ResNet-34 for instance, which has 33 convolutional layers. We replace as many layers as possible with our block from the last convolutional layer to the first one until we hit the memory bottleneck.

The time complexity of our block is $O(HWT^2d_k)$ compared to the convolutional block $O(HWTC)$, which is time-efficient since the temporal size is typically small after a few layers. The memory cost is $O(HWT^2N_hd_k^h)$ compared to the convolutional block $O(HWTC)$.

[1]The number of channels, time or frames, height and width.
[2]Bias terms are ignored when we mention linear transformations.

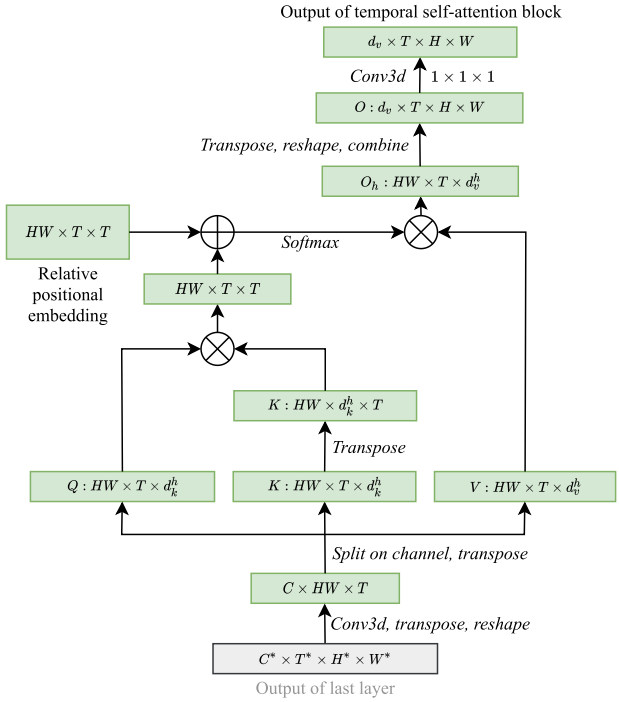


Fig. 3. The general pipeline of our temporal relative self-attention. Rectangles in the workflow represent tensors with shape specified, and italic words stand for tensor operations. $\otimes$ and $+$ denote tensor product and addition.

C. Multi-Task Assembling

Using the model we discussed in previous sections, it can solve the task-level severity classification on our PD dataset. Given a sample related to a specific task from the dataset, we can predict its task severity S_t. Nonetheless, it is more clinically interesting to tell the severity score of a patient S_p instead of tasks. Therefore, we propose two multi-task assembling methods to combine the tasks to do severity classification for patients. Note that the following methods require trained models on the PD dataset for task-level classification.

1) Vector Averaging and Vector Weighting: We use the trained model as a feature extractor to compress the information of a video into a dense one. We first extract the flattened vector $F \in \mathbb{R}^d$ of dimension size d as the compressed information, which is the input feature of the fully connected layer. Each video, containing only a single task from a patient, produces one feature vector F_m of task m and all videos from that patient produce feature vectors $F_M \in \mathbb{R}^{d \times M}$ of all M tasks. Different tasks may contribute unequally to a patient's severity score, so we use two strategies to convert (or combine) F_M into a vector $F \in \mathbb{R}^d$, representing the feature of a patient.

The first approach is to average features, formulated as

$$F = \frac{1}{M} \sum_{m=1}^{M} F_m, \tag{7}$$

by assuming each feature (task) contributes equally. The second approach is to take the weighted average of features as the following

$$F = \sum_{m=1}^{M} \alpha_m F_m, \tag{8}$$

where α_m ($\sum_{m=1}^{M} \alpha_m = 1$) is the learnable weight for task m. The first approach is a special case of this one. Afterward, F is fed as input to train a shallow neural network.[3] The network is optimized using gradient descent by minimizing the empirical loss $J(\omega)$ (see 1) where N is the number of patients.

2) Attention-Based Feature Weighting: In the feature averaging and weighting approach, we assume task weights are identical across all patients. However, patients may not share the same task weights so that the global task weights may be insufficient and inaccurate. Therefore, we propose to use channel-wise attention-based weighting, which automatically assigns task weights for each patient separately. To do so, we use another feature map $F_M \in \mathbb{R}^{M \times C \times T \times H \times W}$ (M denotes the number of tasks), the output of the last convolutional or our self-attention layer, as the extracted feature for a video.

The first weighting strategy is to apply squeeze-and-excitation block [24] to map the input feature F_M to a set of channel weights. As the task weights are our concerns instead of the channels, we take the task dimension as the channel dimension in the squeeze-and-excitation block. The process can be formulated as follows. First, squeeze global information into a task descriptor by using global average pooling to generate task-wise statistics

$$z_m = \frac{1}{C \times T \times H \times W} \sum_{c=1}^{C} \sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{w=1}^{W} F_m(c, t, h, w), \tag{9}$$

where F_m denotes the feature map for task m. Then we excite the task-wise statistics to task weights ($W_1 \in \mathbb{R}^{\frac{M}{r} \times M}$, $W_2 \in \mathbb{R}^{M \times \frac{M}{r}}$ in which r is the dimensionality-reduction ratio)

$$\alpha_M = \sigma(W_2 \delta(W_1 z_M)), \tag{10}$$

where α_M, σ and δ denote task weights, the sigmoid activation and the ReLU [35] function. Finally we obtain the combined feature map $F \in \mathbb{R}^{C \times T \times H \times W}$

$$F = \alpha_M F_M. \tag{11}$$

Applying the squeeze-and-excitation block to get task weights is rather simple but turns out to be efficient. It flexibly generates different weights for different patients accordingly. However, this approach assumes each feature point in the feature map contributes equally, which means a task weight is a global weight for all feature points. We can explore even further by making each feature point having its own weight $\alpha_{t,h,w,m}$, which brings about the pixel-wise attention-based weighting approach.

We opt to use the self-attention mechanism similar to our temporal relative self-attention block for pixel-wise weighting, by applying it on the task dimension instead of the temporal dimension. First, we reshape and flatten $F_M \in \mathbb{R}^{M \times C \times T \times H \times W}$ into the shape of $(THW \times M \times C)$ and then the output of a

[3] 0, 1 or 2 hidden layers with non-linear activation.

