Learning Recurrent High-order Statistics for Skeleton-based Hand Gesture Recognition

25th International Conference on Pattern Recognition (ICPR 2020), Milan, Italie, January 2021

Xuan Son Nguyen*, Luc Brun†, Olivier Lézoray† and Sébastien Bougleux†
*ETIS, Université Paris Seine, Université Cergy-Pontoise, ENSEA, CNRS, Cergy-Pontoise, France
†Normandie Univ, ENSICAEN, CNRS, UNICAEN, GREYC, 14000 Caen, France

***Abstract*—High-order statistics have been proven useful in the framework of Convolutional Neural Networks (CNN) for a variety of computer vision tasks. In this paper, we propose to exploit high-order statistics in the framework of Recurrent Neural Networks (RNN) for skeleton-based hand gesture recognition. Our method is based on the Statistical Recurrent Units (SRU), an un-gated architecture that has been introduced as an alternative model for Long-Short Term Memory (LSTM) and Gate Recurrent Unit (GRU). The SRU captures sequential information by generating recurrent statistics that depend on a context of previously seen data and by computing moving averages at different scales. The integration of high-order statistics in the SRU significantly improves the performance of the original one, resulting in a model that is competitive to state-of-the-art methods on the Dynamic Hand Gesture (DHG) dataset, and outperforms them on the First-Person Hand Action (FPHA) dataset.**

I. Introduction

Skeleton-based hand gesture recognition has been an active research topic in recent years with many potential applications in assisted living, human-robot interaction, sign language interpretation, smart home control interface. In this work, we focus on deep learning techniques [1], [2], [3], [4], [5], [6] for solving the task as they have shown superior performance over their counterparts based on hand-crafted features [7], [8], [9]. Recently, modeling feature distribution with high-order statistics in deep neural networks has been proven effective. While this idea has been proposed in a number of CNN-based approaches [10], [11], [12], very few RNN-based approaches [13] exploit statistical information higher than first-order. One of the main challenges with such approaches is that the use of high-order statistics results in data lying on a manifold. This requires carefully-designed operations taking into account the geometric structure of data processed by the network [14].

In this work, we propose an approach that combines the ideas of the SRU [15] for learning long term information in sequences and ST-TS-HGR-NET [16], recently introduced for hand gesture recognition. In particular, we introduce high-order statistics into the SRU for improving hand gesture recognition. The motivation to use the SRU is that it has a simple un-gated architecture while being competitive to more sophisticated LSTM and GRU alternatives [15]. This is particularly useful when we want to introduce high-order statistics into the model, since it will simplify the design of operations taking into account the geometric structure of the resulting data. Note that SRU has not been proposed for hand gesture recognition in previous works. Our model is designed to capture the temporal dependency of matrices encoding first-order and second-order statistics of hand joints' coordinates. Moreover, we consider learning statistics resulting from the "covariance" of those matrices that significantly improves the performance. As a result, our method is superior to SRU on the DHG and FPHA datasets. Compared to ST-TS-HGR-NET, our method gives competitive results on the DHG dataset and outperform it on the FPHA dataset while using far less parameters.

II. Related Works

A. Skeleton-Based Gesture Recognition

Existing approaches can be roughly classified into hand-crafted feature based approaches or deep learning approaches. Hand-crafted feature based approaches compute joint features from the relative position between pairs of joints [7], [17], [18], [8], the relative position of 4-tuples of joints [19], Gram matrices [20], or the rotations and translations describing the 3D geometric relationships between body parts [9]. Deep learning approaches based on CNN [21], [22], [23], [1], [24], [16], [2], RNN [3], [4] and LSTM [25], [5], [24], [6], [26] have demonstrated impressive results. Some works have shown the interest of deep learning on manifolds for action and hand gesture recognition, e.g., SPD manifolds [27], [16], Grassmann manifolds [14], and Lie groups [28]. Since a hand or a body skeleton can be naturally represented as a graph (see e.g., Fig. 1), deep learning on graphs has also been applied to skeleton-based action recognition [29], [30].

B. Deep Learning with Second-order Statistics

Modeling feature distribution with high-order statistics has been proven effective in hand-crafted feature based approaches for various vision tasks. Inspired by this idea, some recent CNNs [31], [11], [10], [32] generate image representations by performing second-order pooling (as a covariance matrix) after the last convolutional layers of the networks. They have shown to significantly outperform classical networks [33], [34], [35] based on first-order statistics (mean vector) for producing image representations. Pooling strategies combining first-order and second-order statistics [36], [16], [12] have also demonstrated better performance than those based only on first-order or second-order statistics. Wang et al. [37] proposed a second-order pooling method capable of capturing complex and non-unimodal feature distributions. This relaxes

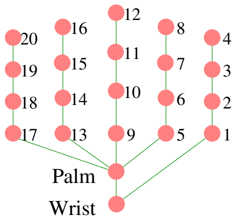


Fig. 1: Hand joints' positions.

the unimodal distribution assumption required by the above works and thus helps increasing the representation ability of CNNs. Kernel-based methods [38], [39], [40] reveal the link between high-order pooling and kernel machines, achieving compact covariance matrix-based representations. While most approaches introduced second-order pooling layers operating at the end of CNNs, Gao et al. [41] proposed a second-order pooling block that can be inserted after any convolutional layer. This allows CNNs to capture second-order statistics in intermediate layers. Similarly, the works of [42], [43], [44] focus on modeling feature interaction in intermediate layers of CNNs. Some works [13], [45], [46], [27] aim at learning covariance matrices that preserve the same geometric structure of the input data.

III. Statistical Recurrent Unit (SRU)

Oliva et al. [15] proposed an un-gated unit referred to as *the statistical recurrent unit* (SRU) for learning long term information in sequences. The SRU captures sequential information by generating recurrent statistics that depend on a context of previously seen data and by computing moving averages at different scales. Let $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T$ be an input sequence on $\mathbb{R}^n$. The update rules for the SRU are as follows:

$$\mathbf{R}_t = \text{ReLU}(\mathbf{W}_r \boldsymbol{\chi}_{t-1} + \mathbf{b}_r) \tag{1}$$

$$\mathbf{P}_t = \text{ReLU}(\mathbf{W}_p \mathbf{R}_t + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}_p) \tag{2}$$

$$\forall \alpha \in \mathcal{A}, \ \boldsymbol{\chi}_t^{(\alpha)} = \alpha \boldsymbol{\chi}_{t-1}^{(\alpha)} + (1-\alpha)\mathbf{P}_t \tag{3}$$

$$\mathbf{O}_t = \text{ReLU}(\mathbf{W}_o \boldsymbol{\chi}_t + \mathbf{b}_o) \tag{4}$$

where $\text{ReLU}(\mathbf{x}) = \max(\mathbf{x}, 0)$ is the standard linear rectifier, $\mathbf{W}_r, \mathbf{W}_p, \mathbf{W}_x, \mathbf{W}_o, \mathbf{b}_r, \mathbf{b}_p, \mathbf{b}_o$ are the parameters of the model, $\mathbf{R}_t$ encodes the features of averages, $\mathbf{P}_t$ are the statistics that are dependent not only on the current input $\mathbf{x}_t$ but also on $\mathbf{R}_t$, $\mathbf{O}_t$ is the output that is fed upwards in the network, $\mathcal{A}$ is the set of different scales, $\boldsymbol{\chi}_t$ is the moving average that summarizes statistics of the sequence up to frame t. The moving averages are concatenated over all scales $\boldsymbol{\chi}_t = \{\boldsymbol{\chi}_t^{(\alpha)}\}_{\alpha \in \mathcal{A}}$ and then used to create an output of the network at frame t.

