Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric Hand Gestures

Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric Hand Gestures

Danilo Avola, *Member, IEEE,* Marco Bernardi, *Student Member, IEEE,* Luigi Cinque, *Senior Member, IEEE,* Gian Luca Foresti, *Senior Member, IEEE,* and Cristiano Massaroni, *Student Member, IEEE*

***Abstract*—Hand gesture recognition is still a topic of great interest for the computer vision community. In particular, sign language and semaphoric hand gestures are two foremost areas of interest due to their importance in Human-Human Communication (HHC) and Human-Computer Interaction (HCI), respectively. Any hand gesture can be represented by sets of feature vectors that change over time. Recurrent Neural Networks (RNNs) are suited to analyse this type of sets thanks to their ability to model the long term contextual information of temporal sequences. In this paper, a RNN is trained by using as features the angles formed by the finger bones of the human hands. The selected features, acquired by a Leap Motion Controller (LMC) sensor, are chosen because the majority of human hand gestures produce joint movements that generate truly characteristic corners. The proposed method, including the effectiveness of the selected angles, was initially tested by creating a very challenging dataset composed by a large number of gestures defined by the American Sign Language (ASL). On the latter, an accuracy of over 96% was achieved. Afterwards, by using the SHREC dataset, a wide collection of semaphoric hand gestures, the method was also proven to outperform in accuracy competing approaches of the current literature.**

***Index Terms*—Hand gesture recognition, sign language, semaphoric gestures, leap motion controller (LMC), recurrent neural network (RNN), Long Short Term Memory (LSTM).**

I. Introduction

HANDS can express a wide range of information thanks to the many gestures that their fingers can compose. Different categorizations of hand gestures can be defined depending on the type of information that the hands intend to transmit. Based on the researches of Kendon [1] and Quek *et al.* [2], a possible taxonomy of hand gesture categories can be proposed as follows:

- **Deictic** are the hand gestures that involve a pointing activity to establish the identity or spatial location of an object within the context of an application domain;
- **Manipulative** are usually performed by freehand movements to mimic manipulations of physical objects, such as in virtual or augmented reality interfaces;
- **Semaphoric** are specific hand gestures that define a set of commands and/or symbols to interact with machines. They are often used alternatively to the speech modality, when the latter is unusable or ineffective;
- **Gesticulation** is one of the most natural forms of gesturing. It is commonly used in combination with conversational speech interfaces. These hand gestures are often unpredictable and difficult to analyse;
- **Language** are the hand gestures used for sign language. They are performed by combining a set of gestures to form grammatical structures for conversational style interfaces. In case of finger spelling, these gestures can be considered like semaphoric ones.

Hand gesture recognition provides a means to decode the information expressed by the reported categories, which are always more used to interact with innovative applications, such as interactive games [3], [4], serious games [5], [6], sign language recognisers [7], [8], [9], [10], emotional expression identifiers [11], [12], remote controllers in robotics [13], [14], advanced computer interfaces [15], [16], [17], [18], and others. In general, the approaches used in hand gesture recognition can be divided into two main classes: 3D model-based [19] and appearance-based [20]. The first uses key elements of the body parts to acquire relevant 3D information, while the second uses images or video sequences to acquire key features. In the past, several RGB cameras were necessary to obtain a 3D model of the body parts, including hands. Recent works, supported by advanced devices, e.g., Microsoft Kinect [21] or LMC [22], as well as novel modelling algorithms based on depth map concept [23], have enabled the use of 3D models within everyday application domains.