TABLE I
THE SUMMARY OF FOUR TASK-ASSEMBLING METHODS

	vector averaging	vector weighting	channel-wise attention weighting	pixel-wise attention weighting
Input type	*avgpool*	*avgpool*	*layer4*	*layer4*
Weights differ among tasks	✗	✓	✓	✓
Weights differ among patients	✗	✗	✓	✓
Weights differ among feature points	✗	✗	✗	✓
Core mechanism	averaging	learnable weight vector	squeeze-and-excitation [24]	self-attention

single attention head can be computed as

$$O_h = \text{Softmax}\left(\frac{(F_M W_q)(F_M W_k)^T}{\sqrt{d_k^h}}\right)(F_M W_v), \qquad (12)$$

where W_q, $W_k \in \mathbb{R}^{C\times d_k^h}$ and $W_v \in \mathbb{R}^{C\times d_v^h}$ are learned linear transformations. Afterwards, we combine attention results of all heads and project using $O^W \in \mathbb{R}^{d_v\times d_v}$ to form the task weighted feature map

$$F = [O_1, \ldots, O_{N_h}]\, O^W. \qquad (13)$$

Note that the task weights for each feature point $\alpha_{t,h,w,m}$ is implicitly embedded in the computation of attention output.

Task weighted features using both approaches are fed into a shallow neural network consisting of batch normalization [25], the ReLU function, global average pooling, and a fully connected layer.

The summary of the proposed four task-assembling methods can be found in Table I. Vector averaging and vector weighting use the outputs of the last global average pooling layer while attention-based weighting methods use the outputs of the last convolutional/self-attention layer in the network. We denote *avgpool* and *layer4* as the feature types.

IV. EXPERIMENTAL SETTINGS

A. Dataset

In this paper, we introduce a new video dataset for Parkinson's disease analysis. We develop this dataset principally because there is a lack of such datasets for Parkinson's disease analysis. We believe that having one will facilitate research in this area because the dataset simulates the procedure of how experts assess patients' symptoms using MDS-UPDRS-III scores. Besides, the dataset is challenging enough to act as a performance benchmark where the advantages of different architectures can be demonstrated.

1) Data Collection: Routine video recordings of consecutive patients who underwent either a Levodopa Challenge Test (LCT [40], [42]) prior to DBS surgery, or underwent a Stimulator Challenge Test (SCT, [8], [22]) after DBS surgery, were collected. All patients fulfilled the criteria for idiopathic PD. Patients who underwent a LCT were videotaped twice (i.e. Med-OFF and Med-ON); patients who underwent SCT were videotaped three times (Med-OFF/Stim-ON [31], etc). Video recordings were made with the camera in a fixed position, with a complete overview of the patient central on the screen. Due to the varying nature of the examination room, the camera's position and angle towards the patient varied, as well as the background and surroundings. During the MDS-UPDRS-III examination, the zoom-function was occasionally used to focus on the hands or feet.

All videos were made in one continuous recording of the examination. Separate segments were created by clipping the videos per task (left and right separately if required): bradykinesia of the hands (MDS-UPDRS-III items 3.4, 3.5, 3.6), bradykinesia of the legs (items 3.7, 3.8), postural tremor (item 3.15), kinetic tremor (item 3.16). Rigidity was not included as this symptom is not assessed through visual observation; global bradykinesia, speech, freezing-of-gait, and rest-tremor were not included as no specific video-segment pertained to those tasks and they were evaluated throughout the entire recording. The local medical ethics committee waived the formal evaluation of the study. All patients gave written informed consent.

We are not allowed to make the dataset publicly under the Dutch privacy law.

2) Dataset Overview: The dataset contains 39 subjects (all patients) and 1082 video fragments after cutting. Each sample in the dataset is of resolution 1920 by 1080 and 25 fps. The duration of samples may be different on different tasks. Fig. 4(a) shows the duration distribution of our dataset.

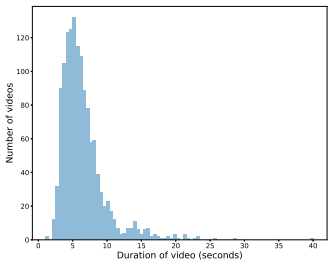
The dataset contains $T = 11$ tasks for most of the patients based on the MDS-UPDRS-III, namely finger tapping, gait freezing, hand movements, leg agility, pronation, toe tapping, arising from chair, kinetic tremor, postural tremor, postural stability and rest tremor. Note that not all tasks are used in the experiments. Each video has a task-level severity score $S_t \in \{0, 1, 2, 3, 4\}$ (0: normal, 1: slight, 2: mild, 3: moderate and 4: severe) labeled by experts. We have to emphasize that a task score of 0 does not mean that the subject is not a PD patient but indicates that the subject may have low severity on the specific task. Each patient has a patient-level severity score, which is the sum of all task-level severity scores, as shown in the following equation:

$$S_p = \sum_{t=1}^{T} S_t. \qquad (14)$$

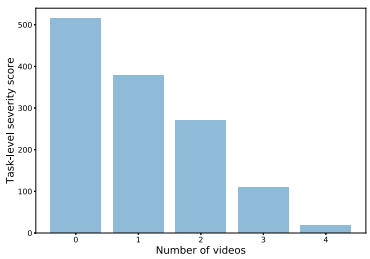
The distributions of S_t (over all tasks) and S_p are shown in Fig. 4(b) and Fig. 4(c).

B. Settings

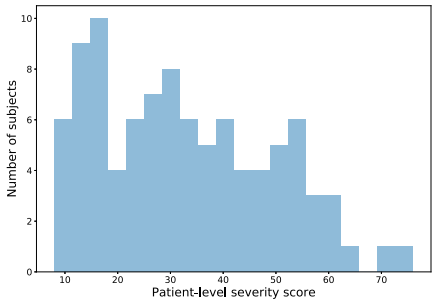
To evaluate our methods for Parkinson's severity classification, we use the above-described dataset. In our experiments, only RGB frames are used as the input for the deep architectures. The clips are resized to $32 \times 224 \times 224$ resolution without changing their spatial aspect ratios.



(a) The histogram of the duration of samples, using 80 bins. The average duration is 6.3 seconds, and 90% of samples are shorter than 10 seconds, with less than five samples longer than 25 seconds.

