Efficient Multi-stream Temporal Learning and Post-fusion Strategy for 3D Skeleton-based Hand Activity Recognition

Yasser Boutaleb[1,2], Catherine Soladie[2], Nam-Duong Duong[1], Amine Kacete[1], Jérôme Royan[1] and Renaud Seguier[2]

[1]*IRT b-com, 1219 Avenue des Champs Blancs, 35510 Cesson-Sevigné, France*

[2]*IETR/CentraleSupelec, Avenue de la Boulaie, 35510 Cesson-Sevigné, France*

Keywords: First-person Hand Activity, Multi-stream Learning, 3D Hand Skeleton, Hand-crafted Features, Temporal Learning.

Abstract: Recognizing first-person hand activity is a challenging task, especially when not enough data are available. In this paper, we tackle this challenge by proposing a new hybrid learning pipeline for skeleton-based hand activity recognition, which is composed of three blocks. First, for a given sequence of hand's joint positions, the spatial features are extracted using a dedicated combination of local and global spacial hand-crafted features. Then, the temporal dependencies are learned using a multi-stream learning strategy. Finally, a hand activity sequence classifier is learned, via our Post-fusion strategy, applied to the previously learned temporal dependencies. The experiments, evaluated on two real-world data sets, show that our approach performs better than the state-of-the-art approaches. For more ablation study, we compared our Post-fusion strategy with three traditional fusion baselines and showed an improvement above 2.4% of accuracy.

1 INTRODUCTION

Understanding human activity from a first-person (egocentric) perspective by focusing on the hands is getting interest of both the computer vision and the robotics communities. Indeed, the range of potential applications includes, among others, Human Computer Interaction (Sridhar et al., 2015), Humanoid Robotics (Ramirez-Amaro et al., 2017), and Virtual/Augmented Realty (Surie et al., 2007). In particular, with the development of effective and low cost depth camera sensors over the last years (e.g., Microsoft Kinect or Intel RealSense), 3D skeletal data acquisition becomes possible with a sufficient accuracy (Yuan et al., 2018; Moon et al., 2018). This has promoted the problem of human activity recognition.

The 3D Skeletal data provides a robust high-level description regarding common problems in RGB imaging, such as background subtraction and light variation. To this end, many skeleton-based approaches have been proposed. Most of them are based on end-to-end Deep Learning (DL) (Du et al., 2015; Wang and Wang, 2017) which have been proven to be effective when a large amount of data is available. Hence, for some industrial applications, providing large-scale labeled data sets is still hard and expansive to achieve due to the manual data annotation pro-

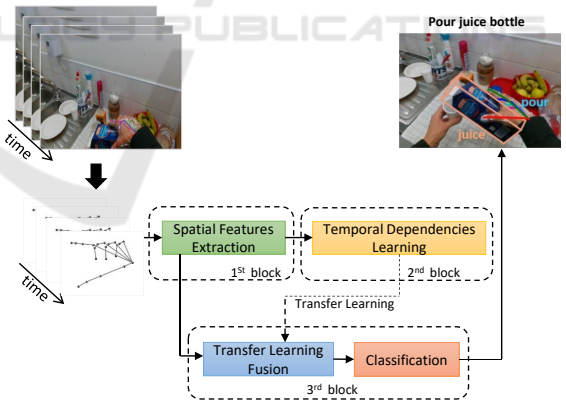

Figure 1: Our proposed learning pipeline for 3D skeleton-based first-person hand activity recognition. For a given 3D hand skeleton activity sequence, in the first block, the spacial features are extracted using existent and new hand-crafted methods. Then, in the second block, the temporal dependencies are learned. In the last block, a hand activity sequence classifier is learned, using a Post-fusion strategy, which is applied to the previously learned temporal dependencies. For the test predictions, only the first and the last blocks are involved.

Boutaleb, Y., Soladie, C., Duong, N., Kacete, A., Royan, J. and Seguier, R.
Efficient Multi-stream Temporal Learning and Post-fusion Strategy for 3D Skeleton-based Hand Activity Recognition.
DOI: 10.5220/0010232702930302
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 4: VISAPP*, pages 293-302
ISBN: 978-989-758-488-6

cess. On the other hand, pure hand-crafted (HC) features based approaches (Smedt et al., 2016; Devanne et al., 2015; Kacem et al., 2017; Zhang et al., 2016; Hussein et al., 2013) can deal with limited amount of data. Yet, they are still struggling to learn temporal dependencies along the sequence time-steps. As a tuning alternative between performance and data acquisition cost, hybrid methods combines DL and pure HC methods (Avola et al., 2019; Chen et al., 2017; Liu et al., 2019; Zhang et al., 2019).

Motivated by all these observations, we introduce in this paper, a new hybrid approach for 3D skeleton-based first-person hand activity recognition. We highlight the contributions as follows:

- A novel hybrid learning pipeline for first-person hand activity recognition that consists of three sequential blocks as illustrated in Figure 1: In the first block, we extract the spatial features using our proposed selection of existent and new HC features extraction methods. Then, in the second block, we differ from the existing methods by learning the temporal dependencies independently on each HC features, using a simplified separated Neural Network (NN) to avoid the overfitting problem. Finally, we exploit the knowledge from the previous block to classify activities using a tuning strategy that we called a Post-fusion. Once the learning is completed, for the predictions, only the first and the last blocks are involved. This multi-steps learning pipeline allows training on a limited number of samples while ensuring good accuracy.
- A combination of three local and global HC features extraction methods for 3D skeleton-based first-person hand activity recognition, that we summarize as follows. (1) Inspired by (Smedt et al., 2016), we use a Shape of Connected Joints (SoCJ), which characterizes the activity sequence by the variation of the physical hand shape at each time-step. (2) Our proposed Intra/Inter Finger Relative Distances (IIFRD) which also relevantly characterizes the activity sequence at each time-step by the variation of the Inter-fingers relative distances between the physically adjacent fingers pairs, and the Intra-Finger Relative Distances which belong to the distance between two opposite joints of a pairs of directly connected segments of the finger. (3) Since the SoCJ and IIFRD only focus on the local features of the hand at each time-step, we proposed a complimentary HC method called Global Relative Translations (GRT), that focuses on the global features of the entire sequence, by exploiting the displacement of the hand during the activity.