In this paper, a 3D model-based method for the recognition of sign language and semaphoric hand gestures is presented. Specifically, the proposed approach uses a skeletal-based modelling, where a virtual representation of the skeleton hands (or, in general, of other parts of the body) is mapped to specific segments. This technique uses joint angle parameters along with segment lengths, instead of intensive processing of all 3D model parameters. Then, it measures the variations over time of the skeleton joints whose spatial coordinates are acquired by a LMC. In particular, the angles formed by a specific subset of joints that involve distal, intermediate, and proximal phalanges for the index, middle, ring, and pinky, as well as the metacarpal for the thumb, can be considered highly discriminating to recognize many types of hand gestures, as confirmed by our tests. Notice that, these features were selected as they are easy and quick to be extracted. Spatial information about the fingertips are also considered by the method to manage not articulated movements of the hands. Finally, to obtain a more accurate classifier, the proposed approach also takes into account the information of the intra-finger angles and the spatial data of the palm of the hand. During the design of the

proposed method, the following two challenges were fixed:

- the search for a robust solution usable in real contexts and able to also recognize hand gestures that are similar to each other;
- the achievement of the highest accuracy level compared with competing works of the current state-of-the-art.

The goals reported above were obtained by using a stack of RNNs [24] with Long Short Term Memory (LSTM) [25] architecture, a particular type of Deep Neural Network (DNN) where connections between units form a directed cycle within the same layer. The RNNs, unlike the common DNNs, can model long term contextual information of temporal sequences, thus obtaining excellent results in fields such as sound analysis and speak recognition, as reported in [26]. The LSTM is an architecture where a RNN uses special units instead of common activation functions. LSTM units help to propagate and preserve the error through time and layers. This aspect of the LSTMs allows the net to learn continuously over many time steps, thereby opening a channel to link causes and effects remotely. An architecture formed by two or more stacked LSTM RNNs is defined as Deep LSTM (DLSTM). Such an architecture allows to learn at different time scales over the input sequences [27]. Initially, in the experimental session, the method was tested by creating a dataset composed by a challenging subset of hand gestures defined by the ASL [28]. The latter was chosen because it is composed of a wide range of hand gestures with a high degree of complexity. Afterwards, by using the wide collection of semaphoric hand gestures contained in the SHREC dataset [29], the proposed method was also compared with competing works of the current literature. Summarizing, the method reported in this paper presents the following contributions:

- the selection of a simple set of features, based on the joint angles, that are highly discriminative for the recognition of any type of hand gesture, especially for sign language and semaphoric hand gestures;
- the creation, by the LMC, of a large dataset to support the comparison of sign language recognizers based on the hand skeleton model. Notice that, the LMC guarantees a high precision in the estimation of the joint positions [30];
- the capability of analysing and recognizing a large number of hand gestures in two main areas of interest like sign language and semaphoric hand gestures. Notice that, the study of static and dynamic hand gestures of the ASL provides a prerequisite for achieving wider recognition systems for the sign language;
- for the first time in hand gesture recognition field, the use of the DLSTM, in combination with the hand skeleton extracted by a LMC, is proposed. Furthermore, even more important, an accuracy of over 96% on the created sign language based dataset and a comparison on the SHREC dataset that outperforms in accuracy competing approaches of the current literature, are reported.

The rest of the paper is structured as follows. In Section II, the state-of-the-art of the hand gesture recognition is presented. The proposed method is described in Section III. Extensive experimental results and comparisons with competing works are discussed in Section IV. Finally, conclusions are drawn in Section V.

II. Related Work

In the current literature, hand and body gesture recognition is based on a conventional scheme: the features are acquired from one or more sensors (such as, Kinect [31], [32], [33], LMC [34], [8]) and machine learning techniques (e.g., Support Vector Machines (SVMs) [35], [36], Hidden Markov Models (HMMs) [37], [38], Convolutional Neural Networks (CNNs) [39], [26]) are used to perform a classification phase. A reference work is reported in [35], where an SVM is used with Histogram of Oriented Gradients (HOGs) as feature vectors. Wang *et al.* [36] and Suryanarayan *et al.* [40] used an SVM with volumetric shape descriptors. Using the same classifier, Marin *et al.* [9] applied a combination of features extracted by Kinect and LMC sensors. Other interesting solutions are based on HMMs, such as that proposed in Zun *et al.* [41], where a robust hand tracking to recognize hand signed digit gestures is reported.