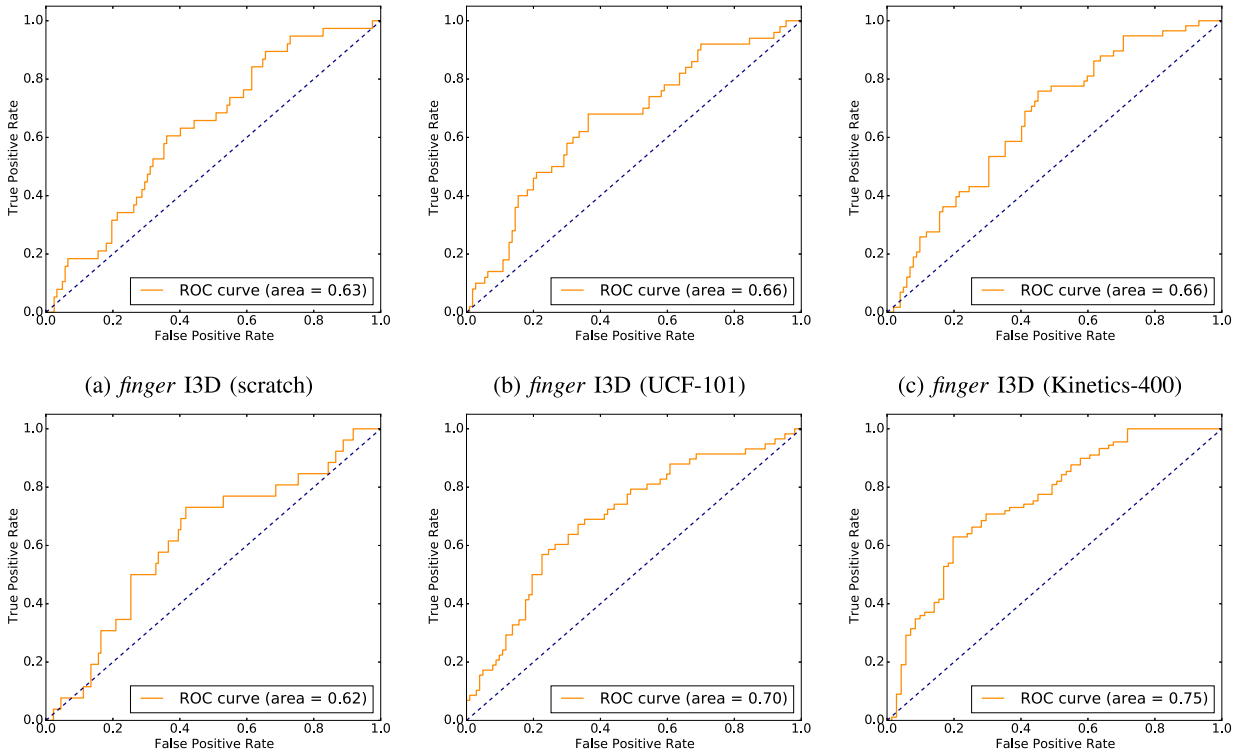


(b) The bar chart shows the distribution of task-level severity score. From low to high severity class, the number of samples decreases, which shows the class imbalance issue in our dataset.



(c) The histogram of patient-level severity score using 20 bins. Compared to task-level severity distribution, patient-level severity distribution has no obvious imbalance issue. The number of patients across the range of severity is approximately on the same level.

Fig. 4. Distributions of the sample duration and task/patient-level severity of our dataset.



(a) *finger* I3D (scratch) (b) *finger* I3D (UCF-101) (c) *finger* I3D (Kinetics-400)

(d) *hand* I3D (scratch) (e) *hand* I3D (UCF-101) (f) *hand* I3D (Kinetics-400)

Fig. 5. ROC curves of all 6 settings in Table III.

The dataset is split into five folds at the patient level but not the video level. One subject only appears in either the training or testing fold to avoid network cheating by recognizing the appearance of the patient. We train networks on four of them and test it on the remaining one in the cross validation setting. The overall accuracy is obtained by taking the average of the individual accuracy tested on each fold.

I3D is pre-trained on both UCF-101 (by ourselves) and Kinetics-400 (by [21]). TSA is pre-trained only from UCF-101 (by ourselves). Batch size of 15, learning rate of 0.001 without decay and weight decay (λ) of 0.01 are used.

The task-level score $S_t \in 0, 1, 2, 3, 4$ is split into two classes: class 0 for $\{0, 1\}$ and class 1 for $\{2, 3, 4\}$ since we are more interested in whether the model can distinguish between the slight and severe group of patients. The patient-level score S_p is split into three classes in the way that each class has an equal number of patients. Method specific settings are provided alongside when showing the results in Section V.

TABLE II
TOP-1 ACCURACY ON UCF-101 AND HMDB-51. ALL ACCURACY ARE AVERAGED OVER THREE SPLITS. BOTH METHODS USE RESNET-18 AS THE BACKBONE. TSA SHOWS BETTER PERFORMANCE ON BOTH DATASETS SO THAT IT CAN BE FURTHER APPLIED TO PD DATASET

Method (scratch)	UCF-101	HMDB-51
I3D ResNet-18 [21]	42.4	17.1
TSA ResNet-18	51.5±2.6	22.1±1.9

We briefly introduce the results in order shown in the next section. We first validate the performance of TSA on public dataset in section V-A, and then inspect the performance improvement using transfer learning in section V-B. In Section V-C, we show results on seven PD tasks using models with different settings followed with comparison between those models. In Section V-D, we analyze the performance on patient-level severity classification, compare different strategies to combine PD tasks, and show the model behavior on classifying only the highest and lowest severity class.

V. RESULTS

In this section, we show the results of our experiments. We test seven tasks with high quality videos, *finger tapping*, *hand movements*, *pronation*, *toe tapping*, *leg agility*, *postural tremor* and *kinetic tremor*. They are denoted as *finger*, *hand*, *pronation*, *toe*, *leg*, *postural* and *kinetic* for simplicity. We use ResNet-34 as backbone because through experiments we find that ResNet-34 is the most suitable one in this study, considering the size and difficulty of our dataset. One can of course use other backbones if the size, complexity and classes of the dataset are different from ours. We have to emphasize that, in all experiments, although patients contribute more than one video, no patient is included into both the training- and test-set because even though videos of a patient are separate ones, they are still from the same patient.

A. Validate Temporal Relative Self-Attention Network

Before applying TSA on PD dataset, we first check whether it works better than I3D on two frequently used public datasets UCF-101 and HMDB-51. Hyper-parameters are chosen without optimization: input shape of $64 \times 224 \times 224$, *lr* of 0.001, batch size of 45, weight decay of 10^{-5} and optimizer of SGD with momentum [47]. The backbone is ResNet-18 for fast illustration. Table II shows that TSA outperforms I3D when both trained from scratch. The performance improvements demonstrate the effectiveness of TSA and the possibility of applying it to our PD dataset.

B. Benefit From Transfer Learning

We utilize three datasets: Kinetics-400 [27] and UCF-101 [44] to pre-train our models considering their large sizes, high quality and popularity. Then, we fine-tune the pretrained models on our PD dataset. Since our dataset contains periodic and subtle motions while public datasets have easily distinguishable motions, the relatedness between our dataset and public datasets is not tight. As such, the parameters from the convolutional stem may not be optimal after transferring to our dataset. Thus all layers of the model rather than part of them are fine-tuned.