Note that the moving averages are dependent not only on the current input but also on the statistics computed from values of previous points. This approach is different from treating the sequence as a set of i.i.d. points drawn from some distribution and marginalizing out time, which loses temporal information.

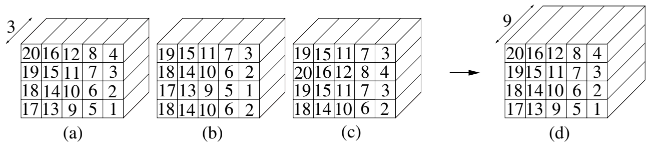


Fig. 2: Construction of our network input.

While the SRU is based on an un-gated architecture, it has been shown to be competitive with more sophisticated LSTM and GRU alternatives [15].

IV. Proposed Approach

We propose in Section IV-A a method for constructing our network input which takes into account the correlations of neighboring hand joints. In Section IV-B, a sequence model based on SRU and high-order statistics is then introduced. Based on the proposed model, we show how to build a network architecture for hand gesture recognition in Section IV-C. Finally, an interpretation of the model from a statistical point of view is provided in Section IV-D.

A. Network Input

In this work, we use skeletal data obtained by an Intel RealSense camera for hand gesture recognition. The hand joints' positions at each frame are illustrated in Fig. 1. Given a sequence of hand joints' 3D coordinates, we first remove the two joints at the palm and wrist (if any) and then renumber the joint indices from 1 to N ($N = 20$). The construction of our network input at a given frame t is illustrated in Fig. 2. Fig. 2(a) represents the 3-dimensional array containing the hand joints' 3D coordinates at frame t where each number indicates a joint index. Each joint $i, i = 1, \dots, N$ belonging to a finger has at most two neighboring joints belonging to the same finger, referred to as the lower and upper neighbors of i. For a hand joint with no lower neighbor (1,5,9,13,17) or no upper neighbor (4,8,12,16,20), we duplicate its unique neighbor so that it has two neighbors. The 3-dimensional array in Fig. 2(b) is constructed by replacing the coordinates of each joint in Fig. 2(a) by those of its lower neighbor. Similarly, the 3-dimensional array in Fig. 2(c) is constructed by replacing the coordinates of each joint in Fig. 2(a) by those of its upper neighbor. Our network input at frame t is then formed by concatenating the three 3-dimensional arrays along the depth dimension (Fig. 2(d)). Thus, the feature vector of a hand joint has 9 dimensions. This method for constructing the network input allows to take into account the correlations of neighboring hand joints and has been shown to greatly improve recognition accuracy (see Section V).

B. SRU Based on High-order Statistics (SRU-HOS)

Differently from previous approaches that are based only on first-order and second-order statistics, we propose to combine those statistics with higher-order statistics computed from

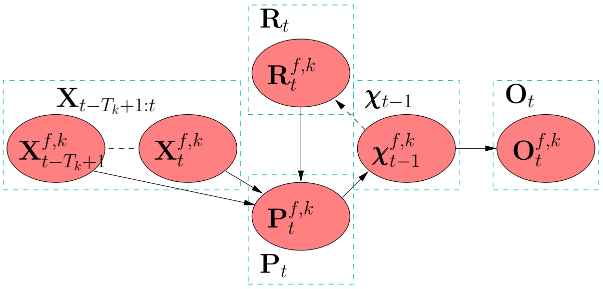


Fig. 3: Graphical representation of our model. Solid lines indicate a dependence on the current value of a node. Dashed lines indicate a dependence on the previous value of a node. Here, $T_1 = t_1$, and $T_2 = t_1 + t_2 - 1$.

them. For each finger $f = 1, \dots, 5$, we consider two statistics $(\boldsymbol{\chi}_t^{f,k})_{k=1,2}$ at frame t that are updated using the following equations (see Fig. 3):

$$\mathbf{R}_t^{f,k} = \text{ReEig}\left(\sum_{\alpha \in \mathcal{A}} \frac{(w_r^{k,\alpha})^2}{\sum_{\alpha \in \mathcal{A}} (w_r^{k,\alpha})^2} \boldsymbol{\chi}_{t-1}^{f,k,\alpha} \right) \tag{5}$$

$$\begin{aligned} \mathbf{P}_t^{f,k} = \text{ReEig}\Bigg(& \frac{(w_p^k)^2}{(w_p^k)^2 + (w_x^k)^2} \mathbf{R}_t^{f,k} + \\ & \frac{(w_x^k)^2}{(w_p^k)^2 + (w_x^k)^2} h^k(\mathbf{X}_t^f) \Bigg) \end{aligned} \tag{6}$$

$$\forall \alpha \in \mathcal{A}, \quad \boldsymbol{\chi}_t^{f,k,\alpha} = \alpha \boldsymbol{\chi}_{t-1}^{f,k,\alpha} + (1 - \alpha) \mathbf{P}_t^{f,k} \tag{7}$$

$$\mathbf{O}_t^{f,k} = \text{ReEig}\left(\sum_{\alpha \in \mathcal{A}} \frac{(w_o^{k,\alpha})^2}{\sum_{\alpha \in \mathcal{A}} (w_o^{k,\alpha})^2} \boldsymbol{\chi}_t^{f,k,\alpha} \right), \tag{8}$$

where $\mathcal{A}$ is a set of different scales, $w_r^{k,\alpha}, w_p^k, w_x^k, w_o^{k,\alpha} \in \mathbb{R}$ are the parameters of the model, $\mathbf{X}_t^f \in \mathbb{R}^{9 \times |J_f|}$ is the matrix at frame t representing the 3D coordinates of hand joints belonging to finger f (Section IV-A), J_f is the set of hand joints belonging to finger f, $\boldsymbol{\chi}_0^{f,k,\alpha}$ are the initial states, $h^1(\mathbf{X}_t^f)$ and $h^2(\mathbf{X}_t^f)$ compute statistics from frames $t - t_1 + 1$ to t and frames $t - t_1 - t_2 + 2$ to t, $t_1, t_2 > 0$ are two constants, and $\text{ReEig}(\mathbf{Y}) = \mathbf{U} \max(\epsilon \mathbf{I}, \mathbf{V}) \mathbf{U}^T$, where $\mathbf{Y}$ is a symmetric positive definite (SPD) matrix and $\mathbf{Y} = \mathbf{U}\mathbf{V}\mathbf{U}^T$ its eigen-decomposition, ϵ is a rectification threshold, $\mathbf{I}$ is the identity matrix, $\max(\epsilon \mathbf{I}, \mathbf{V})$ is a diagonal matrix whose diagonal elements are defined as:

$$(\max(\epsilon \mathbf{I}, \mathbf{V}))(i,i) = \begin{cases} \mathbf{V}(i,i) & \text{if } \mathbf{V}(i,i) > \epsilon \\ \epsilon & \text{if } \mathbf{V}(i,i) \leq \epsilon. \end{cases} \tag{9}$$

