An Oculomotor Digital Parkinson Biomarker From a Deep Riemannian Representation

Juan Olmos[1],[0000−0002−6017−0867] Antoine Manzanera[2][0000−0001−5718−411X], and Fabio Martínez[1,*][0000−0001−7353−049X]

[1] Biomedical Imaging, Vision and Learning Laboratory (BIVL[2]ab), Universidad Industrial de Santander (UIS), Cra 27 Calle 9 Ciudad Universitaria, Bucaramanga, Colombia.
jaolmosr@correo.uis.edu.co, famarcar@saber.uis.edu.co

[2] U2IS, ENSTA Paris, Institut Polytechnique de Paris, 828 Boulevard des Maréchaux, 91762 Palaiseau CEDEX, France.
antoine.manzanera@ensta-paris.fr

Abstract. Parkinson's disease (PD) is characterized by motor alterations and associated with dopamine neurotransmitters degeneration, affecting 3 % of the population over 65 years of age. Today, there is no definitive biomarker for an early diagnosis and progression characterization. Recently, oculomotor alterations have shown promising evidence to quantify PD patterns. Current capture and oculomotor setups however require sophisticated protocols, limiting the analysis to coarse measures that poorly exploit alterations and restrict their standard use in clinical environments. Computational based deep learning strategies today bring a robust alternative by discovering in video sequences hidden patterns associated to the disease. However, these approaches are dependent on large training data volumes to cover the variability of patterns of interest. This work introduces a novel strategy that exploits data geometry within a deep Riemannian manifold, withstanding data scarcity and discovering oculomotor PD hidden patterns. First, oculomotor information is encoded as symmetric matrices that capture second order statistics of deep features computed by a convolutional scheme. These symmetric matrices then form an embedded representation, which is decoded by a Riemannian network to discriminate Parkinsonian patients w.r.t a control population. The proposed strategy, evaluated on a fixational eye experiment, proves to be a promising approach to represent PD patterns.

Keywords: Oculomotor patterns · Parkinson's Disease classification · SPD pooling · Deep non-linear learning · Riemannian manifold

1 Introduction

Neurological diseases are currently the major cause of disability across the world [5]. Parkinson's disease (PD) is the second most common neurodegenerative disorder, affecting around 2-3 % of the global population over the age of 65. Actually, this disease reports a prevalence around 22%, being the neurological disorder of fastest growth world-around [16, 5]. The PD is related to the disruption

of the dopamine neurotransmitters that control voluntary movement, producing in consequence alterations in the patient's movement. Currently, this disease has no cure, but the early and personalized treatment planning is fundamental to slow down motor symptoms and disabilities. Nonetheless, today there is no definitive disease biomarker and the diagnosis is commonly subject to observational analysis, reporting errors up to 24% [16, 19]. In the literature have been done multiple efforts to characterize and measure motor disabilities correlated with PD. However, the motion patterns related to tremor in hands, disabilities during gait and trunk rigidity [22], are mostly captured at an advanced stage of the disease. More recently, different studies have experimentally supported the hypothesis of a strong correlation of oculomotor patterns with PD, even at early stages [22, 4, 8]. These patterns, however, are collected from sophisticated capture devices that simplify eye dynamic to global displacement trajectories, making difficult to address the wide range of disease evolution.

Computational approaches have emerged as an alternative to support the quantification of motor patterns directly from video analysis. More recently, deep learning approaches have revealed determining advantages to discover hidden patterns and to characterize kinematic descriptors in the modelling of gait videos, eye movement disorders, cerebrospinal data and tracking eye movement, among many others [21, 9]. These deep strategies however require a huge amount of training information to deal with observation variability and the quantification of proper hidden variables demand deeper architectures to discriminate the data [13, 21]. These requirements are rarely realistic in clinical scenarios with large pattern variability, and where annotated examples are difficult to get, above all in the task of discovering motor anomalies associated with PD. To avoid such challenging training scheme, some works have used a collection of convolutional responses extracted from the first layers of a Convolutional Neural Network (CNN) architecture previously trained on a general natural image classification problem [17]. Subsequently, different works propose pooling methods to compact these representations in low dimension descriptors using Symmetric Positive Definite (SPD) matrices that summarize feature statistics [17, 2]. However, these matrices belong to a Riemannian manifold, making it necessary to design proper methods regarding the geometric structure [11, 3]. Despite current efforts, manifold learning methods still use shallow learning and machine learning algorithms need been redesigned to take into account data geometry [11].