TABLE III
ACCURACY, PRECISONS, RECALL AND MCC (WITH CI 95% AND *P*-VALUE) ON TASK *FINGER* AND *HAND* (BINARY CLASSIFICATION) USING I3D WITH AND WITHOUT TRANSFER LEARNING. DATASETS IN THE BRACKETS DENOTE WHERE THE MODEL IS PRETRAINED. I3D USING TRANSFER LEARNING ACHIEVES BETTER RESULTS THAN I3D TRAINED FROM SCRATCH ON BOTH *FINGER* AND *HAND* TASKS. MOREOVER, TRANSFER LEARNING WITH A LARGER DATASET (I.E., KINETICS-400) HAS MORE BENEFITS TO THE MODEL

Method	Metric	*finger*	*hand*
I3D (scratch)	acc	65.4	65.6
	MCC	0.32±0.08	0.31±0.06
	CI 95%	[0.16, 0.48]	[0.19, 0.43]
	p-value	7.3×10^{-5}	3.7×10^{-7}
I3D (UCF-101)	acc	68.6	70.0
	MCC	0.34±0.10	0.39±0.05
	CI 95%	[0.14, 0.54]	[0.29, 0.49]
	p-value	7.1×10^{-4}	3.8×10^{-14}
I3D (Kinetics-400)	acc	69.2	77.5
	MCC	0.35±0.06	0.54±0.07
	CI 95%	[0.23, 0.47]	[0.40, 0.68]
	p-value	1.1×10^{-8}	7.0×10^{-14}

TABLE IV
THE NUMBER OF SAMPLES IN EACH CLASS OF SEVEN TASKS IN OUR PD DATASET

Task	*finger*	*hand*	*kinetic*	*leg*	*postural*	*pronation*	*toe*
Class 0	66	89	130	145	62	104	87
Class 1	91	71	38	39	23	72	71

I3D and task *finger* and *hand* are used to demonstrate the function of transfer learning. Convergence is confirmed for every compared setting for a fair comparison. Note that for task-level classification we have binary classes. In Table III, I3D trained from scratch, I3D pretrained from UCF-101, and I3D pretrained from Kinetics-400 are compared based on the binary accuracy, precision, recall, and Matthews correlation coefficient (MCC). Here the MCC is formed as:

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \tag{15}$$

where TP, TN, FP and FN stand for true positive, true negative, false positive and false negative. We also show the receiver operating characteristic (ROC) curves of all 6 settings based on Table III. In general, I3D pretrained from the two datasets outperform I3D (scratch), demonstrating that I3D can benefit from non-medical datasets with transfer learning. Moreover, the performance improvement of I3D (Kinetics-400) from I3D (scratch) is more notable than I3D (UCF-101) especially on task *hand*, which indicates the model would benefit more from a larger dataset with transfer learning.

C. Task-Level Severity Classification

Building a model good at predicting the task severity score is our first concern and affects the later experiments and research. Two architectures - I3D and our TSA are compared in Table V

TABLE V
ACCURACY, PRECISION, RECALL, AND MCC (WITH CI 95% AND *P*-VALUE) ON SEVEN TASKS FROM MDS-UPDRS-III USING I3D AND TSA. EACH ROW SHOWS THE PERFORMANCE OF A TASK AND EACH COLUMN GIVES THE RESULT OF A MEASUREMENT (TWO CLASSES). DATASETS IN THE BRACKETS DENOTE ON WHICH PUBLIC DATASET THE MODEL IS PRETRAINED IN GENERAL, I3D PRETRAINED ON KINETICS-400 OUTPERFORMS I3D PRETRAINED ON UCF-101, INDICATING TRANSFER LEARNING FROM LARGER DATASETS HAS MORE BENEFITS THAN SMALLER DATASETS. TSA PRETRAINED FROM A SMALLER DATASET, UCF-101, IS COMPARABLE TO KINETICS-400 PRETRAINED I3D

Task	I3D (UCF-101)				I3D (Kinetics-400)				TSA (UCF-101)			
	acc	precision recall	MCC	95% CI *p*-value	acc	precision recall	MCC	95% CI *p*-value	acc	precision recall	MCC	95% CI *p*-value
finger	68.6	0.55 0.76	0.34±0.09	[0.16, 0.52] 1.7×10^{-4}	69.2	0.57 0.76	0.35±0.06	[0.23, 0.47] 1.1×10^{-8}	78.2	0.75 0.81	**0.55±0.08**	[0.39, 0.71] 2.1×10^{-11}
hand	70.0	0.76 0.61	0.39±0.07	[0.25, 0.53] 4.5×10^{-8}	77.5	0.80 0.75	**0.54±0.11**	[0.32, 0.56] 1.3×10^{-6}	75.6	0.79 0.72	0.50±0.06	[0.38, 0.62] 7.2×10^{-16}
kinetic	78.0	0.87 0.10	0.22±0.06	[0.10, 0.34] 2.7×10^{-4}	73.8	0.82 0.49	0.33±0.10	[0.13, 0.53] 1.0×10^{-3}	79.2	0.87 0.51	**0.40±0.09**	[0.22, 0.58] 1.1×10^{-5}
leg	79.3	0.88 0.17	0.24±0.05	[0.14, 0.34] 2.2×10^{-6}	79.3	0.88 0.14	0.26±0.06	[0.14, 0.38] 1.8×10^{-5}	70.1	0.81 0.35	**0.29±0.04**	[0.21, 0.37] 1.8×10^{-12}
postural	74.1	0.85 0.08	0.18±0.04	[0.10, 0.26] 8.7×10^{-6}	77.6	0.87 0.34	0.30±0.08	[0.14, 0.46] 2.0×10^{-4}	70.6	0.78 0.56	**0.35±0.09**	[0.17, 0.53] 1.1×10^{-4}
pronation	68.8	0.76 0.56	0.34±0.06	[0.22, 0.46] 2.7×10^{-8}	77.8	0.87 0.71	**0.53±0.07**	[0.39, 0.67] 1.7×10^{-4}	72.2	0.76 0.67	0.43±0.04	[0.35, 0.51] 5.9×10^{-25}
toe	64.6	0.72 0.52	0.31±0.07	[0.20, 0.48] 1.7×10^{-6}	67.7	0.70 0.65	**0.38±0.08**	[0.22, 0.54] 2.8×10^{-6}	62.0	0.68 0.53	0.29±0.06	[0.17, 0.41] 1.9×10^{-6}
average	-	-	0.29±0.08	-	-	-	0.38±0.11	-	-	-	0.40±0.10	

on seven tasks from MDS-UPDRS-III. The class distribution can be found on Table IV. In general, the class imbalance in task *finger*, *hand*, *pronation* and *toe* is acceptable. In remaining tasks, the class imbalance issue is severe. Note that we replace convolutional layers in 3D ResNet-34 *layer3* and *layer4* with temporal relative self-attention block to construct our TSA network. The dataset in the brackets denotes on which the model is pretrained. We show the MCC along with precision and recall.

1) Task-Level Performance: Fig. 6 shows the ROC curve for each task in the setting which achieves the best performance (bold numbers) in Table V. Three out of seven tasks have the best MCC higher than 0.5, and only one task *leg* is under 0.3. The average MCC across all seven tasks is 0.40, sufficiently good for classification on a medical dataset. It demonstrates that deep architectures can predict the task (i.e., task from MDS-UPDRS) severity of a patient with decent accuracy given the video from that task.