The function ReEig was proposed in [27] for introducing non-linear transformation of SPD matrices. Let $\mathbf{X}_t = [\mathbf{x}_t^1, \dots, \mathbf{x}_t^N]$, where $\mathbf{x}_t^i \in \mathbb{R}^9$ represents the feature vector of joint i at frame t, $i = 1, \dots, N$. Assuming that $\mathbf{x}_j^i, i \in J_f, j = t - t_1 + 1, \dots, t$ are independent and identically distributed samples from the following Gaussian distribution:

$$\begin{aligned} & \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_t^{f,1}, \boldsymbol{\Sigma}_t^{f,1}) = \\ & \frac{1}{|2\pi \boldsymbol{\Sigma}_t^{f,1}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_t^{f,1})^T (\boldsymbol{\Sigma}_t^{f,1})^{-1} (\mathbf{x} - \boldsymbol{\mu}_t^{f,1}) \right), \end{aligned} \tag{10}$$

where $|.|$ is the determinant, $\boldsymbol{\mu}_t^{f,1}$ is the mean vector and $\boldsymbol{\Sigma}_t^{f,1}$ is the covariance matrix:

$$\boldsymbol{\mu}_t^{f,1} = \frac{1}{|J_f| t_1} \sum_{j=t-t_1+1}^{t} \sum_{i \in J_f} \mathbf{x}_j^i,$$

$$\boldsymbol{\Sigma}_t^{f,1} = \frac{1}{|J_f| t_1} \sum_{j=t-t_1+1}^{t} \sum_{i \in J_f} (\mathbf{x}_j^i - \boldsymbol{\mu}_t^{f,1})(\mathbf{x}_j^i - \boldsymbol{\mu}_t^{f,1})^T.$$

Then $h^1(\mathbf{X}_t^f)$ is designed to capture the Gaussian distribution $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_t^{f,1}, \boldsymbol{\Sigma}_t^{f,1})$ [47]:

$$h^1(\mathbf{X}_t^f) = \begin{bmatrix} \boldsymbol{\Sigma}_t^{f,1} + \boldsymbol{\mu}_t^{f,1} (\boldsymbol{\mu}_t^{f,1})^T & \boldsymbol{\mu}_t^{f,1} \\ (\boldsymbol{\mu}_t^{f,1})^T & 1 \end{bmatrix}. \tag{11}$$

By introducing $h^1(\mathbf{X}_t^f)$ in Eq. (6), the proposed model learns the first-order (mean) and second-order (covariance) statistics of the hand joints' feature vector. To strengthen the learning capabilities of the model, it also learns higher-order statistic $h^2(\mathbf{X}_t^f)$ defined by the "covariance" of $h^1(\mathbf{X}_t^f)$. Since $h^1(\mathbf{X}_t^f)$ lies on the manifold of SPD matrices, it is projected onto its tangent space before the covariance can be measured. The projection is performed by the matrix function $\log_p(\mathbf{Y}) = \mathbf{U} \log(\mathbf{V}) \mathbf{U}^T$ [48], where $\mathbf{Y}$ is a SPD matrix and $\mathbf{Y} = \mathbf{U}\mathbf{V}\mathbf{U}^T$ is its eigen-decomposition. Then the statistic $h^2(\mathbf{X}_t^f)$ is defined as:

$$\boldsymbol{\mu}_t^{f,2} = \tfrac{1}{t_2} \sum_{i=t-t_2+1}^{t} \text{vl}(\text{h}^1(\mathbf{X}_i^f)), \tag{12}$$

$$\begin{aligned} h^2(\mathbf{X}_t^f) = \tfrac{1}{t_2} \sum_{i=t-t_2+1}^{t} & (\text{vl}(\text{h}^1(\mathbf{X}_i^f)) - \boldsymbol{\mu}_t^{f,2}) \\ & \cdot (\text{vl}(\text{h}^1(\mathbf{X}_i^f)) - \boldsymbol{\mu}_t^{f,2})^T, \end{aligned} \tag{13}$$

where $\text{vl}(\cdot) = \text{vec}(\log(\cdot))$, $\text{vec} : \mathbb{R}^{d \times d} \to \mathbb{R}^{d(d+1)/2}$ the special vectorization operator [49] that transforms any symmetric matrix $\mathbf{Y} \in \mathbb{R}^{d \times d}$ into a vector $\text{vec}(\mathbf{Y}) = [\mathbf{Y}(u,u), (\sqrt{2}\mathbf{Y}(u,v))_{v=u+1,\dots,d}]_{u=1,\dots,d}^T$.

Let M be the index of the last frame of the sequence, $\mathbf{X}^f$ be the matrix representing the 3D coordinates of hand joints belonging to finger f, $\mathbf{O}_M^{f,k}$, $k = 1, 2$ be the final outputs of our model. We can then write the final outputs of our model as a function of the input and the model parameters as follows:

$$\begin{aligned} \{\mathbf{O}_M^{f,k}\}_{k=1,2} = & \text{SRU-HOS}(\mathbf{X}^f, \{\boldsymbol{\chi}_0^{f,k,\alpha}\}_{k=1,2,\alpha \in \mathcal{A}}, \\ & \{w_p^k\}_{k=1,2}, \{w_x^k\}_{k=1,2}, \{w_r^{k,\alpha}\}_{k=1,2,\alpha \in \mathcal{A}}, \\ & \{w_o^{k,\alpha}\}_{k=1,2,\alpha \in \mathcal{A}}) \end{aligned} \tag{14}$$