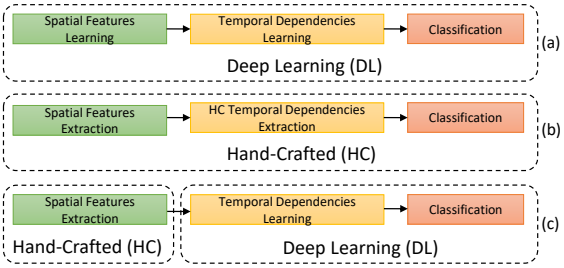

Figure 2: A categorization of 3D Skeleton-based activity recognition methods. (a) Deep Learning based methods (DL). (b) Hand-Crafted methods (HC). (c) Hybrid methods that combine both DL and HC methods.

The remainder of this paper is organised as follows. After giving a review on the related work in Section 2, we describe our proposed approach for 3D skeleton-based hand activity recognition in Section 3. Then, we show the benefit of the proposed approach by presenting and discussing the experimental results in Section 4. Section 5 concludes the paper.

2 RELATED WORK

This section presents related work for 3D skeleton-based first-person hand activity recognition. We also introduce other similar 3D skeleton-based approaches, namely, dynamic hand gesture recognition and human activity recognition. We categorize these approaches as DL based (Sec 2.1), HC based (Sec 2.2), and Hybrid methods (Sec 2.3) as illustrated in Figure 2.

2.1 Deep Learning based Methods

Recently, end-to-end DL approaches (Figure 2. (a)) showed great success in different applications, including skeleton-based activity recognition (Wang et al., 2019). The great performance of Convolutional Neural Networks (CNNs) has motivated (Caetano et al., 2019a; Caetano et al., 2019b) to formulate the recognition problem as an image classification problem by representing a sequence of 3D skeleton joints as a 2D image input for a deep CNN. Also, attracted by the success of CNNs, (Li et al., 2017) proposed a two-stream feature convolutional learning to manage both spatial and temporal domains. On the other hand, other recent works (Yan et al., 2018; Tang et al., 2018; Shi et al., 2018; Li et al., 2019; Si et al., 2019) focused on the Graph Convolutional Networks (GCNs) to exploit the connections between the skeleton joints formed by the physical structure. Furthermore, to better exploit information in the temporal dimension,

many other works focused on Recurrent Neural Networks (RNNs) equipped with Long Short Term Memory (LSTMs) cells (Du et al., 2015; Liu et al., 2016) or Gated Recurrent Units (GRUs) (Maghoumi and LaViola, 2018) for their capabilities of reasoning along the temporal dimension to learn the temporal dependencies. More recently, (Nguyen et al., 2019) proposed a new NN layer that uses Symmetric Positive Definite (SPD) matrix as low-dimensional discriminative descriptor for classification, which has been proven to be effective. However, most of these methods still have difficulties to extract relevant spatial features due to the limited number of training samples and the sparseness of the 3D skeleton data.

2.2 Hand-crafted based Methods

In spite of the success of DL approaches in the last few years, classical methods that mainly use HC techniques to perform recognition (Figure 2. (b)) still get attention. Most of these approaches focus on representing the data in a non-Euclidean domain. A reference work was proposed by (Vemulapalli et al., 2014) where they represented the 3D skeleton data as points on a Lie group. Also, in (Devanne et al., 2015), a Riemannian manifold is used as a non-Euclidean domain to formulate the recognition problem as a problem of computing the similarity between the shape of trajectories. Similarly and besides, (Zhang et al., 2016; Kacem et al., 2017), exploited the Gram Matrix to handle the temporal dimension and distances measurement for the classification.

On the other hand, more closer to our HC features extraction methods, some recent work focused on the exploitation of the 3D geometrical information. (Smedt et al., 2016) proposed a set of HC geometrical features based on the connection between the hand joints and rotations, that they represented as histograms input vectors for a SVM to classify hand gestures. Similarly to (Evangelidis et al., 2014; Zhang et al., 2013), (Smedt et al., 2016) used a Temporal Pyramid representation to manage the temporal dimension. In (Ohn-Bar and Trivedi, 2013), joint angles similarities and a Histogram of Oriented Gradients (HOG) are used as input into a SVM to classify activities. This category of methods has been proven to be very effective in providing relevant spatial features, but most of them are still struggling to learn long-term temporal dependencies.