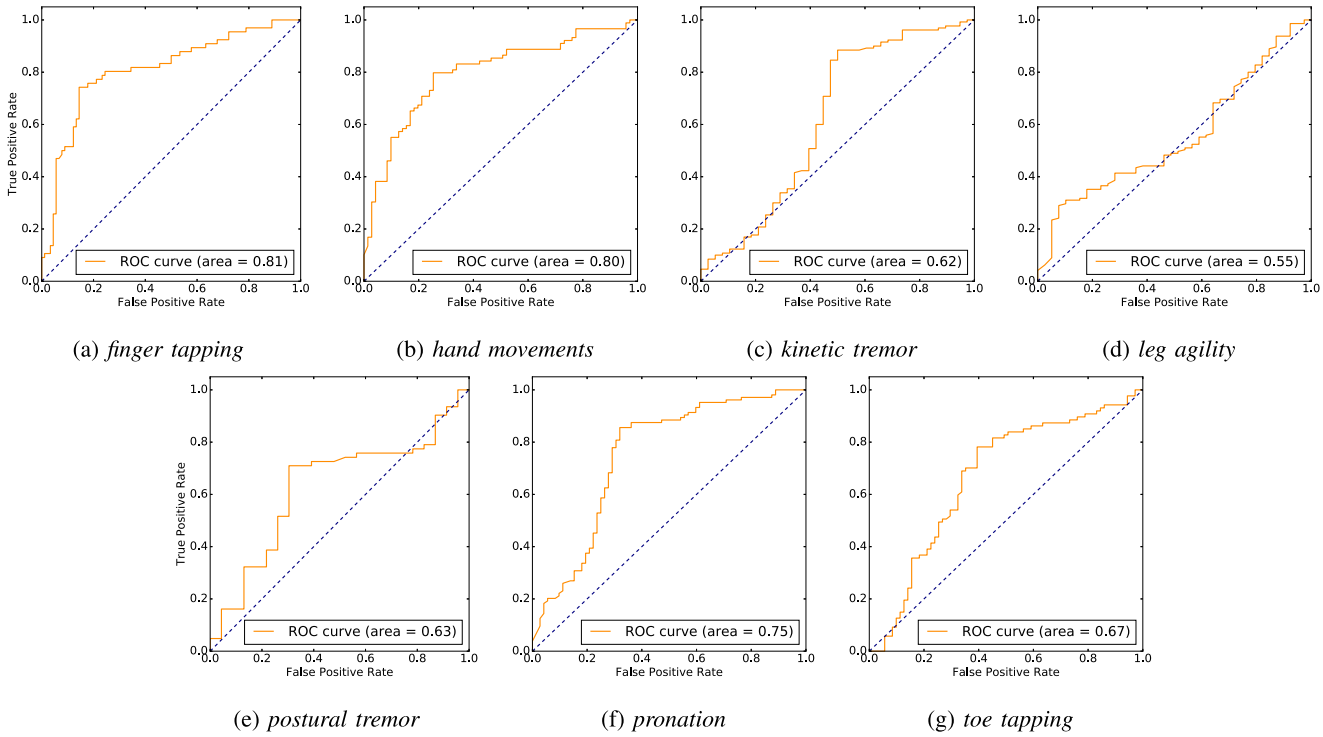
In particular, task *finger*, *hand* and *pronation* are the top-3 well-classified task in terms of MCC and ROC curves in Fig. 6(a), 6(b) and 6(f), because 1) most of the videos are zoomed in to focus on the objects, making it easier for the model to look at the relevant patterns and 2) the class imbalance problem is slight compared to task *kinetic*, *leg* and *postural*. On the opposite, task *leg* has the lowest MCC, and the ROC curve in Fig. 6(d) does not bulge towards the top-left corner of the figure, indicating a corrupt model for task *leg*. Inspecting Table V, we can observe quite low recalls of 0.17 and 0.14 using I3Ds and an inadequate recall of 0.35 using TSA.

The performance discrepancy between tasks exposes some disadvantages of our architectures. First, the ratio of objects, e.g., hand in task *hand movements* and toe in task *toe tapping*, occupying the bounding box of the video matters. In task *finger tapping*, *hand movements* and *pronation*, the zoom-function is occasionally used to focus on the objects, and most of the videos are zoomed in during the pre-processing stage, which gives the architectures cleaner and more easy-to-identify input data. Second, the effects of the class imbalance problem on the architectures cannot be ignored. Due to the PD dataset is a periodic- and subtle-motion dataset, which is different from public datasets. Identifying task severity is harder than classifying different human actions. In such a case, the extreme class imbalance can corrupt the architectures' behavior even if the class-balanced loss [9] is adopted. However, the class imbalance is everywhere in real-world settings or at least in Parkinson's disease. As such, we leave solving class imbalance on the PD dataset as one of the future work.

2) Model Comparison: In Table V, we see that in terms of the MCC, TSA (UCF-101) outperforms I3D (UCF-101) on six tasks with a significant margin. Besides, the average MCC of the former is also clearly better than the latter. Since the only difference between the two is the backbone used, we can conclude that our TSA performs better than I3D on the PD dataset.

Also, compared to I3D (Kinetics-400), TSA (UCF-101) still has 1.5% improvements even if pretrained from a much smaller and less complex dataset. It demonstrates that TSA is better at dealing with the large discrepancy of motion difference between non-medical datasets and our PD dataset. So we think TSA pretrained from Kinetics-400 would further improve the performance. Due to the limit of time and computation resource, we leave it as the future work.

Regarding the time cost of the temporal relative self-attention, it is completely acceptable as the network can still run with a bit more time cost. However, the memory cost can be problematic if



(a) *finger tapping* (b) *hand movements* (c) *kinetic tremor* (d) *leg agility*

(e) *postural tremor* (f) *pronation* (g) *toe tapping*

Fig. 6. ROC curves for seven tasks in the setting where the best performance is achieved in Table V. The ROCs on task *finger*, *hand*, *pronation* and *toe* are well shaped, indicating that models on these tasks performs well. The remaining ROCs are close to the diagonals, which means the models' performance is not good.

TABLE VI

CLINICAL INFORMATION FOR THREE CLASSES. NOTE THAT EACH PATIENT IS VIDEOTAPED TWO OR THREE TIMES, AND THE SEVERITY SCORE OF EACH TIME MAY FALL INTO DIFFERENT CLASSES. FOR SIMPLICITY, L-OFF, L-ON, A, B, AND C DENOTE LEVODOPA CHALLENGE TEST OFF, LEVODOPA CHALLENGE TEST ON, MED-OFF-STIM-ON, MED-OFF-STIM-OFF, AND MED-ON-STIM-ON. EACH CLASS HAS AN APPROXIMATELY EQUAL NUMBER OF PATIENTS AND VIDEOS, I.E., NO SEVERE CLASS IMBALANCE ISSUE

Class	Score	Number of patients	Age	Disease duration	Male/Female	Number of video fragments					
						all	L-OFF	L-ON	A	B	C
0	15±4	32	61±8	11±4	22/10	351	0	130	66	0	155
1	32±5	32	65±9	12±5	28/4	374	62	36	145	65	66
2	53±8	31	64±8	11±5	21/10	357	152	21	12	172	0
total	33±16	39	63±8	11±5	28/11	1082	214	187	223	237	221

the network is too deep due to the hardware memory limitation. As such, we give some useful solutions in terms of the algorithm itself:

1) only replace convolutional layers with small temporal size (usually the last few),
2) reduce d_k and
3) use large kernel size or stride on the temporal dimension at the first few layers to quickly decrease the temporal size to the one you want and use kernel size of 1 at following layers to maintain the temporal size unchanged until the last layer.