This work introduces a novel digital biomarker that captures discriminatory PD fixational eye patterns, following a Riemannian deep representation, that compactly codes in symmetric matrices second order statistics of deep convolutional features. For this purpose, each video sequence is transformed in spatio-temporal slices that recover tiny tremor patterns, during ocular fixation experiment. From this input, an end-to-end learning process is herein proposed through a hybrid deep network, whose first layers are convolutional (CNN) and last layers are Riemannian (SPDNet). The CNN module projects the input slices to deep features, which are then summarized in symmetric positive embedding matrices,

allowing to exploit feature correlations that are related to visual observations. Hence, these embedding matrices feed the Riemannian module, that focus on non-linear learning, while preserving geometry of input SPD data, achieving a discrimination between Parkinsonian and control classes. The result is a method able to discriminate between oculomotor patterns of patients diagnosed with the disease and a control population.

2 Proposed Method

We hypothesize that symmetric positive embedding vectors are key to discover new PD biomarkers from deep oculomotor patterns. In this work, we designed an end-to-end Riemannian deep strategy (ConvSPD network) that uses spatio-temporal slice observations of a fixational experiment. Each slice is overcompletely represented by convolutional deep features, which in turn are summarized in symmetric embedding vectors and finally exploited by a Riemannian module to carry out the classification task. The proposed representation preserves the Riemannian geometry of data and robustly discriminates Parkinson patients from a control population. Figure 1 summarizes the proposed pipeline.

2.1 Convolutional Module and Symmetric Pooling Representation.

A convolutional scheme is here introduced as the first part of the proposed approach to represent slices of oculomotor sequences. More precisely, this CNN module learns to extract early to mid-level (textural) features, to capture relevant patterns in eye micro-movements during fixational experiment. The convolutional representation is structured in several layers, hierarchically organized to progressively increase the time $\times$ space receptive field, as well as the semantic level w.r.t. Parkinsonian classification.

In the deepest layer, the corresponding D feature maps form a tensor $X \in \mathbb{R}^{D \times W \times H}$, where (W, H) are the dimensions of the feature maps, resulting from the successive convolutional and pooling processes along the representation. This embedding tensor X, is then summarized in a special symmetric positive embedding matrix by computing second-order statistics from such description [2, 20]. To this end, we implement a special pooling layer (*SPDpool*) that summarizes the information from the last layer into a symmetric positive embedding matrix (SPD matrix).

In general SPD matrices aim to compute second order statistics to recover similarities among features. For instance, by taking the inner product of all pairs of feature vectors, some works propose correlation volumes to compute visual similarities between pixels [18]. In this work, if we suppose that $(k-1)$ is the index of the last convolutional layer, we re-organize the embedding tensor X in a matrix named X_{k-1} with D rows and $W \times H$ columns. Then we compute the outer product of X_{k-1} with itself. Then, the proposed SPDpool layer calculates: $X_k = f_{SPDpool}(X_{k-1}) = \frac{1}{W \times H} X_{k-1} X_{k-1}^T$. This way, the (i, j) element of X_k is the inner product (correlation) between the i-th feature map and the j-th

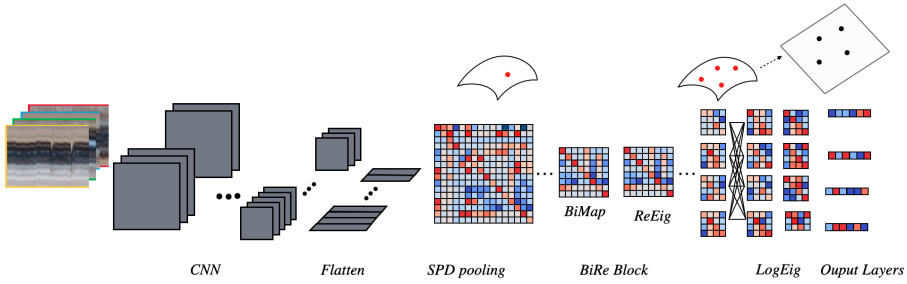

Fig. 1. **Architecture of the proposed ConvSPD network.** ~~**Pipeline of the proposed ConvSPD network.**~~ Starting from a collection of space × time slice images summarizing the oculomotor video sequence, a CNN module first projects them to an overcomplete embedding of deep features. These features are then flattened and compacted using a SPD pooling layer. ~~Flattening these SPD representations,~~ The classification is finally performed by a Riemannian non-linear SPDNet module.