2.3 Hybrid Methods

This category (Figure 2. (c)) combines the two previously introduced approaches (Sec 2.1 and Sec 2.2) to overcome their limitations. Seeking to exploit their advantages, our proposed approach falls in this category since DL methods have been proven powerful in learning temporal dependencies, while HC features based methods gives very discriminating spatial features. (Avola et al., 2019) concatenated a set of HC features as an input into a deep LSTMs to classify American Sign Language (ASL) and semaphoric hand gestures. Similarly, aiming at classifying Human 3D Gaits, (Liu et al., 2019) concatenated relative distances and angles as a feature input vector into an LSTM to manage the temporal dimension, while in parallel a CNN is exploited to learn spatial features from 2D Gait Energy Images. Yet, the early fusion of different features spaces increases the input complexity and the learning noise (Ying, 2019), especially when only few training data are provided. To prevent this problem, (Chen et al., 2017), proposed an end-to-end slow fusion based architecture. But this type of complex architecture requires a lot of data amount. More recently (Zhang et al., 2019) proposed a Quaternion Product Unit (QPU), that describes 3D skeleton with rotation-invariant and rotation-equivariant features, which can be fed into a Deep Neural Networks (DNN) to recognize activities. Yet, results given by their experiments showed a low score of accuracy.

3 PROPOSED METHOD

This section details our proposed hybrid approach following the illustration of Figure 1. In the first block, we extract the HC features (Sec 3.1). Then, in the second block, we learn the temporal dependencies (Sec 3.2). Once the temporal learning is ended, in the last block we transfer and exploit the knowledge from the previous block to learn classifying activities (Sec 3.3).

3.1 Hand-crafted Features Extraction

A 3D hand skeleton activity sequence is the only input, that we denote $S(t)$. At each time-step t, the hand is represented by a configuration of physically connected n joints $\{J_j^t = (x_j^t, y_j^t, z_j^t)\}_{j=1:n}$. Each joint is represented by 3D Cartesian coordinates forming a set of segments that yields to the hand bones, the phalanges and metacarpals (Figure 3). We define and formulate the activity sequence as follows:

$$S(t) = \{\{J_j^t\}_{j=1:n}\}_{t=0:T} \quad (1)$$

where T is the max length of the sequence.

In order to exploit the 3D geometrical information, in the first block, we use three HC features extraction methods, that provide relevant features for

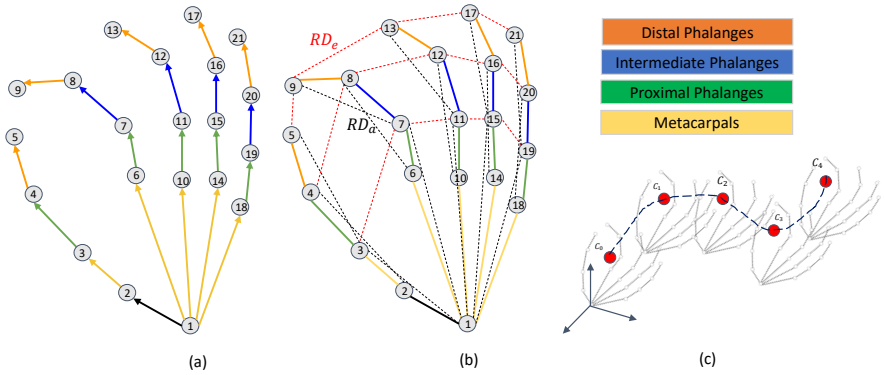

Figure 3: The proposed selection of hand-crafted features. (a) Shape of Connected Joints (SoCJ), the hand shape is represented by 3D vectors between physically connected joints (Smedt et al., 2016). (b) Our proposed Intra/Inter Finger Relative Distances (IIFRD) which characterize the activities by the high variation of intra-finger distances RD_a (in black) and inter-finger distances RD_e (in red). (c) Our proposed Global Relative Translation which aims to characterize the activity sequence by the translation of the joints centroid at each time-step.

first-person hand activity recognition. Note that before proceeding to feature extraction, the 3D hand skeleton data are normalized, such as all the hands of all the subjects are adjusted to the same average size while keeping the angles intact. In the following descriptions, we model the hand skeleton by fixing the number of joints to $n = 21$.

Shape of Connected Joints (SoCJ). Inspired by (Smedt et al., 2016), we use the SoCJ to represent the variation of the hand shape during the activtiy. For each finger, we compute the 3D vectors between physically connected joints, from the wrist up to the fingertips (Figure 3 (a)). Let $F_1 = \{J_j\}_{j=1:5}$ be a set of joints which are ordered such as they represent the physical connections of the thumb finger and wrist joint J_1 as shown in Figure 3 (a), the $SoCJ(F_1)$ can be computed as follows:

$$SoCJ(F_1) = \{J_j - J_{j-1}\}_{j=5:2} \quad (2)$$

By applying the SoCJ to all the fingers, as a result, for each time-step t, we obtain a feature descriptor $\{SoCJ(F_l^t)\}_{l=1:5} \in \mathbb{R}^{4\times5\times3}$, where F_l is the l-th finger. $\Psi_1(.)$ denote the SoCJ method applied to the entire activity sequence $S(t)$, that we define as follows:

$$\Psi_1(S(t)) = \{\{SoCJ(F_l^t)\}_{l=1:5}\}_{t=0:T} \quad (3)$$

Intra/Inter Finger Relative Distances (IIFRD). We exploit the periodic variation of the intra-finger and inter-fingers relative distances, which relevantly characterizes the activity sequence (Figure 3 (b)).

- The intra-finger relative distances, that we denote RD_a, gives strong internal dependencies between finger's connected segments. It represents the distance between two opposite joints of a pairs of directly connected segments from each fingertip down to the wrist (Figure 3 (b) in black). Lets take F_1 as described previously. The $RD_a(F_1)$ can be computed as follows:

$$RD_a(F_1) = \{d(J_j, J_{j-2})\}_{j=5:3} \quad (4)$$

where d is the Euclidean distance. By applying the RD_a to all fingers, for each time-step t, we get a feature descriptor $a(t) = \{RD_a(F_l^t)\}_{l=1:5} \in \mathbb{R}^{15}$.