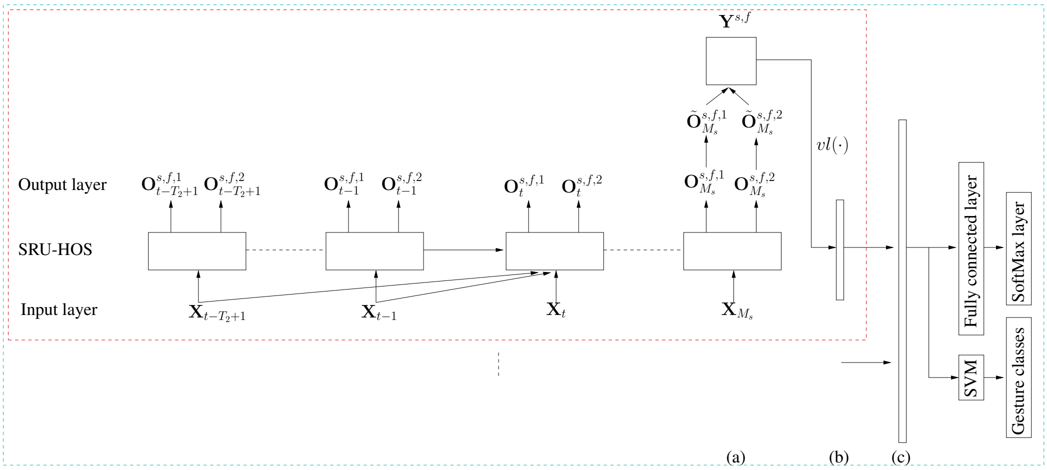


Fig. 4: The proposed network architecture. The network consists of 30 SRU-HOS, each of them produces statistics for a sub-sequence s, $s = 1, \dots, 6$ associated with a finger f, $f = 1, \dots, 5$. (a) The final outputs of each SRU-HOS are combined into $\mathbf{Y}^{s,f}$ (b) $\mathbf{Y}^{s,f}$ is then transformed into an Euclidean space and vectorized to give a representation of sub-sequence s associated with finger f. (c) All vectors at step (b) are concatenated to obtain a global representation of the sequence. This representation is fed to a fully connected layer and then a softmax layer. For efficient gesture recognition, we use the vectors obtained at step (c) to train a SVM classifier.

C. *The Proposed Network*

Our proposed network is illustrated in Fig. 4. To better capture temporal dependencies of a skeleton sequence, we create six sub-sequences by dividing the original sequence into one sequence, then two sequences of equal length, and three sequences of equal length. For each sub-sequence $s, s = 1, \dots, 6$, and a finger $f, f = 1, \dots, 5$, we create a model for learning statistics from s associated with f. Denote by $\mathbf{X}^{s,f}$ the matrix representing the 3D coordinates of hand joints belonging to finger within sub-sequence s, $\boldsymbol{\chi}_0^{s,f,k,\alpha}$ are the initial states, $w_r^{s,k,\alpha}, w_p^{s,k}, w_x^{s,k}, w_o^{s,k,\alpha} \in \mathbb{R}$ are the parameters of the model. Then the final outputs of the model can be given by:

$$\begin{aligned}\{\mathbf{O}^{s,f,k}_{M_s}\}_{k=1,2} = &\text{SRU-HOS}(\mathbf{X}^{s,f}, \{\boldsymbol{\chi}_0^{s,f,k,\alpha}\}_{k=1,2,\alpha\in\mathcal{A}},\\ &\{w_p^{s,k}\}_{k=1,2}, \{w_x^{s,k}\}_{k=1,2},\\ &\{w_r^{s,k,\alpha}\}_{k=1,2,\alpha\in\mathcal{A}},\\ &\{w_o^{s,k,\alpha}\}_{k=1,2,\alpha\in\mathcal{A}})\end{aligned} \tag{15}$$

For a more accurate classification, the outputs $\mathbf{O}^{s,f,1}_{M_s} \in \mathbb{R}^{10\times10}$ and $\mathbf{O}^{s,f,2}_{M_s} \in \mathbb{R}^{55\times55}$ (Eq. (8)) are tuned w.r.t. parameters $\mathbf{W}^{s,f,1} \in \mathbb{R}^{10\times10}$ and $\mathbf{W}^{s,f,2} \in \mathbb{R}^{55\times55}$:

$$\tilde{\mathbf{O}}^{s,f,1}_{M_s} = \mathbf{W}^{s,f,1}\mathbf{O}^{s,f,1}_{M_s}(\mathbf{W}^{s,f,1})^T, \tag{16}$$

$$\tilde{\mathbf{O}}^{s,f,2}_{M_s} = \mathbf{W}^{s,f,2}\mathbf{O}^{s,f,2}_{M_s}(\mathbf{W}^{s,f,2})^T. \tag{17}$$

Both matrices $\mathbf{W}^{s,f,k}$, $k = 1, 2$ have full row-rank so that $\tilde{\mathbf{O}}^{s,f,k}_{M_s}$ is SPD as $\mathbf{O}^{s,f,k}_{M_s}$. These weights can be optimized using a variant of stochastic gradient descent on Stiefel manifolds [27]. Since $\mathbf{O}^{s,f,2}_{M_s}$ encodes the "covariance" of $h^1(\mathbf{X}^{s,f}_t)$, these outputs are grouped into the matrix $\mathbf{Y}^{s,f}$ to obtain the final representation of sub-sequence s associated with finger f as follows:

$$\mathbf{Y}^{s,f} = \begin{bmatrix} \tilde{\mathbf{O}}^{s,f,2}_{M_s} + \text{vl}(\tilde{\mathbf{O}}^{s,f,1}_{M_s})\text{vl}(\tilde{\mathbf{O}}^{s,f,1}_{M_s})^T & \text{vl}(\tilde{\mathbf{O}}^{s,f,1}_{M_s}) \\ \text{vl}(\tilde{\mathbf{O}}^{s,f,1}_{M_s})^T & 1 \end{bmatrix}, \tag{18}$$

We transform $\mathbf{Y}^{s,f}$ into an Euclidean space and then vectorize the resulting matrix using the $vl(.)$ function to create a representation of sub-sequence s associated with finger f. A global representation of the original sequence is obtained by concatenating all resulting vectors for $s = 1, \dots, 6$ and $f = 1, \dots, 5$, i.e. $[\text{vl}(\mathbf{Y}^{1,1})^T, \dots, \text{vl}(\mathbf{Y}^{6,5})^T]^T$. Finally, the network is trained by passing this vector to a fully connected layer and by minimizing cross-entropy (see Fig. 4).

It is worth pointing out the differences between our network and ST-TS-HGR-NET. First, both networks capture weak orders of frames by computing statistics from six temporal sub-sequences and then combining them to represent the original sequence. However, each sub-sequence in ST-TS-HGR-NET is represented by the mean and covariance computed from statistics of all frames of the sub-sequence without taking into account the temporal order of frames. In contrast, this order is used in Eq. (6) of our network where $\mathbf{P}^{s,f,k}_t$ is computed from the current input $\mathbf{X}^{s,f}_t$ and the statistics of the previous data $\mathbf{R}^{s,f,k}_t$. Thus, temporal information could be better preserved by our network than ST-TS-HGR-NET. Second, in our network, different sets of weights are learned for different sub-sequences but these weights are shared for different fingers. In ST-TS-HGR-NET, a set of weights is learned for each sub-sequence associated with each finger. This results in a large set of parameters to be trained compared to our network (see Section V-C).