feature map. This provides second order statistics, and has been used to measure statistical discrepancy [14]. Here, the resultant symmetric positive embedding $D \times D$ matrix allows to capture the most statistically relevant relationships in the previous CNN module, with respect to the Parkinson classification task.

2.2 Riemmanian Module Structure

In the same way as the SPD Gram matrix was used in [6] and [7] to model texture and pictorial style respectively, we may consider the Parkinsonian stage of the patient as a stationary parameter, in the sense that it will affect the ocumolotor action independently on the temporal position (phase) of the movement. In short, the oculomotor task (fixation or tracking) represents the "content" (layout) of the input slices, whereas the stage of the disease represents their "style". However, unlike [7], who use Gram matrices in their loss function, we choose to go on working with stationary features that may be interpreted in terms of distributions, by completing the CNN module by a Riemannian SPD network.

Then, once obtained the SPD embedding, we maintain the deep representation while taking into account the Riemmanian geometry of SPD matrices. We then base the next processing layers on the SPDNet framework, that carries out a deep non-linear representation [11]. This requires the codification of special layers, such as: *BiMap*, *ReEig*, and *LogEig*. We now describe the respective Riemmanian layers and their particular learning procedure.

The BiMap layer is a fully connected layer designed to generate a new bank of more compact and discriminative SPD matrices by a bilinear mapping

$$X_k = W_k X_{k-1} W_k^T. \tag{1}$$

Here, $X_{k-1} \in S_{++}^{d_{k-1}}$ is the input $d_{k-1} \times d_{k-1}$ SPD matrix of the layer $(k-1)$, and $W_k \in \mathbb{R}_*^{d_k \times d_{k-1}}$ is the transformation matrix (connection weight). Similarly to the CNN, the sizes of the SPD matrices decrease from layer to layer, i.e. $d_k < d_{k-1}$, and the *BiMap* layer actually uses m different weight matrices $\{W_k^{(i)}\}_{i=1}^m$ in order to generate m SPD matrices $\{X_k^{(i)}\}_{i=1}^m$ in the k-th layer.

The ReEig layer was inspired by rectified linear units (ReLU) of CNNs. It is composed of a non-linear function to improve the training process by rectifying the SPD matrices:

$$X_k = U_{k-1} \max(\varepsilon I, \Sigma_{k-1}) U_{k-1}^T,$$

where U_{k-1} and Σ_{k-1} come from the eigen decomposition $X_{k-1} = U_{k-1}\Sigma_{k-1}U_{k-1}^T$. Here, $\varepsilon \in \mathbb{R}_+^*$ is a rectification threshold, I is the identity matrix and Σ_{k-1} the diagonal matrix of the eigenvalues of X_{k-1}. This operation tunes up the eigenvalues avoiding non-positivenes and improving the discriminative performance.

The LogEig layer results from the necessity to project SPD matrices back to Euclidean space where the classifiers are designed. In Riemannian manifolds we can attach each point to a flat tangent space with a vector space structure. This structure ease classic Euclidean computations. To map elements here is used the Riemannian logarithm map. To facilitate the computing and work on the same tangent space, we map the SPD matrix X_{k-1} onto the tangent space at the identity by

$$X_k = \log(X_{k-1}) = U_{k-1} \log\left(\Sigma_{k-1}\right) U_{k-1}^T.$$

Thereafter, the output layers correspond to the classic final layers of neural networks, e.g. a flatten layer or a fully connected layer, and the final output layer should be a softmax operation.