Different well-known techniques are extended and customized to reach increasingly better results. An example is shown in [7], where a semi-Markov conditional model to perform finger-spelling gesture recognition on video sequences is presented. The Hidden Conditional Random Field (HCRF) method, proposed in Wang *et al.* [37], is instead used to recognize different human gestures. Lu *et al.* [8] use an extension of the HCRF to recognize dynamic hand gestures driven by depth data. Regarding the hand pose estimation, the solution proposed in Li *et al.* [38] shows excellent results by applying a Randomized Decision Tree (RDT).

Another common solution is based on the use of Dynamic Time Warping (DTW). Although DTW does not belong to the class of machine learning techniques, it is often used in time series classification. In Vikram *et al.* [34], a DTW to support a handwriting recognition process based on the trajectory of the fingers extracted by a LMC is presented. In [42], the DTW with a novel error metric to match patterns, combined with a statistical classifier, is used to perform a tool to aid the study of basic music conducting gestures. In Sohn *et al.* [10], a pattern matching method by the combination of a DTW and a simple K-Nearest Neighbor (K-NN) classifier is used.

Recently, the great performance of the deep neural networks has motivated the use of the CNNs in different application domains, including the gesture recognition as proposed in [39]. Moreover, analysing the behaviour of these nets in other fields [24], [25], [26], we have understood that the RNNs can be suitably used to support the classification of temporal data sequences. In addition, some recent works, such as [43], have shown how the LSTMs are potentially more effective than CNNs in recognizing gestures. Based on these observations, the proposed method was designed starting from two works that achieve outstanding results in the current literature: the first, proposed by Du *et al.* [44], where an hierarchical RNN for skeleton action recognition is used, and the second, proposed by Graves *et al.* [26], that uses a Deep Bidirectional LSTM for the speech recognition.

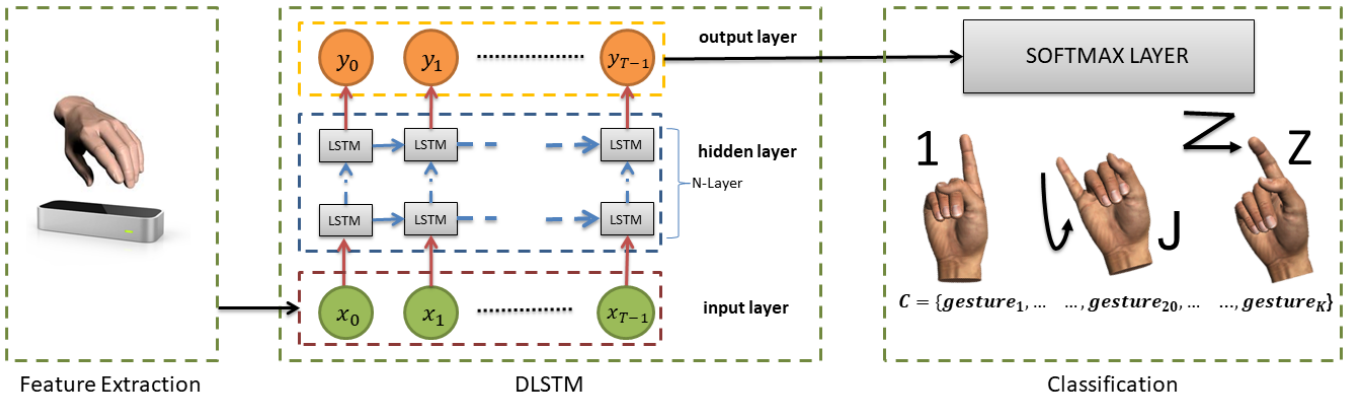


Fig. 1: Logical architecture of the proposed method. The training phase is performed by a DLSTM with two stacked LSTM RNNs. Given a sequence of input vectors, the DLSTM returns an output vector for each time instant t, with $0 \leq t \leq T-1$, that contains the probabilities for each class. K and T are the different classes of the hand gestures and the maximum number of time instants in which a gesture is acquired, respectively.