Another issue of TSA is that a large learning rate is possible to cause the exploding gradients problem, which can be overcome by applying approaches such as the ReLU activation function and pre-training.

D. Patient-Level Severity Classification

We use the trained model on each task as the feature extractor to extract the learned patterns and apply the proposed four task-assembling methods to incorporate tasks to produce a single concluding severity score for a patient. The patient-level severity is split into three classes by cut-off: *slight* $\in [0, 23]$, *moderate* $\in (23, 40]$ and *severe* $\in (40, -]$ with approximately equal number of videos. Table VI shows the number of video fragments in each class. Experiments are repeated 20 times to ensure validity.

1) Single-Task Baseline: To demonstrate the effectiveness of task-assembling methods, we first do patient-level severity classification using only one single task as the baseline. The result is shown in Table VII. The best MCC is 0.31 using single task *hand*, which is served as the baseline to compare with assembling methods.

TABLE VII
SINGLE TASK BASELINE FOR PATIENT-LEVEL SEVERITY CLASSIFICATION (THREE CLASSES). EACH ROW SHOWS THE PERFORMANCE OF A TASK AND COLUMNS GIVE THE RESULT OF ACCURACY AND MCC WITH STANDARD DEVIATION PROVIDED. RANK IS CALCULATED BASED ON THE AVERAGE MCC FROM TWO INPUTS. THE TOP-3 WELL-PERFORMED TASKS USED FOR PATIENT-LEVEL CLASSIFICATION ARE TASK *HAND*, *KINETIC* AND *FINGER*. TASK *HAND* ACHIEVES A MCC OF 0.31, WHICH IS USED AS THE BEST SINGLE-TASK BASELINE

Task	Input	Accuracy	MCC	Rank
finger	*avgpool*	60.3±2.8	0.30±0.05	3
	layer4	60.7±3.2	0.31±0.04	
hand	*avgpool*	61.5±2.8	0.31±0.04	1
	layer4	60.7±3.1	0.30±0.03	
kinetic	*avgpool*	59.7±2.7	0.29±0.04	2
	layer4	60.5±3.4	0.30±0.04	
leg	*avgpool*	50.6±2.7	0.21±0.05	6
	layer4	60.0±3.7	0.27±0.04	
postural	*avgpool*	54.9±2.5	0.19±0.05	5
	layer4	60.8±3.5	0.29±0.04	
pronation	*avgpool*	59.3±3.2	0.20±0.05	4
	layer4	61.3±3.4	0.31±0.04	
toe	*avgpool*	51.3±2.8	0.17±0.06	7
	layer4	60.6±3.9	0.28±0.04	

TABLE VIII
PATIENT-LEVEL SEVERITY CLASSIFICATION (THREE CLASSES) USING SINGLE TASK AS A BASELINE AND TASK-ASSEMBLING APPROACHES (SEVEN TASKS). EACH ROW SHOWS THE PERFORMANCE OF A TASK-ASSEMBLING METHOD ON THE INPUT FROM A CERTAIN LAYER. THE FOUR TASK-ASSEMBLING METHODS OUTPERFORM THE SINGLE-TASK BASELINE WITH THE CHANNEL-WISE AND PIXEL-WISE ATTENTION WEIGHTING BEING THE BEST METHODS

Method	Input	Accuracy	MCC
single task baseline	*avgpool*	61.5±2.8	0.31±0.04
vector averaging	*avgpool*	62.7±2.4	0.32±0.06
vector weighting	*avgpool*	64.1±2.4	0.37±0.06
channel-wise attention weighting	*layer4*	64.5±3.1	0.38±0.05
pixel-wise attention weighting	*layer4*	64.5±2.8	0.39±0.06

2) Benefit From Task-Assembling Methods: Four task-assembling methods incorporate seven tasks used in task-level severity classification. From Table VIII, we see that all task-assembling methods, including the most straightforward averaging strategy, outperforms the single-task baseline. The best method is the pixel-wise self-attention based weighting in terms of the MCC, with an improvement of 25.8% from the baseline. These results demonstrate that patient-level severity classification benefits from all tasks combined compared to based on a single task, which is intuitive since it is also hard for experts to diagnose a patient by inspecting just one task.

Comparing all four methods, we see the weighting strategy is better than just simple averaging, indicating that each task contributes unequally to the patient-level severity. Moreover, the attention-based weighting slightly outperforms the learnable vector-based weighting. It is because 1) *layer4* has more feature points, potentially more representable for a task than *avgpool*, and 2) attention-based weighting gives more flexibility to the weights such that patients can have task weights exclusively learned based on their condition.

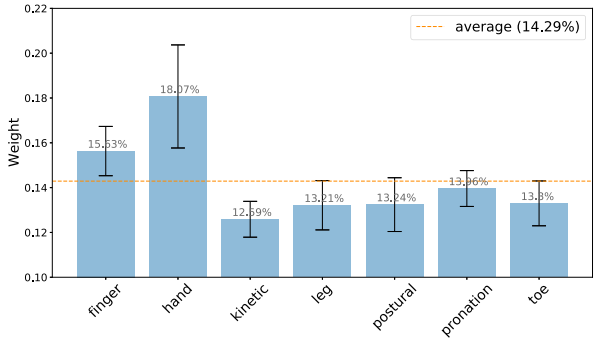


Fig. 7. Weights for seven tasks learned by vector weighting method. The weights of task *finger* and *hand* are higher than the average, which means in the task-assembling approach, i.e., vector weighting, they contribute more than other tasks in the prediction of the patient-level severity.