- The inter-finger relative distances, that we denote by RD_e (Figure 3 (b) in red), gives external dependencies between adjacent fingers pairs. For instance, lets take F_1 as described previously and $F_2 = \{J_j\}_{j=7:9}$, two sets of connected joints, that refers to the thumb and the index fingers respectively. The $RD_e(F_1, F_2)$ is computed as follows:

$$RD_e(F_1, F_2) = \{d(J_j, J_{j+4})\}_{j=3:5} \quad (5)$$

By applying the RD_e to the four pairs of adjacent fingers, for each time-step t, we obtain a feature descriptor $e(t) = \{RD_e(F_l^t, F_{l+1}^t)\}_{l=1:4} \in \mathbb{R}^{12}$.

Finally, by concatenating the two descriptors $a(t)$ and $e(t)$, for each time-step t, we obtain a final feature descriptor $\{a(t), e(t)\} \in \mathbb{R}^{15+12}$. We denote by $\Psi_2(.)$ the IIFRD method applied to the entire activity sequence $S(t)$, that we define as follows:

$$\Psi_2(S(t)) = \{a(t), e(t)\}_{t=0:T} \quad (6)$$

Global Relative Translations (GRT). Unlike the IIFRD and SoCJ descriptors, which only consider the local features that belong to the fingers motion at each time-step, the GRT characterize the activity sequence by computing the relative displacement of all the hand joints along the sequence time-steps (Figure 3 (c)). To this end, for each sequence, we fix the wrist joint J_1^0 of the first time-step $t = 0$ as the origin. Then, we

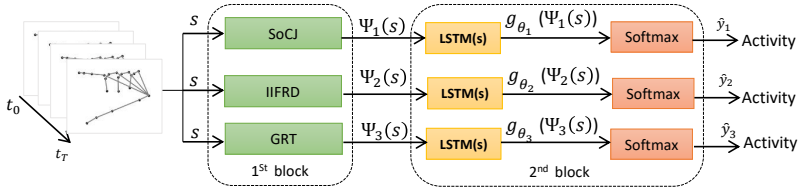

Figure 4: Illustration of the first and second blocks of the proposed learning pipeline. For each hand-crafted feature descriptor (SoCJ, RD, and GRT) seen in Figure 3, a Neural Network composed of staked LSTM layers and a softmax layer are trained independently to learn temporal dependencies.

transform all remaining joints of the sequence to this new coordinate system as follows:

$$\hat{J}_j^t = J_j^t - J_1^0 \tag{7}$$

where the $\hat{J}_j^t$ the new transformed j-th joint at the time-step t. Once the transformation is done, at each time-step, we compute the centroid of the transformed joints $C_t = \frac{1}{21}\sum_{j=1}^{21}\hat{J}_j^t$. We denote by $\Psi_3(.)$ the application of GRT to the entire sequence $S(t)$, that we define as follows:

$$\Psi_3((S(t)) = \{C_t\}_{t=0:T} \tag{8}$$

The GRT gives discriminate complementary information to the IIFRD and the SoCJ by considering the global trajectory of the hand along the activity. In Section 4.3 we quantitatively show the benefit of these complementary information.

3.2 Temporal Dependencies Learning

Learning long and complex activities requires considering the temporal dimension to make use of the long-term dependencies between sequence time-steps. To this end, we use LSTMs cells for its great success and capabilities to learn these long/short term dependencies. Moreover, in contrast to traditional RNNs, LSTMs overcome the vanishing gradient problem by using a specific circuit of gates (Hochreiter and Schmidhuber, 1997).

(Avola et al., 2019; Liu et al., 2019) concatenate different types of features spaces as one input vector, which may complicate the input and confuse the NN. In contrast, for each HC features descriptor (seen in Sec 3.1), we train separately a simple NN that consists of staked LSTM layers followed by a softmax layer to classify activities. Therefore, in total, we train three NN separately as shown in Figure 4.

More formally, lets $\{\Psi_k(S)\}_{k=1:3}$ be the set of the three feature descriptors corresponding to Eq.3, Eq.6 and Eq.8 defined in Section 3.1, where S is the activity sequence input. For each feature descriptor $\Psi_k(S)$, we model the temporal dependencies with a composite function $g_{\theta_k}(\Psi_k(S))$, where $g_{\theta_k}(\cdot)$ is the k-th LSTM sub-network with θ the learnable parameters, while the output of $g_{\theta_k}(\cdot)$ refers to the last hidden state of the last LSTM unit. For each network we define a cross entropy loss function $\mathcal{L}_k$ as follows:

$$\mathcal{L}_k = -\sum_{c=1}^{N} y^c log(\hat{y}_k^c) \tag{9}$$

where N is the number of classes, y^c the target label and $\hat{y}_k^c$ the softmax output that refers to the predicted label. The temporal learning parameters are optimized by minimizing over a labeled data set:

$$\theta_k^* = \arg\min_{\theta_k} \mathcal{L}_k(y, \hat{y}_k) \tag{10}$$

At the end of the training, as a result, we have a set of three trained LSTM sub-networks, with θ_k^* an optimised parameters:

$$\{g_{\theta_k^*}(\Psi_k(S))\}_{k=1:3} \tag{11}$$

We note that the purpose of this second block is to learn the temporal dependencies, and all the classification results $\hat{y}_k$ are ignored. Only the results shown in Eq.11 are needed for the next block.

This pre-training strategy of multiple networks avoid the fusion of different features spaces, which reduces the input complexity and the noise learning. It also allows the LSTM to focus only on learning over one specific feature input independently, which also helps to avoid the over-fitting problem (Ying, 2019).