Learning Scheme The proposed ConvSPD was trained following a classic back-propagation in convolutional module, but for LogEig and ReEig Riemmanian layers was taken into account structured derivatives in propagation [12]. Besides, to learn weights of the *BiMap* layers (equation 1) it is necessary to consider special optimization constraints. To generate consistent SPD matrices and achieve a feasible optimization, connection weights are non squared, orthogonal, and row full-rank matrices. That implies these weights matrices lie in a compact *Stiefel manifold* $St(d_k, d_{k-1})$ [11]. To calculate the gradient, in the equation (1) the steepest descent direction with respect to W_k on the Stiefel manifold is given by

$$\widetilde{\nabla} L_{W_k^t}^{(k)} = \nabla L_{W_k^t}^{(k)} - \nabla L_{W_k^t}^{(k)} \left(W_k^t\right)^T W_k^t \tag{2}$$

which is the tangent component obtained by subtracting the normal component to the Euclidean gradient $\nabla L_{W_k^t}^{(k)}$. Finally, the step of gradient descent for the connection weights is

$$W_k^{t+1} = \Gamma\left(W_k^t - \alpha \widetilde{\nabla} L_{W_k^t}^{(k)}\right) \tag{3}$$

where Γ is a retraction operation to get back to the Stiefel manifold, and α is the learning rate [11]. Given some ξ in the tangent component, by the orthogonal constrain, the retraction map is reduced to the calculus of the Q factor of the polar decomposition of $W_k^t + \xi$ [1].

3 Experimental Setup

3.1 Dataset Description

A total of 13 PD patients (average age of 72.3 ± 7.4) and 13 control subjects (average age of 72.2 ± 6.1) were captured and analyzed for evaluating the proposed approach. A camera with a temporal resolution of 60 fps was fixed in front of the subjects to capture their upper face region. Participants were invited to sit and observe a simple stimulus as illustrated in Figure 2(a). A single fixation period of 5 seconds was set and individual eyes were manually cropped to 210×140 pixels by centering the first frame to the center of the pupil, to obtain the sequences of interest. In total, 5 video sequences was obtained for each participant. Subjects with different disease degree progression were selected to include inter-subjet variability. With the help of a physical therapist, PD patients were categorized into the Hoehn-Yahr rating scale. A total of five patients were categorized in stage two, six patients on stage three, and two patients on stage four. The dataset was approved by an Ethic Committee. Written informed consent was obtained for every participant. After a search of public datasets, no more data was found similar in the number of samples.

For spatio-temporal slice computation, each video being considered as a volume $\{I(x, y, t)\}_{x=1,y=1,t=1}^{W,H,N}$, with spatial dimension $W \times H$ and N frames, we choose 4 radial directions $\theta_i \in \{0°, 45°, 90°, 135°\}$ around the center, and cut the video volume along each direction as illustrated in Figure 2(a). The output is then four 2d slice images $S_\theta(x, t)$, each one recording subtle eye displacements, capturing potential oculomotor alterations related with PD.

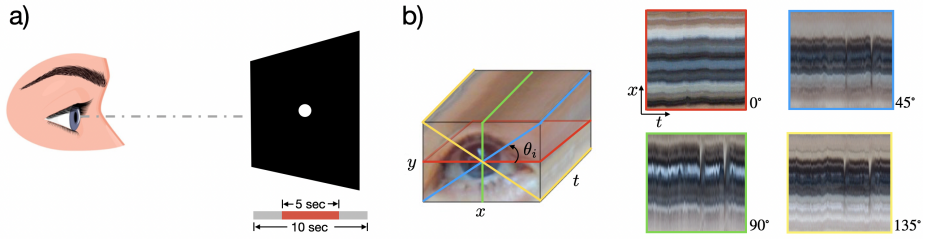

Fig. 2. Dataset: (a) Ocular fixation task. (b) 2D Video slices representation of eye movements.

3.2 Network Configuration.

For the convolutional module, the weights were initialized from a pre-trained **ResNet-18** architecture [10], which has a total of 8 convolutional blocks. Regarding the Riemannian module, we implement BiRe blocks (a BiMap layer, followed by a ReEig layer), which reduce the input dimension by half. A total of four models were herein evaluated using the following architecture:

- A convolutional module using only the first $N = \{2, 4, 6, 8\}$ convolutional blocks, which outputs 64, 128, 256 and 512 feature maps of size 53×75, 27×38, 14×19 and 7×10 respectively.
- A pooling layer using $f_{SPDpool}$.
- A Riemannian module with 3 BiRe blocks and a LogEig layer.
- Output layers: a flatten layer, a fully connected layer, and a softmax dedicated to classification output.