TABLE IX
PATIENT-LEVEL SEVERITY CLASSIFICATION (TWO CLASSES WITH CLASS *MODERATE* REMOVED) USING SINGLE TASK AND TASK-ASSEMBLING APPROACHES (SEVEN TASKS). EACH ROW SHOWS THE PERFORMANCE OF A TASK-ASSEMBLING METHOD ON THE INPUT FROM A CERTAIN LAYER. THE FOUR TASK-ASSEMBLING METHODS OUTPERFORM THE SINGLE-TASK BASELINE WITH THE PIXEL-WISE ATTENTION WEIGHTING BEING THE BEST METHOD

Method	Input	Accuracy	MCC
single task baseline	*avgpool*	81.1±2.2	0.60±0.06
vector averaging	*avgpool*	81.4±1.7	0.61±0.05
vector weighting	*avgpool*	81.9±2.1	0.64±0.07
channel-wise attention weighting	*layer4*	82.2±3.1	0.66±0.09
pixel-wise attention weighting	*layer4*	83.6±1.2	0.68±0.04

We show the weights learned in the vector weighting method in Fig. 7 to give a general feeling of which task may contribute less or more to the prediction of patient-level severity. Weights are averaged across 20 runs on each fold, a total of 100 runs. As the two attention-based weighting methods assign task weights for patients exclusively, it is not intuitive to see the overall weight distribution on tasks. In Fig. 7, we see the top-2 tasks with highest weights are *hand* and *finger*, which well matches the performance rank in Table VII. The rest tasks remain the similar position as in Table VII except that task *kinetic* drops to the lowest rank. We suspect the reason being the effect of severe class imbalance problem of task *kinetic*.

3) Distinguishing Between Slight and Severe Classes: We remove the class *moderate* with the remaining classes untouched to focus on the classification between *slight* and *severe* classes. The result of the best single task baseline and assembling methods are shown in Table IX. By combining seven tasks, we gain 1.7%-13.3% performance improvements compared to using a single task. At best, we can achieve a MCC of 0.68 on distinguishing between *slight* and *severe* classes. Moreover, the attention-based weighting methods still outperform the vector-based ones, matching the case in Table VIII.

In general, attention-based weighting strategy is the first choice to assemble the tasks, but the vector-based one is also

applicable, given its higher time efficiency. It is also worthwhile to exclude some tasks to see the ablation effects on patient-level performance. As the main focus of this paper is to show the potential of combining tasks, we leave it as future work.

In Section V-D2 and V-D3, we empirically show the possibility that a multi-task algorithm based on an incomplete video-overview (i.e. not all MDS-UPDRS-III items are included) can help discriminate between groups of disease severity in both *slight-moderate-severe* and *slight-severe* cases with acceptable MCC, 0.39 for the former case and 0.68 for the latter case. Besides, the performance of single task and weights visualization demonstrates the test of bradykinesia hands among all videotaped items is the best reflections of the total MDS-UPDRS-III.

VI. Conclusion

In this paper, we successfully apply deep architectures on the PD video dataset to automatically identify the task-level severity, i.e., item scores in MDS-UPDRS-III given the video of the task, with satisfactory performance in terms of both accuracy and MCC. Due to the small size of our PD dataset, we employ transfer learning from non-medical datasets to improve the performance of the model.

We propose a temporal self-attention method, TSA, for action recognition problem and validate it on two commonly used public datasets and our PD dataset. The promising results compared to I3D demonstrate the effectiveness of TSA and better ability of handling motion discrepancy between non-medical datasets and our PD dataset during transfer learning. TSA is highly flexible and can be embedded in any 3D network for action recognition by replacing the CNN layer with the temporal relative self-attention block.

We propose four task-assembling methods to incorporate tasks to identify the patient-level severity by using the models trained on each task. Compared to using only a single task, tasks combination can produce a better performance under both classification scenarios: *slight-moderate-severe* and *slight-severe*. It is clinically interesting that through analysis of a limited number of selected tasks, we can deduct a global severity score given the reasonably good accuracy and MCC.

In this study, we focus on only 7 tasks and each of them is based on one particular video-segment. In MDS-UPDRS-III, the scores of other tasks are also indicators for PD severity, such as resting state tremor and freezing of gait. However, video samples from these tasks contain multiple view and scene changes and most part of the video is not highly relevant for severity score prediction. So we exclude these tasks temporarily to prevent from leading to an inaccurate conclusion. In the future work, we will try to include all the tasks with video data and propose new methodologies to overcome these difficulties, further illustrating the feasibility of our methods in this study. The clinical asymmetry which may be present in PD was not considered in this study. Future research should identify whether motor asymmetry plays an important role during automated assessments of motor severity in PD.

We take this study as a preliminary step for PD severity prediction. Several additional steps should still be taken before algorithms can be applied robustly in the real clinical world, such as collection of much more data and findings of more advanced class-imbalance-free models. However, some results of our current methods already matches the clinical description of PD. For instance, the tasks related to finger or hand movements are most sensitive to reflect motor disease severity, in comparison to other tasks. This implies that the severity of upper extremity bradykinesia best reflects the total motor severity, which closely adheres to the clinical diagnosis of PD. Furthermore, the result also shows that bradykinesia of the upper extremity is more sensitive than bradykinesia of the lower extremity, which suggests that assessment of severity should be more focused on upper body bradykinesia than lower body bradykinesia. However, it is questionable whether upper limb bradykinesia should be considered a gold standard. Future research should attempt to replicate and validate this finding before implementation in clinical practice.

The proposed methodology here can be used in other disorders with motor phenotypes, such as classification of disease severity in e.g. Huntington's Disease, or differentiating motor phenotypes such as epilepsy vs. psychogenic non-epileptic seizures, indicating its utility beyond Parkinson's Disease.

Acknowledgment

This work is part of the research programme C2D-Horizontal Data Science for Evolving Content with project name DACCOMPLI and project number 628.011.002, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO) and is also supported by the Leiden Institute of Advanced Computer Science (LIACS).

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *International Workshop on Human Behavior Understanding*. Berlin, Heidelberg: Springer, 2011, pp. 29–39.

[2] E. A. Belalcazar-Bolanos, J. D. Arias-Londono, J. F. Vargas-Bonilla, and J. R. Orozco-Arroyave, "Nonlinear glottal flow features in Parkinson's disease detection," in *Proc. 20th Symp. Signal Process., Images Comput. Vis. (STSIVA)*, 2015, pp. 1–6.

[3] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3286–3295.

[4] A. H. Butt, E. Rovini, C. Dolciotti, P. Bongioanni, G. De Petris, and F. Cavallo, "Leap motion evaluation for assessment of upper limb motor skills in Parkinson's disease," in *Proc. Int. Conf. Rehabil. Robot.*, 2017, pp. 116–121.

[5] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6299–6308.

[6] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Linear algorithms for online multitask classification," *J. Mach. Learn. Res.*, vol. 11, pp. 2901–2934, 2010.

[7] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," 2015, *arXiv:1506.07503*.

[8] K. L. Chou, J. L. Taylor, and P. G. Patil, "The MDS UPDRS tracks motor and non-motor improvement due to subthalamic nucleus deep brain stimulation in Parkinson disease," *Parkinsonism Related Disord.*, vol. 19, no. 11, pp. 966–969, 2013.

[9] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9268–9277.

[10] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 781–796, Aug. 2000.

[11] A. Davis, K. L. Bouman, J. G. Chen, M. Rubinstein, F. Durand, and W. T. Freeman, "Visual vibrometry: Estimating material properties from small motion in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5335–5343.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[13] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2625–2634.