3.3 Post-fusion Strategy and Classification

Once the temporal dependencies are learned in the second block (Sec 3.2), we proceed to the final classification. To this end, we train another multi-input NN that exploits the resulted three pre-trained LSTM layers introduced in (Sec 3.2) that we transfer with a fixed optimized parameters θ^* as illustrated in Figure 5.

Seeking to ensure the best classification accuracy, the three parallel outputs branches of the transferred LSTMs are concatenated, then fed into a Multi Layers Perception (MLP) that consists of two Fully Connected (FC) layers, followed by a softmax layer (Figure 5). We model this network as shown in Eq.12,

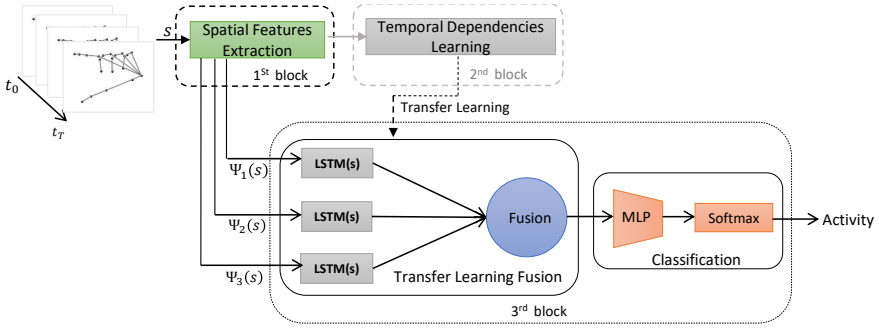

Figure 5: Illustration of the third block of the proposed pipeline. Once the temporal dependencies are learned in the second block. The LSTM layers are transferred to the third block with fixed parameters. Their outputs are fused and fed into a MLP followed by a softmax layer for the final classification.

where f_ϕ is a MLP+softmax with learnable parameters ϕ, and h is the concatenation function:

$$f_\phi(h(\{g_{\theta_k^*}(\Psi_k(S))\}_{k=1:3})) \quad (12)$$

The learnable parameters ϕ are optimized using the same loss function as in the previous block (Sec 3.2) by minimizing over the same training data set. Note that for the test predictions, only this network is involved using the three HC features descriptors introduced in (Sec 3.1) as a multi-input.

The proposed Post-fusion strategy aims at ensuring a good accuracy score through tuning between the pre-trained networks outputs. In Section 4.4, we quantitatively show the efficiency of this strategy compared to other traditional fusing and classification methods.

4 EXPERIMENTS

4.1 Data Sets

FPHA Data Set. Proposed by (Garcia-Hernando et al., 2018). It's the only publicly available data set for first-person hand activity recognition. This data set provides RGB and depth images with the 3D annotations of the 21 hand joints, the 6 Dof object poses, and the activity classes. It is a diverse data set that includes 1175 activity videos belonging to 45 different activity categories, in 3 different scenarios performed by 6 actors with high inter-subject and intra-subject variability of style, speed, scale, and viewpoint. It represents a real challenge for activity recognition algorithms. We note that, for the proposed method, we only need the 3D coordinates of the hand joints. For all the experiments, we used the setting proposed in (Garcia-Hernando et al., 2018), with exactly the same distribution of data: 600 activity sequences for training and 575 for testing.

Dynamic Hand Gesture (DHG) 14/28 Data Set. Proposed by (Smedt et al., 2016), which is basically devoted to a related domain, namely the hand gesture recognition. We use it in order to better validate our proposed approach. The data set contains 14 gestures performed in two ways: using one finger and the whole hand. Each gesture is performed 5 times by 20 participants in 2 ways, resulting in 2800 sequences. Sequences are labelled following their gesture, the number of fingers used, the performer and the trial. Each frame contains a depth image, the coordinates of 22 joints both in the 2D depth image space and in the 3D world space forming a full hand skeleton. Note that for our experiment we only need the 3D hand skeleton joints. We ignored the palm center and we only considered the remaining 21 hand joints.

4.2 Implementation Details

The Learning of Temporal Dependencies. For every extracted HC features, we trained different configurations of separated NNs that consists of 1,2,3 and 4 staked LSTM layers followed by a softmax. We selected the best configuration that gives the best accuracy score: only one LSTM layer of 100 units for the FPHA data set, and two staked LSTMs of 200 units for the DHG 14/28 data set. We set the probability of dropout to 0.5 (outside and inside the LSTM gates). We use Adam with a learning rate of 0.001 for the optimization. All the networks are trained with a batch size of 128 for 2000 to 3000 epochs. We also, padded all the sequence lengths to 300 time-steps per sequence.

Post-fusion and Classification. Once all the temporal dependencies are learned (end of block 2), in the Post-fusion step, we recover the pre-trained LSTM networks, we fix all their weights, and we discard the softmax layers. Then, the three outputs branches from the three parallel transferred LSTMs are concatenated and followed by a MLP that consists of two dense

layers of 256 and 128 neurons respectively, equipped with a relu activation function. At the end of the network, a softmax layer is used for the final classification. This network is trained until 100 epochs, with the same batch size and optimization parameters as the previous networks. Our implementations are based on the Keras framework.

4.3 Hand-crafted Features Analysis

Table 1: Test accuracy results on the FPHA data set. The selection hand-crafted features independently, and the combinations using our proposed approach.