D. Interpretation of the Model

From a statistical point of view, our model can be interpreted as an extension of the SRU model. Let $(\mathbf{y}_t)_{t=1,\dots,M}$ be an input sequence of real-valued points and let $(\gamma_t)_{t=1,\dots,M}$ be the statistics on these points. In a SRU, the statistic γ_t on the t-th point $\mathbf{y}_t$ is defined as a function of $\mathbf{y}_t$ and the statistic γ_{t-1} on the previous point: $\gamma_t = \gamma(\mathbf{y}_t, \gamma_{t-1})$ with γ_0 an initial constant vector. In our model, two statistics γ_t^1 and γ_t^2 are computed for $t = t_1 + t_2 - 1, \dots, M$. Each statistic is defined as a function of the statistic on the previous point, and the statistic generated by a function h or g from $\mathbf{y}_t$ and previous points:

$$\gamma_t^1 = \gamma(h(\mathbf{y}_{t-t_1+1}, \dots, \mathbf{y}_t), \gamma_{t-1}^1), \tag{19}$$

$$\begin{aligned}\gamma_t^2 = \gamma(g(h(&\mathbf{y}_{t-t_1-t_2+2}, \dots, \mathbf{y}_{t-t_2+1}), \dots, \\ &h(\mathbf{y}_{t-t_1+1}, \dots, \mathbf{y}_t)), \gamma_{t-1}^2).\end{aligned} \tag{20}$$

As with an SRU, the statistics produced by Eq. (19) can perfectly encode a sequence if our model is given enough dimensions. For example, consider a sequence $y_1, \dots, y_M \in \mathbb{R}^+$ and statistics $\gamma_t = (0, \dots, 0)$ for $t = 1, \dots, t_1 - 1$ and $\gamma_t = (0, \dots, \frac{M}{t_1} y_{t-t_1+1}, \dots, \frac{M}{t_1} y_t, 0, \dots)$ for $t = t_1, \dots, M$. The values at indices $t - t_1 + 1, \dots, t$ are $\frac{M}{t_1} y_{t-t_1+1}, \dots, \frac{M}{t_1} y_t$, respectively. The average of these statistics is given by:

$$\begin{aligned}\frac{1}{M}\sum_{t=1}^{M}\gamma_t =& (\tfrac{1}{t_1} y_1, \tfrac{2}{t_1} y_2, \dots, \tfrac{t_1-1}{t_1} y_{t_1-1}, y_{t_1}, \dots, y_{M-t_1+1}, \\ &\tfrac{t_1-1}{t_1} y_{M-t_1+2}, \dots, \tfrac{1}{t_1} y_M).\end{aligned} \tag{21}$$

The average can be used to recover the original sequence. Since the statistics generated by h depend on the current and previous points, the number of samples for the estimate of $\mathbf{\Sigma}_t^{s,f,1}$ is increased. It is particularly useful in our method to avoid ill-conditioned matrices (Eq. (11)).

V. Experiments

Our network, referred to as SRU-HOS-NET, is validated on the Dynamic Hand Gesture (DHG) dataset [17], [50] and the First-Person Hand Action (FPHA) dataset [51]. The batch size and the learning rate were set to 40 and 0.9, respectively. The set of values for α was set similarly to [15], i.e., $\mathcal{A} = \{0.01, 0.25, 0.5, 0.9, 0.99\}$. The rectification threshold for the ReEig$(\cdot)$ function was set to 10^{-4} [27]. For sequences having more or less than 150 frames, we used interpolation in order to fix the number of frames of all sequences to 150. Gesture recognition was performed by training a SVM classifier using the output of SRU-HOS-NET (the vector obtained before the fully connected layer) at epoch 50 (see Fig. 4). In our experiments, we found that this method for classifying gestures gave competitive results compared to those obtained by training the network with a softmax layer. Indeed, if one uses a classical FC/SoftMax Layer for the classification, the networks usually converges in more than 200 epochs. However similar results can be obtained with a SVM taking the output of the network trained during four times less epochs. We used the LIBLINEAR library [52] to train our SVM classifier with L2-regularized L2-loss (dual) and default parameter settings.

Dataset	t_1			t_2			**Feature concatenation**	
	3	10	20	10	15	20	No	Yes
DHG (14 gestures)	92.62	93.93	**94.40**	93.21	**94.40**	94.05	85.36	**94.40**
DHG (28 gestures)	87.98	88.57	**89.52**	88.45	**89.52**	89.29	78.09	**89.52**
FPHA	**94.61**	93.56	92.52	93.57	**94.61**	94.26	83.48	**94.61**

TABLE I: Ablation analysis of our network. The best result in each row of a sub-table (t_1, t_2, **Feature concatenation**) is marked in bold.

A. Datasets and Experimental Protocols

DHG dataset. It contains 14 gestures performed in two ways: using one finger and the whole hand. Each gesture is executed several times by different actors. The gestures are subdivided into categories of fine and coarse : grab (fine), tap (coarse), expand (fine), pinch (fine), rotation clockwise (rot-cw, fine), rotation counterclockwise (rot-ccw, fine), swipe right (swipe-r, coarse), swipe left (swipe-l, coarse), swipe up (swipe-u, coarse), swipe down (swipe-d, coarse), swipe x (swipe-x, coarse), swipe v (swipe-v, coarse), swipe + (swipe-+, coarse), shake (coarse). The dataset provides the 3D coordinates of 22 hand joints as illustrated in Fig. 1. It has been split into 1960 train sequences (70% of the dataset) and 840 test sequences (30% of the dataset) [50].

FPHA dataset. This dataset contains 1175 action videos belonging to 45 different action categories, in 3 different scenarios, and performed by 6 actors. The action categories are: charge cell phone, clean glasses, close juice bottle, close liquid soap, close milk, close peanut butter, drink mug, flip pages, flip sponge, give card, give coin, handshake, high five, light candle, open juice bottle, open letter, open liquid soap, open milk, open peanut butter, open soda can, open wallet, pour juice bottle, pour liquid soap, pour milk, pour wine, prick, put salt, put sugar, put tea bag, read letter, receive coin, scoop spoon, scratch sponge, sprinkle, squeeze paper, squeeze sponge, stir, take letter from envelope, tear paper, toast wine, unfold glasses, use calculator, use flash, wash sponge, write. Action sequences present high inter-subject and intra-subject variability of style, speed, scale, and viewpoint. The dataset provides the 3D coordinates of 21 hand joints as DHG dataset except for the palm joint. We used the 1:1 setting proposed in [51] with 600 action sequences for training and 575 for testing.

B. Ablation Study

In this section, we investigate the impact of different components on our network's performance. We set the default values of t_2 to 15.

Statistics	DHG (14 gestures)		DHG (28 gestures)		FPHA	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
$h^1(.)$	85.00	85.00	76.43	76.03	77.04	76.80
$h^2(.)$	**89.29**	**89.16**	**86.07**	**85.81**	**93.57**	**93.32**

TABLE II: Effectiveness of $h^1(.)$ and $h^2(.)$. The best result in each column is marked in bold.

Model	Number of parameters
ST-TS-HGR-NET	672,243
SRU-HOS-NET	**18,894**

TABLE III: Comparison on network complexity.