III. Method

Let us consider, each hand gesture acquired by a user is represented by a set $X = \{x_0, x_1, ..., x_{T-1}\}$ of feature vectors, where T indicates the maximum number of time instants, inside a time interval Θ, in which the features are extracted by a LMC. Notice that, a LMC is chosen as reference device for the acquisitions because it is optimized for the hands and the obtained skeleton model provides very accurate dynamic information about finger bones [45]. A DLSTM is applied to model these sequences of data, where a time series of feature vectors (one vector for each time instant) is converted into a series of output probability vectors $Y = \{y_0, y_1, ..., y_{T-1}\}$. Each $y_t \in Y$ indicates the class probability of the gesture carried out at time t, with $0 \leq t \leq T-1$. Finally, the classification of the gestures is performed by a softmax layer [46] using $K = |C|$ classes, where C is the set of the considered gesture classes. The logical architecture of the proposed method is shown in Fig. 1.

A. Feature Extraction

Each gesture can be considered as the composition of different poses, where each pose is characterized by particular angles. Such a concept has already been applied in several works, using the angles formed by the body joints to recognize human actions [47], [48], [49]. So, each feature vector $x_t \in X$, with $0 \leq t \leq T-1$, is mainly composed by (Fig. 2):

- the internal angles ω_1, ω_2, ω_3, and ω_4 of the joints between distal phalanges and intermediate phalanges. The internal angle ω_0, considered for the thumb, is computed between distal phalanx and proximal phalanx;
- the internal angles β_1, β_2, β_3, and β_4 of the joints between intermediate phalanges and proximal phalanges. The internal angle β_0, considered for the thumb, is computed between proximal phalanx and metacarpal.

Each finger can be seen as a set of segments, where $\overline{CD}$ is the distal phalanx, $\overline{BC}$ is the intermediate phalanx (with the exception of the thumb, where $\overline{BC}$ is the proximal phalanx), and $\overline{AB}$ is the proximal phalanx (with the exception of the thumb, where $\overline{AB}$ is the metacarpal). The angles are calculated as follows:

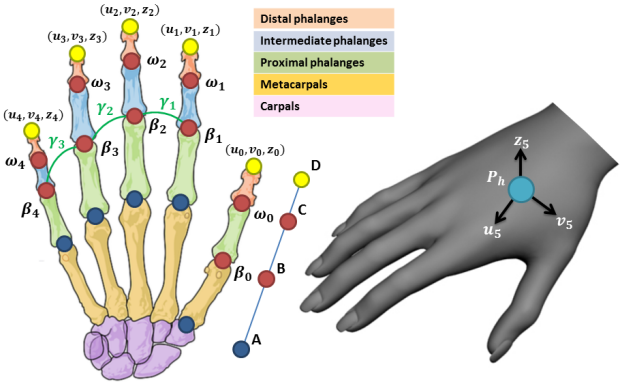


Fig. 2: The features extracted from the hand: joint angles and fingertip positions. The yellow points indicate the fingertip positions on which the 3D displacements are computed. The red points indicate the joints on which the angles are computed.

$$\omega_j = \arccos\left(\frac{\overline{BC} \cdot \overline{CD}}{|\overline{BC}||\overline{CD}|}\right) \quad (1)$$

$$\beta_j = \arccos\left(\frac{\overline{AB} \cdot \overline{BC}}{|\overline{AB}||\overline{BC}|}\right) \quad (2)$$

where, $j = 0, .., 4$. Since the information provided by the angles is not sufficient to manage all types of existing hand gestures, especially dynamic gestures that perform movements in 3D space, additional information is used by considering the following features:

- 3D displacements u_5, v_5, z_5 of the position of the central point P_h of the palm of the hand. These features are considered to manage hand translation on the 3D space;
- 3D displacements u_l, v_l, z_l of the fingertip positions, with $l = 0, .., 4$. These features are considered to manage hand rotation in 3D space;

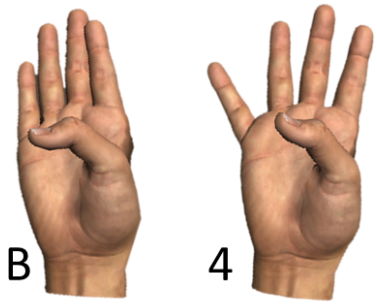


Fig. 3: Example of static gestures differentiated by the intra-finger angles γ_1, γ_2, and γ_3.