Hand-crafted Feature	Acc.(%)
Shape of Connected Joints (SoCJ)	89.91
Intera/Inter Finger Relative Distance (IIFRD)	88.17
Global Relative Translations (GRT)	58.26
IIFRD + GRT	93.73
SoCJ + GRT	92.17
SoCJ + IIFRD	93.91
SoCJ + IIFRD + GRT	**96.17**

In order to analyse the effectiveness of the selected HC features, we evaluated each one independently using a simplified end-to-end NN architecture composed of one LSTM layer of 100 units with a dropout of 0.5 (outside and inside the LSTM gates) and a softmax layer. We also evaluated possible HC features combinations by using our approach with the same configuration introduced in (Sec 4.2). The results in Table 1. show that the SoCJ and IIFRD, alone, are capable of achieving a good accuracy of 89.91% and 88.17% respectively. As expected, the GRT alone is unable to classify activities by achieving only 58.26% of accuracy. But, it boosts the performance if combined with the SoCJ, the IIFRD, or both, by achieving the best accuracy of 96.17%. This can be explained by the fact that the SoCJ, and the IIFRD focus on the local features based on the motion of the fingers, ignoring translations between the activity sequence time-steps, while the GRT focuses on the global feature based on the displacement of the hand during the activity, which provides an important complimentary information. The combination of the three selected HC features allowed us to overcome the commonly confused classes "open wallet" and "use calculator" even if the hand poses are dissimilar but more subtle (Garcia-Hernando et al., 2018). Nevertheless, we still get confusion between "open wallet" and "flip sponge" classes, due to the limited displacement of the hand, the shortness of the activities, and the limited number of samples in the data set compared to the other classes (Garcia-Hernando et al., 2018). Please refer to the appendix for more details.

4.4 Post-fusion Strategy and Classification Analysis

We compared our Post-fusion strategy with three traditional baselines, the early, the slow, and the late fusion, that we define as follows:

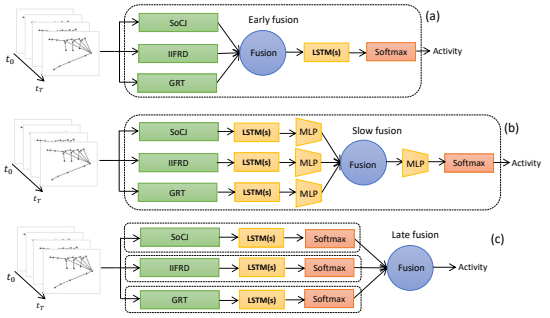

Figure 6: Hybrid fusion and classification baselines, (a) **Early fusion**, is an end-to-end architecture, where the extracted hand-crafted features are concatenated and fed into a temporal learning network followed by a classifier. (b) **Slow fusion**, is an end-to-end architecture, where, for each extracted hand-crafted feature, the temporal dependencies are learned separately, then concatenated and fed into a classifier. **(c) Late fusion**, is a multi-stream learning, where for each extracted features, an end-to-end temporal network is trained separately, and at the end, a majority vote is applied to the their classifier outputs.

Early Fusion. (Figure 6. (a)) As in (Avola et al., 2019; Liu et al., 2019), we concatenate our extracted HC features descriptors in one unified vector that we fed into deep staked LSTM layers of 200 units followed by a softmax layer. We evaluated the baseline with a configuration of 2, 3 and 4 staked layers, then, the best accuracy results are selected for the comparison.

Slow Fusion. (Figure 6. (b)) As in (Chen et al., 2017), for each extracted HC features descriptor, we used 2 stacked LSTM layers of 200 units, followed by a Fully Connected layer (FC) of 128 neurons. The outputs from the three parallel FC branches are concatenated and followed by 2 sequential FC layers of 256 and 128 neurons respectively. At the end of the network, a softmax layer is used for the classification. All the layers are followed by a dropout layer, and all the FC layers are equipped with a relu activation function.

Late Fusion. (Figure 6. (c)) In contrast to the previously introduced end-to-end baselines, in this architecture, for each HC features a NN composed of a LSTM layer of 100 units and softmax layer is trained separately. At the end of training, a majority vote is applied by adding the softmax outputs scores.

Table 2: Accuracy results on 50%, and 100% of the 600 FPHA data set training samples. For the test, all the 575 testing samples are kept. We compared our proposed approach with three traditional fusion and classification baselines.

Architecture	300 samples **Acc.(%)**	600 samples **Acc.(%)**
Early fusion	75.65	90.95
Slow fusion	63.47	86.43
Late fusion	76.26	93.73
Our	**79.78**	**96.17**

We trained our network architecture and the selected baselines with 100%, then only 50% of the 600 training samples of the FPHA data set that belongs to the subject 1, 3, and 6. For the test, we kept all the 575 testing samples.

Our proposed approach outperforms the baselines with more then 3.52% using 50% and 2.44% of accuracy using 100% of 600 training samples respectively, which confirms the effectiveness of our fusion strategy. Moreover, by using only 300 samples (a half) of the FPHA data set training samples, our approach is achieving the state-of-the-art performance. Thanks to its simplified architecture, the early fusion outperforms the slow fusion, which is more complex and implies the over-fitting problem. The late fusion outperforms both, thanks to its simplified NNs trained independently, which helps to overcome the over-fitting problem. The late fusion preforms well, but its naive fusion can not ensure good tuning between the NNs outputs. Please refer to the appendix for more comparison.

Table 3: Test accuracy comparison of our proposed approach and the state-of-the-art approaches on the FPHA data set. The bests results are marked in bold.