We call this hybrid model a ConvSPD N-th block model, which is trainable end-to-end starting from video slices while learning convolutional and Riemannian structures.

A Leave one out cross validation scheme was carried out to evaluate the performance of the proposed models. In such sense, each fold leaves one patient for testing while the remaining are used for training, for a total of 26 experiments.

During training and validation, we also recover the resultant Riemmanian representation, encoded in SPD embedding matrices. We compute distances among these matrices, that represent points in the learned Riemmanian manifold, to measure the discriminatory capability of such representation. We use the Riemannian distance $d_R(x, y) = ||\log_x(y)||_x$, where x, y are embedding points, $\log(\cdot)$ is the Riemannian logarithm map and $||\cdot||_x$ an *affine-invariant* norm [15]. With the class-labeled data, we consider two disjoint sets P (Parkinsonian) and C (control) in the manifold, respectively. Therefore we define the separation metric as the Riemannian distance between P and C, as $d_R(P, C) = \frac{1}{|P||C|} \sum_{x \in P} \sum_{y \in C} d_R(x, y)$.

4 Evaluation and Results

In the proposed architecture, a deeper convolutional module produces more feature maps but with a smaller size. Therefore, a first validation of the proposed approach was carried out varying the number of convolutional blocks previous to the SPD pooling layer. Such validation was carried out according to the classification task, measuring the sensitivity, specificity, precision, accuracy, and F1-Score of each model respectively. Table 1 summarizes performance achieved from the different configurations, achieving outstanding accuracy scores, above 90% for all validations. In all metrics it is visible that intermediate and deep blocks aid the network to achieve better results. This way, mid-level feature maps support the ConvSPD 4th-Block model to accurately predict PD patients with an accuracy of 97.7%. Moreover, the results of ConvSPD 4th-Block and 6th-Block

enhance the trade-off between the size and the number of features, where small features (ConvSPD 8th-Block) turn out to be statistically insufficient and few features (ConvSPD 2nd-Block) limit the measurement of similarities. As comparison with standard convolutional nets, we implement the complete **ResNet-18** CNN, achieving only an accuracy of 94.26%, with a relative more complex architecture and a total of 11.7M parameters. As baseline comparison, we compared with a machine learning approach that classifies SPD matrices constructed from convolutional responses [17]. Under the same data validation conditions, this approach reports an accuracy of 87.7%, 10% less than the proposed end-to-end ConvSPD 4th-Block model.

Table 1. Classification results of the proposed ConvSPD N-th block models for $N = 2, 4, 6, 8$.

Model	Sen(%)	Spe(%)	Pr(%)	Acc(%)	F1(%)	Parameters	Time
ConvSPD 8th-Block	94.4	94.4	94.4	94.4	94.4	11.6M	1.63s
ConvSPD 6th-Block	95.4	94.4	94.4	94.9	94.9	2.83M	0.88s
ConvSPD 4th-Block	97.2	98.5	98.4	**97.8**	97.8	0.70M	0.54s
ConvSPD 2nd-Block	94.9	93.1	93.2	94.0	94.0	0.16M	0.27s

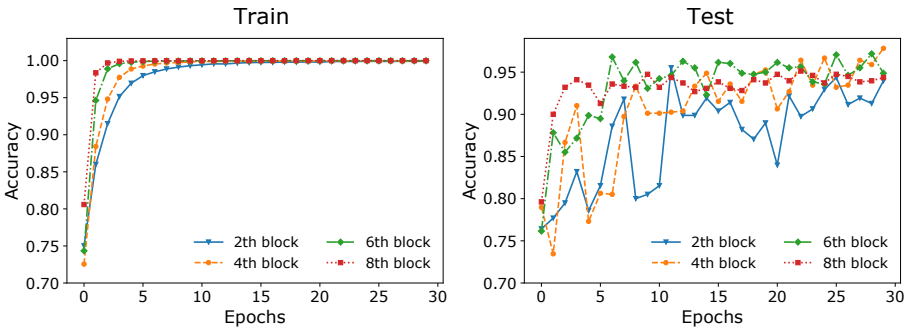

Fig. 3. Convergence of the models

The convergence of the proposed approach was also analyzed to measure the stability of the SPD component together with convolutional representation. Figure 3shows the effective convergence performance for all models. Besides, deeper models present a rapid and stable convergence as expected, thanks to the

wide receptive field. On the other hand, since shallow models have feature maps with less semantics, these results show the ability of the Riemannian module to improve the networks generalization.