Method	Year	Color	Depth	Pose	RNN/LSTM	Accuracy (%)	
						14 gestures	28 gestures
HON4D [53]	2013	✗	✓	✗	✗	78.53	74.03
Devanne et al. [54]	2015	✗	✗	✓	✗	79.61	62.00
Huang et al. [27]	2017	✗	✗	✓	✗	75.24	69.64
De Smedt et al. [17]	2016	✗	✗	✓	✗	88.24	81.90
Devineau et al. [21]	2018	✗	✗	✓	✗	91.28	84.35
SRU [15]	2018	✗	✗	✓	✓	82.02	76.31
SRU-SPD [13]	2018	✗	✗	✓	✓	86.31	80.83
ST-TS-HGR-NET [16]	2019	✗	✗	✓	✗	94.29	89.40
SRU-HOS-NET		✗	✗	✓	✓	**94.40**	**89.52**

TABLE IV: Performance of our method and state-of-the-art methods on DHG dataset. The best result in each column is marked in bold.

Time windows t_1. Tab. I (column t_1) shows the impact of the time window t_1 on SRU-HOS-NET, where t_1 is set to $t_1 = 3, 10, 20$. As can be observed, SRU-HOS-NET achieves the best results with $t_1 = 20$ and $t_1 = 3$ on DHG and FPHA datasets, respectively. Results show that combining the current point with a sufficient number of previous points for estimating $h^1(\cdot)$ can lead to non-trivial performance gain. In the following, we set t_1 to 20 and 3 for experiments on DHG and FPHA datasets, respectively.

Time windows t_2. The impact of the time window t_2 on SRU-HOS-NET is shown in Tab. I (column t_2). Here we test with three different settings of t_2: $t_2 = 10, 15, 20$. For both the datasets, SRU-HOS-NET gives the best results with $t_2 = 15$.

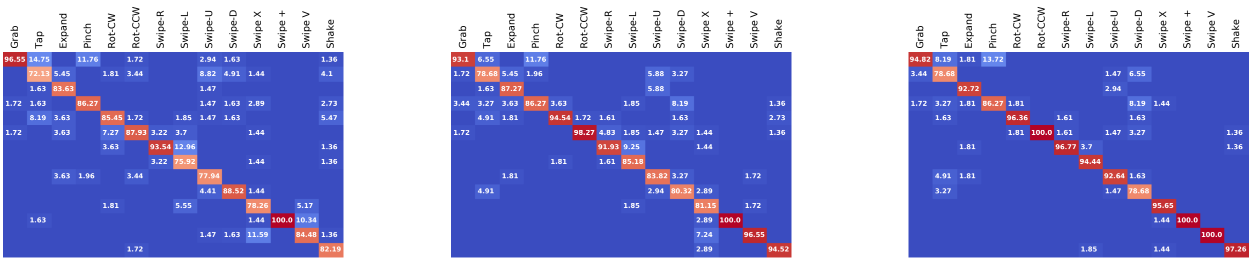
Network input. The impact of the feature concatenation method (section IV-A) can be seen in Tab. I (column **Feature concatenation**), where the performance of SRU-HOS-NET drops significantly when feature concatenation is not used, i.e., the hand joints' coordinates are fed directly as input of SRU-HOS-NET. Results indicate that feature concatenation is crucial for obtaining good performance in our method.

Effectiveness of $h^1(\cdot)$ and $h^2(\cdot)$. These experiments are conducted to study the impact of $h^1(\cdot)$ and $h^2(\cdot)$ in our model. In the first experiment, we remove $h^1(\cdot)$ from our model to evaluate the performance of $h^2(\cdot)$. In this case, the output of SRU-HOS-NET before the fully connected layer is: $[(\text{vec}(\log(\tilde{\mathbf{O}}_{M_1}^{1,1,2})))^T, \dots, (\text{vec}(\log(\tilde{\mathbf{O}}_{M_6}^{6,5,2})))^T]^T$. In the second experiment, we remove $h^2(\cdot)$ from our model to evaluate the performance of $h^1(\cdot)$. The output of SRU-HOS-NET before the fully connected layer is: $[(\text{vec}(\log(\tilde{\mathbf{O}}_{M_1}^{1,1,1})))^T, \dots, (\text{vec}(\log(\tilde{\mathbf{O}}_{M_6}^{6,5,1})))^T]^T$. Results of these experiments are given in Tab. II. Note that the accuracy of SRU-HOS-NET degrades significantly when $h^2(\cdot)$ is not used. Specifically, the accuracies of SRU-HOS-NET are decreased by 9.40, 13.09, and 17.57 percent points on DHG dataset with 14 and 28 gestures and FPHA dataset, respectively. The confusion matrices of SRU-HOS-NET on DHG dataset when using $h^1(.)$, $h^2(.)$, and both $h^1(.)$ and $h^2(.)$ are given in Fig. 5. When using only $h^1(.)$, SRU-HOS-NET gives more than 90% accuracy for the gestures 'grab', 'swipe-r', and 'swipe-+'. When using only $h^2(.)$, SRU-HOS-NET gives more than 90% accuracy for the gestures 'grab', 'swipe-r', 'swipe-+', 'rot-cw', 'rot-ccw', 'swipe v', and 'shake'. Thus, $h^2(.)$ captures more discriminative information than $h^1(.)$ for recognizing gestures with more than 90% accuracy. Moreover, when using both $h^1(.)$ and $h^2(.)$, SRU-HOS-NET gives more than 90% accuracy for the gestures 'grab', 'swipe-r', 'swipe-+', 'rot-cw', 'rot-ccw', 'swipe v', 'shake', 'expand', 'swipe-l', 'swipe-u', and 'swipe-x'. Thus, both $h^1(.)$ and $h^2(.)$ contribute to the overall performance of SRU-HOS-NET. We can also remark that the difference in performance between $h^1(.)$ et $h^2(.)$ on FPHA dataset is larger than that on DHG dataset. Note that the DHG dataset contains a set of coarse gestures that are mainly distinguished by the global movement of the hand, while all gestures of the FPHA dataset involve both the global movement of the hand and that of the fingers. This indicates that higher-order statistics could be particularly useful for fine-grained gesture recognition.

C. Results on DHG dataset

The comparison of our method and state-of-the-art methods on DHG dataset is given in Tab. IV. For SRU and the method of [13], referred to as SRU-SPD, we ran the codes from[1,2] with the parameter settings established by the authors for obtaining their accuracies. SRU-SPD is a variant of SRU for SPD matrices, which is based on the concept of weighted Fréchet mean of SPD matrices and the Cholesky factorization for parametrization of SPD matrices. As can be observed in Tab. IV, SRU-HOS-NET and ST-TS-HGR-NET are competitive and give the best results on two evaluation protocols (14 and 28 gestures). However, in terms of network complexity, SRU-HOS-NET has far less parameters than ST-TS-HGR-NET (Tab. III). Here, the number of parameters of ST-TS-HGR-NET is about 35 times more than that of SRU-HOS-NET. Note that SRU-HOS-NET and SRU-SPD significantly outperform SRU, demonstrating the superior performance of combining first-order and second-order statistics over classical first-order statistics on this dataset. While both SRU-HOS-NET and SRU-SPD are based on the formulation of SRU, SRU-HOS-NET outperforms SRU-SPD by large margins, i.e., 8.09 and 8.69 percent points on DHG dataset with 14 and 28 gestures, respectively. This probably can be explained by the fact that (1) Our method for constructing the network input by taking into account the correlation of neighboring hand joints is effective, and (2) Similarly to SRU-SPD, SRU-HOS-NET captures first-order and second-order statistics of

[1] https://github.com/junieroliva/recurrent
[2] https://github.com/zhenxingjian/SPD-SRU



(a) $h^1(.)$

(b) $h^2(.)$

(c) $h^1(.)$ and $h^2(.)$

Fig. 5: The confusion matrices of SRU-HOS-NET on DHG dataset using (a) $h^1(.)$, (b) $h^2(.)$, and (c) both $h^1(.)$ and $h^2(.)$.