Method	**Color**	**Depth**	**Pose**	**Acc.(%)**
(Feichtenhofer et al., 2016)	✓	✗	✗	61.56
(Feichtenhofer et al., 2016)	✓	✗	✗	69.91
(Feichtenhofer et al., 2016)	✓	✗	✗	75.30
(Ohn-Bar and Trivedi, 2014)	✗	✓	✗	59.83
(Ohn-Bar and Trivedi, 2014)	✗	✓	✓	66.78
(Oreifej and Liu, 2013)	✗	✓	✗	70.61
(Rahmani and Mian, 2016)	✗	✓	✗	69.21
(Garcia-Hernando et al., 2018)	✗	✗	✓	78.73
(Garcia-Hernando et al., 2018)	✗	✗	✓	80.14
(Zanfir et al., 2013)	✗	✗	✓	56.34
(Vemulapalli et al., 2014)	✗	✗	✓	82.69
(Du et al., 2015)	✗	✗	✓	77.40
(Zhang et al., 2016)	✗	✗	✓	85.39
(Garcia-Hernando et al., 2018)	✗	✗	✓	80.69
(fang Hu et al., 2015)	✓	✗	✗	66.78
(fang Hu et al., 2015)	✗	✓	✗	60.17
(fang Hu et al., 2015)	✗	✗	✓	74.60
(fang Hu et al., 2015)	✓	✓	✓	78.78
(Huang and Gool, 2016)	✗	✗	✓	84.35
(Huang et al., 2016)	✗	✗	✓	77.57
(Tekin et al., 2019)	✓	✗	✗	82.26
(Zhang et al., 2019)	✗	✗	✓	82.26
(Lohit et al., 2019)	✗	✗	✓	82.75
(Nguyen et al., 2019)	✗	✗	✓	**93.22**
(Rastgoo et al., 2020)	✓	✗	✓	**91.12**
Our	✗	✗	✓	**96.17**

4.5 State-of-the-Art Comparison

Table 3 shows the accuracy of our approach compared with the state-of-the-art approaches on the FPHA data set. We note that the accuracy results of (Feichtenhofer et al., 2016; Ohn-Bar and Trivedi, 2014; Oreifej and Liu, 2013; Rahmani and Mian, 2016; Zanfir et al., 2013; Vemulapalli et al., 2014; Du et al., 2015; Zhang et al., 2016; fang Hu et al., 2015) and (Huang and Gool, 2016; Huang et al., 2016) are reported by (Garcia-Hernando et al., 2018) and (Nguyen et al., 2019) respectively, where the recognition may need the full body joints instead of hands and some of them might not be tailored for hand activities.

The best performing approaches among state-of-the-art methods are the NN based on SPD manifold learning (Nguyen et al., 2019), and the multi-modal approach proposed by Razieh et al (Rastgoo et al., 2020), which gives 93.22% and 91.12% of accuracy respectively, 3.26% inferior to our proposed approach. The remaining methods are outperformed by our approach by more than 11% of accuracy.

Table 4: Accuracy comparison of our proposed approach and the state-of-the-art approaches on DHG-14/28 data set. The bests results are marked in bold.

Method	**Color**	**Depth**	**Pose**	**Accuracy (%)**	
				14 gest	**18 gest**
(Oreifej and Liu, 2013)	✗	✓	✗	78.53	74.03
(Devanne et al., 2015)	✗	✗	✓	79.61	62.00
(Huang and Gool, 2016)	✗	✗	✓	75.24	69.64
(Ohn-Bar and Trivedi, 2014)	✗	✗	✓	83.85	76.53
(Chen et al., 2017)	✗	✗	✓	84.68	80.32
(Smedt et al., 2016)	✗	✗	✓	88.24	81.90
(Devineau et al., 2018)	✗	✗	✓	91.28	84.35
(Nguyen et al., 2019)	✗	✗	✓	94.29	89.40
(Maghoumi and LaViola, 2018)	✗	✗	✓	94.50	**91.40**
(Avola et al., 2019)	✗	✗	✓	**97.62**	**91.43**
Our	✗	✗	✓	**95.21**	90.10

Table 4 shows that our proposed approach is achieving the state-of-art results on the DHG-14/28 data set, even that our selected HC features methods are adapted to the first-person hand activity recognition and not to the hand gesture recognition problem. The approach proposed by (Avola et al., 2019) outperforms all the state-of-the-art approaches including ours, thanks to theirs proposed HC features which are well adapted to American Signe Language (ASL)

and semaphoric hand gestures. Furthermore, the time sampling strategy used in (Avola et al., 2019) allows to better classify the least dynamic and shortest gestures, unlike our proposed approach which is adapted to deal with the hand activities where the hand is supposed to be more dynamic.

5 CONCLUSIONS

In this paper, a novel learning pipeline for first-person hand activity recognition is presented. The proposed pipeline is composed of three blocks. The first block is a new combination of HC features extraction methods. The second block is our multi-stream temporal dependencies learning strategy. In the last block, we introduced our proposed Post-fusion strategy, which has been proven to be more efficient than other existing traditional fusion methods. The proposed approach is evaluated on two real world data set and showed a good accuracy results.

As future improvements, we plan to exploit the color and object pose information in addition to the skeletal data, in order to avoid the ambiguous case where the manipulated objects in different activities may have the same dimension but with different colors.

REFERENCES

Avola, D., Bernardi, M., Cinque, L., Foresti, G. L., and Massaroni, C. (2019). Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. *IEEE Transactions on Multimedia*, 21:234–245.

Caetano, C., Brémond, F., and Schwartz, W. R. (2019a). Skeleton image representation for 3d action recognition based on tree structure and reference joints. *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 16–23.

Caetano, C., de Souza, J. S., Brémond, F., dos Santos, J. A., and Schwartz, W. R. (2019b). Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8.