The inference time of the proposed descriptor is determinant to be implemented in clinical scenarios, and routine validation of PD patients. For this reason, the Table 1also reports the inference time and the total number of parameters, that requires each of the validated configurations. Inference time refers to the time that takes the model to forward the video slices data and produce a prediction. We can observe that the increasing between models is exponential, the inference time almost doubles, and the number of parameters is quadrupled. However, the ConvSPD 2nd-Block and ConvSPD 4th-Block are significantly light models, and all models except ConvSPD 8th-Block compute the inference in acceptable time ($<$ 1 sec). From this, we can see how the ConvSPD 4th-Block model with almost 20 times fewer parameters outperforms the complete Convolutional model. The implementation of clinical routine requires a trade-off between computational inference time, parameters of the representation to define computational architecture, but also sufficient precision to support diagnosis and following.

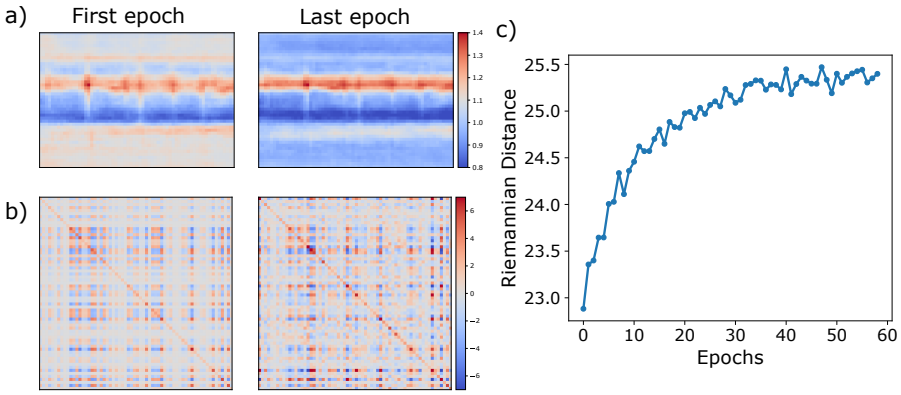

Fig. 4. Analysis of ConvSPD structures. We take sample activations extracted from the ConvSPD 4th-Block model. **a)** Average activations from the second block of the convolutional module on the first (left panel) and last (right panel) epoch. **b)** SPD matrices from the third BiRe block of the Riemannian module on the first (left panel) and last (right panel) epoch. **c)** Evolution of the average Riemannian distance between Parkinson and Control subjects on the test sets.

As visual analysis, we recover the feature maps **from PD patients** of the 2nd-Block at first and last epoch. For this last experiment, **instead of carry out a cross validation, two random subjects were chosen for each class, and the rest was used for training a ConvSPD *4th-Block* model one**

time. ~~two random subjects were chosen for each class, and the rest was used for training a ConvSPD *4th-Block* model.~~ In Figure 4.a) is illustrated an average of these maps, observing the evolution of the model to detect areas of interest in the slice. Specifically, the model focus on the center of slices, indicating that the ConvSPD use eye movement information for the quantification. Similarly, in Figure 4.b) we illustrate the average of the SPD matrices extracted from the third BiRe block. It is observed that the Riemannian module effectively enriches the relationship between different features, while reducing the importance of others. Finally, we measure the average Riemannian distance between the two classes in the test set during training, see Figure 4 .c). Here can be seen a growing trend of the distance, showing the ability of the network to learn Riemannian mappings that separate the SPD matrices, which traduces in a discriminative method. This result evidences the contribution of the Riemannian structures to the discrimination between control and PD patients.