[14] M. Dredze and K. Crammer, "Online methods for multi-domain learning and adaptation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2008, pp. 689–697.

[15] P. Drotár, J. Mekyska, I. Rektorová, L. Masarová, Z. Smékal, and M. Faundez-Zanuy, "Evaluation of handwriting kinematics and pressure for differential diagnosis of Parkinson's disease," *Artif. Intell. Med.*, vol. 67, pp. 39–46, 2016.

[16] L. J. W. Evers, J.H. Krijthe, M.J. Meinders, B.R. Bloem, and T. M. Heskes, "Measuring Parkinson's disease over time: The real-world within-subject reliability of the MDS-UPDRS," *Movement Disord.*, vol. 34, no. 10, pp. 1480–1487, 2019.

[17] S. R. L. E. Fahn, "Unified Parkinson's disease rating scale," *Recent Develop. Parkinson's Dis.*, 1987.

[18] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1933–1941.

[19] C. Ferraris *et al.*, "Automated assessment of motor impairments in Parkinson's disease," *Clin. Neurol. Int.*, vol. 1, no. 4, 2020.

[20] C. G. Goetz *et al.* "Movement disorder society-sponsored revision of the unified Parkinson's disease rating scale (MDS-UPDRS): Scale presentation and clinimetric testing results," *Movement Disord.: Official J. Movement Disord. Soc.*, vol. 23, no. 15, pp. 2129–2170, 2008.

[21] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imagenet,?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6546–6555.

[22] C. J. Hartmann *et al.*, "Long-term evaluation of impedance levels and clinical development in subthalamic deep brain stimulation for Parkinson's disease," *Parkinsonism Related Disord.*, vol. 21, no. 10, pp. 1247–1250, 2015.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[26] M. Joshi, M. Dredze, W. Cohen, and C. Rose, "Multi-domain learning: When do domains matter?," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2012, pp. 1302–1312.

[27] W. Kay *et al.* "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.

[28] J. F. P. Kooij and J. C. van Gemert, "Depth-aware motion magnification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 467–482.

[29] M. W. Kouw and M. Loog, "An introduction to domain adaptation and transfer learning," 2018, *arXiv:1812.11806*.

[30] M. Lu *et al.*, "Vision-based estimation of MDS-UPDRS gait scores for assessing Parkinson's disease motor severity," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2020, pp. 637–647.

[31] F. Maier *et al.* "Subjective perceived outcome of subthalamic deep brain stimulation in Parkinson's disease one year after surgery," *Parkinsonism Related Disord.*, vol. 24, pp. 41–47, 2016.

[32] A. Makkos *et al.*, "Are the MDS-UPDRS-based composite scores clinically applicable?," *Movement Disord.*, vol. 33, no. 5, pp. 835–839, 2018.

[33] P. Martinez-Martin *et al.* "Expanded and independent validation of the movement disorder society-unified Parkinson's disease rating scale (MDS-UPDRS)," *J. Neurol.*, vol. 260, no. 1, pp. 228–236, 2013.

[34] M. Merello, E. R. Gerschcovich, D. Ballesteros, and D. Cerquetti, "Correlation between the movement disorders society unified Parkinson's disease rating scale (MDS-UPDRS) and the unified Parkinson's disease rating scale (UPDRS) during L-dopa acute challenge," *Parkinsonism Related Disord.*, vol. 17, no. 9, pp. 705–707, 2011.

[35] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010.

[36] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1717–1724.

[37] L. Silvia *et al.*, "Hand-tremor frequency estimation in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2018.

[38] R. Polana and R. C. Nelson, "Detection and recognition of periodic, nonrigid motion," *Int. J. Comput. Vis.*, vol. 23, no. 3, pp. 261–282, 1997.

[39] M. Raptis and L. Sigal, "Poselet key-framing: A model for human activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2650–2657.

[40] G. Saranza and A. E. Lang, "Levodopa challenge test: Indications, protocol, and guide," *J. Neurol.*, 2020.

[41] S. Satkin and M. Hebert, "Modeling the temporal extent of actions," in *Proc. Eur. Conf. Comput. Vis.* , 2010, pp. 536–548.

[42] S. Schade, F. Sixel-Döring, J. Ebentheuer, X. Schulz, C. Trenkwalder, and B. Mollenhauer, "Acute levodopa challenge test in patients with de novo Parkinson's disease: Data from the denopa cohort," *Movement Disord. Clin. Pract.*, vol. 4, no. 5, pp. 755–762, 2017.

[43] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Deep spatial-semantic attention for fine-grained sketch-based image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5551–5560.

[44] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.

[45] D. G. Standaert, M. H. Saint-Hilaire, and C. A. Thomas, *Parkinson's Disease Handbook*. New York, NY, USA: American Parksinon Disease Association, 2015.

[46] R. Sun, Z. Wang, K. E. Martens, and S. Lewis, "Convolutional 3 D attention network for video based freezing of gait recognition," in *Proc. Digit. Image Comput.: Techn. Appl.*, 2018, pp. 1–7.

[47] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.

[48] S. Sveinbjornsdottir, "The clinical symptoms of Parkinson's disease," *J. Neurochemistry*, vol. 139, pp. 318–324, 2016.

[49] C. Thanawattano, C. Anan, R. Pongthornseri, S. Dumnin, and R. Bhidayasiri, "Temporal fluctuation analysis of tremor signal in Parkinson's disease and essential tremor subjects," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2015, pp. 6054–6057.

[50] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3 D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[51] H. T. Turner and M. L. Dale, "Inconsistent movement disorders society-unified Parkinson's disease rating scale part III ratings in the Parkinson's progression marker initiative," *Movement Disord.*, 2020.

[52] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[53] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, "Phase-based video motion processing," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–10, 2013.

[54] F. Wahid, R. K. Begg, C. J. Hass, S. Halgamuge, and D. C. Ackland, "Classification of Parkinson's disease gait using spatial-temporal gait features," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 6, pp. 1794–1802, Nov. 2015.

[55] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[56] Z. Wang, J. Li, S. Khademi, and J. van Gemert, "Attention-aware age-agnostic visual place recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019.

[57] Z. Wang, M. Loog, and J. van Gemert, "Respecting domain relations: Hypothesis invariance for domain generalization," 2020, *arXiv:2010.07591*.

[58] J. West, D. Ventura, and S. Warnick, "Spring research presentation: A theoretical foundation for inductive transfer," *Brigham Young Univ., College Phys. Math. Sci.*, vol. 1, no. 8, 2007.

[59] C. David *et al.*, "Supervised classification of Bradykinesia for Parkinson's disease diagnosis from smartphone videos," in *Proc. IEEE 32nd Int. Symp. Comput.-Based Med. Syst.*, 2019, pp. 32–37.

[60] A. W. Yu *et al.*, "QANet: Combining local convolution with global self-attention for reading comprehension," 2018, *arXiv:1804.09541*.