- the intra-finger angles γ_1, γ_2, and γ_3, i.e., the angles between two consecutive fingers, where the fingers considered are: the pointer finger, the middle finger, the ring finger, and the pink finger. These features are used to handle special cases of static gestures that differ from each other only in intra-finger angles, as shown in Fig. 3.

All the reported features are independent by the reference. Thus, the input vector assigned to the DLSTM at time t is:

$$x_t = \{\omega_0, ..., \omega_4, \beta_0, ..., \beta_4, u_0, v_0, z_0, ..., u_5, v_5, z_5, \gamma_1, \gamma_2, \gamma_3\} \quad (3)$$

B. Sampling Process

Since each person can perform the same gesture with different speeds, and since the proposed method requires that all the videos that must be analysed are composed by the same number T of samples, a sampling process to select the most significant feature values within the entire time interval Θ of the hand gesture sequences was implemented. This means that data are acquired only in the most significant T time instants, where a time instant $t \in \Theta$ is defined as significant when the joint angles and the hand central point position P_h vary substantially between t and $t+1$ (as explained below).

Let $f_{\omega_i}(t), f_{\beta_i}(t)$, and $f_{\gamma_j}(t)$, with $0 \leq i \leq 4$ and $1 \leq j \leq 3$, be the functions that represent the value of ω_i, β_i, and γ_j angles at time t. In addition, let $f_{\phi_{(t)}}$ be the function that represents the value of ϕ (i.e., the displacement of the centre of the hand P_h with respect to the previous position at time $t-1$) at time t. Then, for each function $f_g(t)$, with $g \in G$ and $G = \{\omega_i, \beta_i, \gamma_j, \phi\}$, the Savitzky-Golay filter [50] is applied. The Savitzky-Golay filter is a digital filter able to smooth a set of digital data in order to increase the signal-to-noise ratio without greatly distorting the signal. Now, the significant variations on the considered features are identified through the relative maximum and minimum of each $f_g(t)$. All the time instants t, associated with at least one relative minimum or relative maximum of a feature g, are used to create a new set Θ^*, which represents a set of possible important time instants to be sampled. In Fig. 4, an example of this sampling phase is shown, where the behaviour of the function $f_{\omega_1}(t)$ (i.e., the angle of the distal phalanx of the index finger) for an instance of the gesture "milk" is considered. By applying the Savitzky-Golay filter, the signal shown in Fig. 4, that is affected by a certain amount of noise due to the acquisition

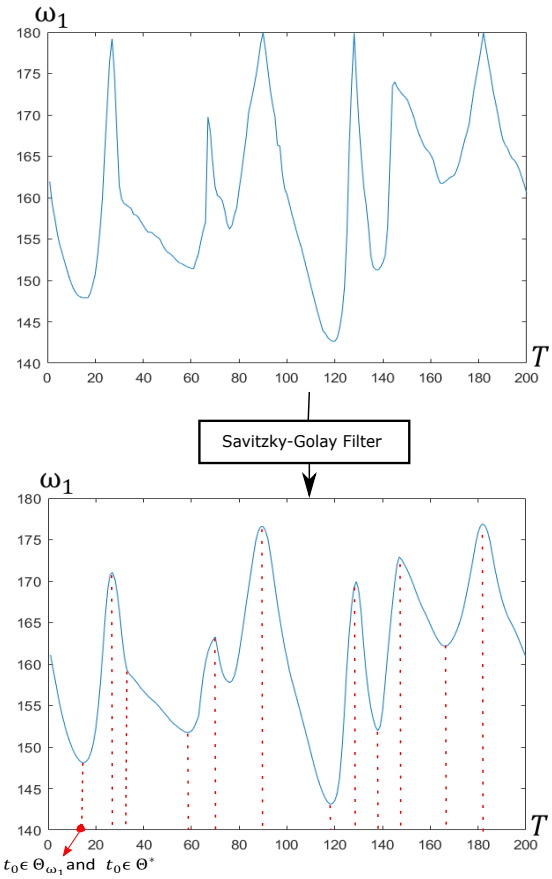


Fig. 4: Sampling example for the feature ω_1 on the "milk" gesture.