Method	Year	Color	Depth	Pose	RNN/LSTM	Accuracy (%)
HON4D [53]	2013	✗	✓	✗	✗	70.61
Novel View [55]	2016	✗	✓	✗	✗	69.21
1-layer LSTM [56]	2016	✗	✗	✓	✓	78.73
2-layer LSTM [56]	2016	✗	✗	✓	✓	80.14
Moving Pose [57]	2013	✗	✗	✓	✗	56.34
Lie Group [9]	2014	✗	✗	✓	✗	82.69
HBRNN [3]	2015	✗	✗	✓	✓	77.40
Gram Matrix [20]	2016	✗	✗	✓	✗	85.39
TF [58]	2017	✗	✗	✓	✗	80.69
JOULE-color [59]	2015	✓	✗	✗	✗	66.78
JOULE-depth [59]	2015	✗	✓	✗	✗	60.17
JOULE-pose [59]	2015	✗	✗	✓	✗	74.60
JOULE-all [59]	2015	✓	✓	✓	✗	78.78
Huang et al. [27]	2017	✗	✗	✓	✗	84.35
Huang et al. [14]	2018	✗	✗	✓	✗	77.57
SRU [15]	2018	✗	✗	✓	✓	72.17
SRU-SPD [13]	2018	✗	✗	✓	✓	78.96
ST-TS-HGR-NET [16]	2019	✗	✗	✓	✗	93.22
SRU-HOS-NET		✗	✗	✓	✓	**94.61**

TABLE V: Performance of our method and state-of-the-art methods on FPHA dataset. The best result is marked in bold.

hand joints' coordinates via $h^1(\cdot)$. However, SRU-HOS-NET captures richer statistics than SRU-SPD thanks to $h^2(\cdot)$, which can be interpreted as second-order statistics computed from $h^1(\cdot)$. The method of [27] also exploits high-order statistics in the framework of CNNs. However, like SRU-SPD, it is only based on the covariance information of hand joints' coordinates. As a result, it performs significantly worse than SRU-HOS-NET, i.e., 19.16 and 19.88 percent points on DHG dataset with 14 and 28 gestures, respectively.

D. Results on FPHA dataset

The comparison of our method and state-of-the-art methods on FPHA dataset is given in Tab. V. From this table, we can make the following observations. First, SRU-HOS-NET achieves the best result on this dataset. In particular, it outperforms the second best method ST-TS-HGR-NET by 1.39 percent point. Second, the performances of the sequential models of [56], [3] are largely behind our model. The best model among them is a 2-layer LSTM, that is 14.47 percent points less accurate than our model. Note that SRU is significantly outperformed by the models of [56], [3]. This is probably because those models are specially designed for skeleton-based action recognition. For example, in [56], the authors have introduced a co-occurrence regularization term into the loss function and an in-depth dropout mechanism for the task of action recognition based on LSTMs. This result also indicates that SRU does not give good performance for hand gesture recognition on this dataset. However, the integration of high-order statistics into SRU leads to state-of-the-art performance on this dataset. Third, compared to SRU-SPD and the networks of [27] that also exploit high-order statistics, SRU-HOS-NET significantly outperforms them by at least 10.26 percent points. Fourth, the performance gaps between SRU-HOS-NET and SRU-SPD (15.65 percent points) and between SRU-HOS-NET and SRU (22.44 percent points) are even more pronounced on this dataset.

VI. Conclusion

We have proposed a new RNN model for skeleton-based hand gesture recognition that integrates high-order statistics in the SRU for learning a discriminative hand gesture representation. We have evaluated the proposed method on two benchmark hand gesture datasets. On the DHG dataset, the proposed method is competitive to state-of-the-art methods, while having far less number of parameters than the best method among them. On the FPHA dataset, the proposed method outperforms state-of-the-art methods by at least 1.39 percent.

References

[1] M. Liu and J. Yuan, "Recognizing Human Actions as The Evolution of Pose Estimation Maps," in *CVPR*, 2018.

[2] P. Wang, Z. Li, Y. Hou, and W. Li, "Action Recognition Based on Joint Trajectory Maps Using Convolutional Neural Networks," in *ACM MM*, 2016, pp. 102–106.

[3] Y. Du, W. Wang, and L. Wang, "Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition," in *CVPR*, 2015, pp. 1110–1118.

[4] H. Wang and L. Wang, "Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks," *CVPR*, pp. 3633–3642, 2017.

[5] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global Context-Aware Attention LSTM Networks for 3D Action Recognition," in *CVPR*, 2017, pp. 3671–3680.

[6] A. Shahroudy, J. Liu, T. T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," in *CVPR*, 2016, pp. 1010–1019.

[7] J. Luo, W. Wang, and H. Qi, "Group Sparsity and Geometry Constrained Dictionary Learning for Action Recognition from Depth Maps," in *ICCV*, Dec 2013, pp. 1809–1816.

[8] X. Yang and Y. L. Tian, "EigenJoints-based Action Recognition Using Naive-Bayes-Nearest-Neighbor," in *CVPRW*, 2012, pp. 14–19.

[9] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group," in *CVPR*, 2014, pp. 588–595.

[10] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN Models for Fine-Grained Visual Recognition," in *ICCV*, 2015, pp. 1449–1457.

[11] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is Second-order Information Helpful for Large-scale Visual Recognition?" in *ICCV*, 2017, pp. 2070–2078.

[12] Q. Wang, P. Li, and L. Zhang, "G2DeNet: Global Gaussian Distribution Embedding Network and Its Application to Visual Recognition," in *CVPR*, 2017, pp. 2730–2739.

[13] R. Chakraborty, C.-H. Yang, X. Zhen, M. Banerjee, D. Archer, D. E. Vaillancourt, V. Singh, and B. C. Vemuri, "A Statistical Recurrent Model on the Manifold of Symmetric Positive Definite Matrices," in *NeurIPS*, 2018, pp. 8897–8908.

[14] Z. Huang, J. Wu, and L. V. Gool, "Building Deep Networks on Grassmann Manifolds," in *AAAI*, 2018, pp. 3279–3286.

[15] J. B. Oliva, B. Póczos, and J. Schneider, "The Statistical Recurrent Unit," in *ICML*, 2017, pp. 2671–2680.

[16] X. Nguyen, L. Brun, O. Lézoray, and S. Bougleux, "A Neural Network Based on SPD Manifold Learning for Skeleton-based Hand Gesture Recognition," in *CVPR*, 2019.