Chen, X., Guo, H., Wang, G., and Zhang, L. (2017). Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2881–2885.

Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., and Bimbo, A. D. (2015). 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE Transactions on Cybernetics*, 45:1340–1352.

Devineau, G., Moutarde, F., Xi, W., and Yang, J. (2018). Deep learning for hand gesture recognition on skeletal data. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 106–113.

Du, Y., Wang, W., and Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118.

Evangelidis, G. D., Singh, G., and Horaud, R. (2014). Skeletal quads: Human action recognition using joint quadruples. *2014 22nd International Conference on Pattern Recognition*, pages 4513–4518.

fang Hu, J., Zheng, W.-S., Lai, J.-H., and Zhang, J. (2015). Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*.

Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941.

Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. (2018). First-person hand action benchmark with rgb-d videos and 3d hand pose annotations.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.

Huang, Z. and Gool, L. V. (2016). A riemannian network for spd matrix learning. *ArXiv*, abs/1608.04233.

Huang, Z., Wu, J., and Gool, L. V. (2016). Building deep networks on grassmann manifolds. In *AAAI*.

Hussein, M. E., Torki, M., Gowayyed, M. A., and El-Saban, M. (2013). Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*.

Kacem, A., Daoudi, M., Amor, B. B., and Paiva, J. C. Á. (2017). A novel space-time representation on the positive semidefinite cone for facial expression recognition. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3199–3208.

Li, C., Zhong, Q., Xie, D., and Pu, S. (2017). Skeleton-based action recognition with convolutional neural networks. *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 597–600.

Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2019). Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*.

Liu, J., Shahroudy, A., Xu, D., and Wang, G. (2016). Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*.

Liu, Y., Jiang, X., Sun, T., and Xu, K. (2019). 3d gait recognition based on a cnn-lstm network with the fusion of skegei and da features. *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8.

Lohit, S., Wang, Q., and Turaga, P. K. (2019). Temporal transformer networks: Joint learning of invariant and

discriminative time warping. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12418–12427.

Maghoumi, M. and LaViola, J. J. (2018). Deepgru: Deep gesture recognition utility. In *ISVC*.

Moon, G., Chang, J., and Lee, K. M. (2018). V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nguyen, X. S., Brun, L., Lézoray, O., and Bougleux, S. (2019). A neural network based on spd manifold learning for skeleton-based hand gesture recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12028–12037.

Ohn-Bar, E. and Trivedi, M. M. (2013). Joint angles similarities and hog2 for action recognition. *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Ohn-Bar, E. and Trivedi, M. M. (2014). Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Transactions on Intelligent Transportation Systems*, 15:2368–2377.

Oreifej, O. and Liu, Z. (2013). Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723.

Rahmani, H. and Mian, A. S. (2016). 3d action recognition from novel viewpoints. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1506–1515.

Ramirez-Amaro, K., Beetz, M., and Cheng, G. (2017). Transferring skills to humanoid robots by extracting semantic representations from observations of human activities. *Artif. Intell.*, 247:95–118.

Rastgoo, R., Kiani, K., and Escalera, S. (2020). Hand sign language recognition using multi-view hand skeleton.

Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2018). Non-local graph convolutional networks for skeleton-based action recognition.

Si, C., Chen, W., Wang, W., Wang, L., and Tan, T. (2019). An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *CVPR*.

Smedt, Q. D., Wannous, H., and Vandeborre, J.-P. (2016). Skeleton-based dynamic hand gesture recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1206–1214.

Sridhar, S., Feit, A. M., Theobalt, C., and Oulasvirta, A. (2015). Investigating the dexterity of multi-finger input for mid-air text entry. In *CHI '15*.

Surie, D., Pederson, T., Lagriffoul, F., Janlert, L.-E., and Sjölie, D. (2007). Activity recognition using an egocentric perspective of everyday objects. In *UIC*.

Tang, Y., Tian, Y., Lu, J., Li, P., and Zhou, J. (2018). Deep progressive reinforcement learning for skeleton-based action recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5323–5332.

Tekin, B., Bogo, F., and Pollefeys, M. (2019). H+o: Unified egocentric recognition of 3d hand-object poses and interactions. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4506–4515.

Vemulapalli, R., Arrate, F., and Chellappa, R. (2014). Human action recognition by representing 3d skeletons as points in a lie group. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595.

Wang, H. and Wang, L. (2017). Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3633–3642.

Wang, L., Huynh, D. Q., and Koniusz, P. (2019). A comparative review of recent kinect-based action recognition algorithms. *IEEE Transactions on Image Processing*, 29:15–28.

Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

Ying, X. (2019). An overview of overfitting and its solutions.

Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Yong Chang, J., Mu Lee, K., Molchanov, P., Kautz, J., Honari, S., Ge, L., Yuan, J., Chen, X., Wang, G., Yang, F., Akiyama, K., Wu, Y., Wan, Q., Madadi, M., Escalera, S., Li, S., Lee, D., Oikonomidis, I., Argyros, A., and Kim, T.-K. (2018). Depth-based 3d hand pose estimation: From current achievements to future goals. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zanfir, M., Leordeanu, M., and Sminchisescu, C. (2013). The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. *2013 IEEE International Conference on Computer Vision*, pages 2752–2759.

Zhang, C., Yang, X., and Tian, Y. (2013). Histogram of 3d facets: A characteristic descriptor for hand gesture recognition. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8.

Zhang, X., Qin, S., Xu, Y., and Xu, H. (2019). Quaternion product units for deep learning on 3d rotation groups. *ArXiv*, abs/1912.07791.

Zhang, X., Wang, Y., Gou, M., Sznaier, M., and Camps, O. (2016). Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.