device or tremors of the hand, can be suitably cleaned. Then, the maximum and minimum relative points are identified and sampled. In the example, only the procedure for the feature ω_1 is shown, but this step is performed for each feature $g \in G$. Now, depending on the cardinality of the set of the sampled time instants, the following cases must be considered:

- if $|\Theta^*| < T$, then the remaining $(|\Theta^*| - T)$ time instants to be sampled are randomly selected in Θ;
- if $|\Theta^*| > T$, then, only some significant time instants are sampled for each g feature. Let Θ_g be the set of the samples in Θ^* obtained from the relative maximum and minimum of the feature g ($\Theta_g \subseteq \Theta^*$), we need to know the number of time instants T_g that can be sampled for each g such that $\sum_{g \in G} T_g = T$. Each T_g is obtained thought the following proportion $|\Theta_g| : |\Theta^*| = T_g : T$. Then, from each Θ_g set, we randomly take T_g samples.

After the sampling step, each acquisition instance is composed by a sequence $\{x_0, .., x_{T-1}\}$ of feature vectors. The proposed sampling procedure is dynamically based on the value of the features.

C. Deep Last Short Term Memory Network

A fundamental component in the proposed work is the network used in the classification of the hand gestures. This network is based on multiple LSTMs, which unlike other types

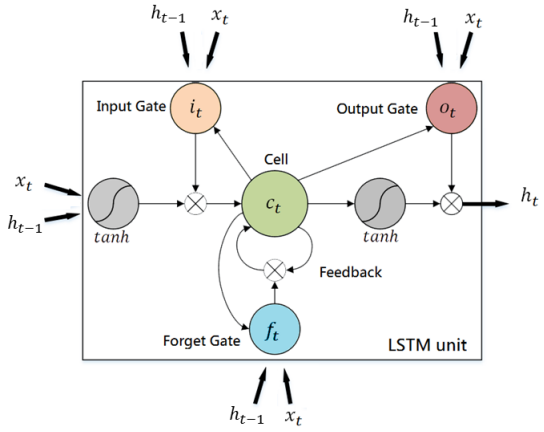


Fig. 5: Example of a LSTM unit. The internal state is maintained with a recurrent connection. The input gate i_t (orange) and the output gate o_t (red) scale the input and output of the cell c_t, while the forget gate f_t (azure) scales the internal state.

of Neural Networks (NNs), are able to efficiently analyse time sequences of data. Several factors, such as the error blowing up problem [51] and the vanishing gradient [52], do not allow the use of common activation functions (e.g., tanh or sigmoid) to suitably train a network composed by multiple RNNs. This problem can be tackled with the LSTM units (Fig. 5).

The LSTM can be seen as memory blocks that are one or more self-connected memory cells and three multiplicative units: the input, output, and forget gates. These gates provide continuous analogues of write, read, and reset operations for the cells. Although an LSTM allows to manage the problem of the vanishing gradient, the input time series often have a temporal hierarchy, with information that is spread out over multiple time scales which can not be adequately recognized by simple recurrent networks such as LSTMs. For this reason, Deep LSTMs were introduced. In fact, by constructing recurring networks formed over multiple layers, a higher abstraction on the input data is reached [27]. Increased input abstraction does not always bring benefits, because the effectiveness of these networks depends on both task and analysed input.