[17] Q. D. Smedt, H. Wannous, and J. Vandeborre, "Skeleton-Based Dynamic Hand Gesture Recognition," in *CVPRW*, June 2016, pp. 1206–1214.

[18] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining Actionlet Ensemble for Action Recognition with Depth Cameras," in *CVPR*, 2012, pp. 1290–1297.

[19] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal Quads: Human Action Recognition Using Joint Quadruples," in *ICPR*, 2014, pp. 4513–4518.

[20] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps, "Efficient Temporal Sequence Comparison and Classification Using Gram Matrix Embeddings on a Riemannian Manifold," in *CVPR*, 2016, pp. 4498–4507.

[21] G. Devineau, F. Moutarde, W. Xi, and J. Yang, "Deep Learning for Hand Gesture Recognition on Skeletal Data," in *IEEE International Conference on Automatic Face Gesture Recognition*, May 2018, pp. 106–113.

[22] Q. Ke, M. Bennamoun, S. An, F. A. Sohel, and F. Boussaïd, "A New Representation of Skeleton Sequences for 3D Action Recognition," in *CVPR*, 2017, pp. 4570–4579.

[23] M. Liu, H. Liu, and C. Chen, "Enhanced Skeleton Visualization for View Invariant Human Action Recognition," *Pattern Recognition*, vol. 68, pp. 346–362, 2017.

[24] J. C. Nez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vlez, "Convolutional Neural Networks and Long Short-Term Memory for Skeleton-based Human Activity and Hand Gesture Recognition," *Pattern Recognition*, vol. 76, no. C, pp. 80–94, 2018.

[25] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition," in *ECCV*, 2016, pp. 816–833.

[26] J. Weng, M. Liu, X. Jiang, and J. Yuan, "Deformable Pose Traversal Convolution for 3D Action and Gesture Recognition," in *ECCV*, 2018.

[27] Z. Huang and L. V. Gool, "A Riemannian Network for SPD Matrix Learning," in *AAAI*, 2017, pp. 2036–2042.

[28] Z. Huang, C. Wan, T. Probst, and L. V. Gool, "Deep Learning on Lie Groups for Skeleton-Based Action Recognition," in *CVPR*, 2017, pp. 6099–6108.

[29] C. Li, Z. Cui, W. Zheng, C. Xu, and J. Yang, "Spatio-Temporal Graph Convolution for Skeleton Based Action Recognition," in *AAAI*, 2018.

[30] S. Yan, Y. Xiong, and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition," in *AAAI*, 2018.

[31] C. Ionescu, O. Vantzos, and C. Sminchisescu, "Matrix Backpropagation for Deep Networks with Structured Layers," in *ICCV*, 2015, pp. 2965–2973.

[32] K. Yu and M. Salzmann, "Statistically-motivated Second-order Pooling," in *ECCV*, 2018.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, June 2016, pp. 770–778.

[34] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *CVPR*, 2017, pp. 2261–2269.

[35] C. Szegedy, , , P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *CVPR*, 2015, pp. 1–9.

[36] X. Dai, J. Y. Ng, and L. S. Davis, "FASON: First and Second Order Information Fusion Network for Texture Recognition," in *CVPR*, 2017, pp. 6100–6108.

[37] Q. Wang, Z. Gao, J. Xie, W. Zuo, and P. Li, "Global Gated Mixture of Second-order Pooling for Improving Deep Convolutional Neural Networks," in *NIPS*. Curran Associates, Inc., 2018, pp. 1277–1286.

[38] S. Cai, W. Zuo, and L. Zhang, "Higher-Order Integration of Hierarchical Convolutional Activations for Fine-Grained Visual Categorization," in *ICCV*, 2017, pp. 511–520.

[39] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie, "Kernel Pooling for Convolutional Neural Networks," in *CVPR*, 2017, pp. 3049–3058.

[40] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact Bilinear Pooling," in *CVPR*, 2016, pp. 317–326.

[41] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global Second-order Pooling Convolutional Networks," in *CVPR*, 2019.

[42] Y. Li, N. Wang, J. Liu, and X. Hou, "Factorized Bilinear Models for Image Recognition," in *ICCV*, 2017, pp. 2098–2106.

[43] Y. Wang, L. Xie, C. Liu, S. Qiao, Y. Zhang, W. Zhang, Q. Tian, and A. L. Yuille, "SORT: Second-Order Response Transform for Visual Recognition," in *ICCV*, 2017, pp. 1368–1377.

[44] G. Zoumpourlis, A. Doumanoglou, N. Vretos, and P. Daras, "Non-linear Convolution Filters for CNN-Based Learning," in *ICCV*, 2017, pp. 4771–4779.

[45] Z. Dong, S. Jia, C. Zhang, M. Pei, and Y. Wu, "Deep Manifold Learning of Symmetric Positive Definite Matrices with Application to Face Recognition," in *AAAI*, 2017, pp. 4009–4015.

[46] M. Engin, L. Wang, L. Zhou, and X. Liu, "DeepKSPD: Learning kernel-matrix-based spd representation for fine-grained image recognition," in *ECCV*, ser. LNCS, vol. 11206. Springer Int. Pub., 2018, pp. 629–645.

[47] M. Lovrić, M. Min-Oo, and E. A. Ruh, "Multivariate Normal Distributions Parametrized As a Riemannian Symmetric Space," *Journal of Multivariate Analysis*, vol. 74, no. 1, pp. 36–48, 2000.

[48] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric Means in a Novel Vector Space Structure on Symmetric Positive Definite Matrices," *SIAM journal on Matrix Analysis Applications*, vol. 29, no. 1, pp. 328–347, 2007.

[49] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian Detection via Classification on Riemannian Manifolds," *TPAMI*, vol. 30, no. 10, pp. 1713–1727, 2008.

[50] Q. D. Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. L. Saux, and D. Filliat, "3D Hand Gesture Recognition Using a Depth and Skeletal Dataset," in *Eurographics Workshop on 3D Object Retrieval*, 2017, pp. 33–38.

[51] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations," in *CVPR*, 2018.

[52] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[53] O. Oreifej and Z. Liu, "HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences," in *CVPR*, June 2013, pp. 716–723.

[54] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. D. Bimbo, "3-D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold," *IEEE Transactions on Cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2015.

[55] H. Rahmani and A. Mian, "3D Action Recognition from Novel Viewpoints," in *CVPR*, June 2016, pp. 1506–1515.

[56] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks," in *AAAI*, 2016, pp. 3697–3703.

[57] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection," in *ICCV*, 2013, pp. 2752–2759.

[58] G. Garcia-Hernando and T.-K. Kim, "Transition Forests: Learning Discriminative Temporal Transitions for Action Recognition," in *CVPR*, 2017, pp. 407–415.

[59] J. Hu, W. Zheng, J. Lai, and J. Zhang, "Jointly Learning Heterogeneous Features for RGB-D Activity Recognition," in *CVPR*, 2015, pp. 5344–5352.