5 Conclusions and Future Work

This paper introduced a Riemannian deep model with the capability to recover Parkinsonian oculomotor patterns under a classificatin task scheme. For this purpose, video sliced data was provided to represent ocular fixation abnormalities. The whole deep representation integrates convolutional and Riemannian modules to effectively discriminate between Parkinson and Control population. In a clinical scenario with few observations, the use of intermediate SPD representations produce stable results considering inter-patients variability. Furthermore, the ConvSPD improves classification performance using SPD structures and results in a lightweight model able to accurately predict in a reasonable inference time. Future works include new experiments with larger datasets and a backward analysis to discover potential relationships that may explain the dissease.

Acknowledgments

The recorded dataset was possible thanks to the support of the Parkinson foundation FAMPAS and the institution *Asilo San Rafael*. Additionally, acknowledgments to the Universidad Industrial de Santander for supporting this research registered by the project: *Cuantificación de patrones locomotores para el diagnóstico y seguimiento remoto en zonas de dificil acceso*, with SIVIE code 2697.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press (2009)
2. Acharya, D., Huang, Z., Pani Paudel, D., Van Gool, L.: Covariance pooling for facial expression recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 367–374 (2018)

3. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: Going beyond Euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017). https://doi.org/10.1109/MSP.2017.2693418
4. Chan, F., Armstrong, I.T., Pari, G., Riopelle, R.J., Munoz, D.P.: Deficits in saccadic eye-movement control in Parkinson's disease. Neuropsychologia **43**, 784–796 (2005)
5. Dorsey, E.R., Elbaz, A., Nichols, E., Abd-Allah, F., Abdelalim, A., Adsuar, J.C., et al.: Global, regional, and national burden of Parkinson's disease, 1990–2016: a systematic analysis for the global burden of disease study 2016. The Lancet Neurology **17**(11), 939–953 (2018)
6. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: Advances in Neural Information Processing Systems. vol. 28 (2015)
7. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2414–2423 (2016)
8. Gitchel, G.T., Wetzel, P.A., Baron, M.S.: Pervasive ocular tremor in patients with Parkinson disease. Archives of Neurology (2012)
9. Guayacán, L.C., Rangel, E., Martínez, F.: Towards understanding spatio-temporal parkinsonian patterns from salient regions of a 3d convolutional network. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). pp. 3688–3691. IEEE (2020)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Huang, Z., Van Gool, L.: A Riemannian network for SPD matrix learning. Thirty-First AAAI Conference on Artificial Intelligence **31**(1) (2017)
12. Ionescu, C., Vantzos, O., Sminchisescu, C.: Matrix backpropagation for deep networks with structured layers. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2965–2973 (2015)
13. Li, P., Xie, J., Wang, Q., Zuo, W.: Is second-order information helpful for large-scale visual recognition? In: Proceedings of the IEEE international conference on computer vision. pp. 2070–2078 (2017)
14. Li, Y., Wang, N., Liu, J., Hou, X.: Demystifying neural style transfer. arXiv preprint arXiv:1701.01036 (2017)
15. Pennec, X., Fillard, P., Ayache, N.: A Riemannian framework for tensor computing. International Journal of computer vision **66**(1), 41–66 (2006)
16. Poewe, W., Seppi, K., Tanner, C.M., Halliday, G.M., Brundin, P., Volkmann, J., et al.: Parkinson disease. Nature Reviews Disease Primers **3**(1), 1–21 (Mar 2017)
17. Salazar, I., Pertuz, S., Contreras, W., Martínez, F.: A convolutional oculomotor representation to model Parkinsonian fixational patterns from magnified videos. Pattern Analysis and Applications **24**(2), 445–457 (2021)
18. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
19. Tolosa, E., Garrido, A., Scholz, S.W., Poewe, W.: Challenges in the diagnosis of Parkinson's disease. The Lancet Neurology **20**(5), 385–397 (2021)
20. Wang, Q., Xie, J., Zuo, W., Zhang, L., Li, P.: Deep CNNs meet global covariance pooling: Better representation and generalization. IEEE transactions on pattern analysis and machine intelligence (2020)
21. Wang, W., Lee, J., Harrou, F., Sun, Y.: Early detection of Parkinson's disease using deep learning and machine learning. IEEE Access **8**, 147635–147646 (2020)

22. Weil, R.S., Schrag, A.E., Warren, J.D., Crutch, S.J., Lees, A.J., Morris, H.R.: Visual dysfunction in Parkinson's disease. Brain: A Journal of Neurology **139**(11), 2827–2843 (Nov 2016)