In several works, such as [26], [53], [54], it was observed empirically that Deep LSTMs work better than shallower ones on speech recognition. The audio signals, analysed for example in speech-to-text task, can be elaborated on more abstractions ranging from the entire pronounced phrase to the syllables of each word. Moreover, each abstraction can be captured in different time scales within the considered period. Like in the case of audio sequences analysed in the speech recognition problem, hand gestures can be examined over multiple time scales. In fact, each gesture can be considered as composed by many small movements and sub-gestures of the hand and, as observed, this type of data processing is particularly suitable for this kind of network.

Based on these considerations, the LSTM stack-based solution was experimented and then compared to the performance of a single-level network. The first step was the definition of the activation functions of memory cell of the $LSTM_0$ (the first layer of the proposed neural network), as well

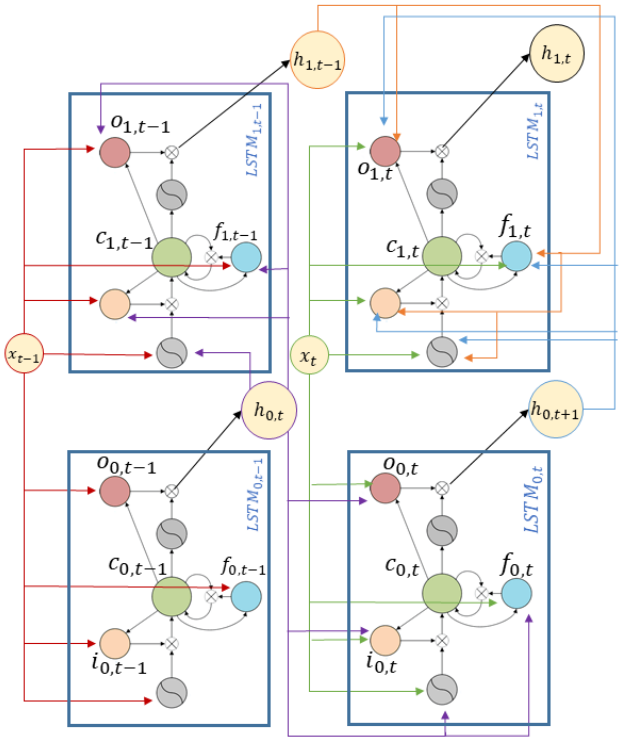


Fig. 6: Example of connections between two stacked LSTMs, where the first level is placed at the bottom of the image and it is represented by the $LSTM_0$. For each level, the units that handle the input x_{t-1} and x_t are shown.

as the computation of the input, output, and forget gates determined by evaluating iteratively the following equations (from $t = 0 \;\; to \;\; T-1$):

$$i_{0,t} = \sigma(W_{xi}x_t + W_{hi}h_{0,t-1} + W_{ci}c_{0,t-1} + b_i) \quad (4)$$

$$f_{0,t} = \sigma(W_{xf}x_t + W_{hf}h_{0,t-1} + W_{cf}c_{0,t-1} + b_f) \quad (5)$$

$$c_{0,t} = f_t \odot c_{t-1} + i_{0,t} \odot \tanh(W_{xc}x_t + W_{hc}h_{0,t-1} + b_c) \quad (6)$$

$$o_{0,t} = \sigma(W_{xo}x_t + W_{ho}h_{0,t-1} + W_{co}c_{0,t-1} + b_o) \quad (7)$$

$$h_{0,t} = o_{0,t} \odot \tanh(c_{0,t}) \quad (8)$$

where, i, f, o, and c denote the input gate, forget gate, output gate, and cell activation vectors, respectively. These vectors have the same length of the hidden vector h. Instead, W_{xi}, W_{xf}, W_{xo}, and W_{xc} are the weights of the input gate, forget gate, output gate and cell to the input. In addition, W_{ic}, W_{fc}, and W_{oc} are the diagonal weights for peep-hole connections. Finally, the terms b_i, b_f, b_c, and b_o indicate the input, forget, cell and output bias vectors, respectively. We have that σ is the logistic sigmoid function and $\odot$ is the element-wise product of the vectors. Once the activation functions for the first level are defined, the next step is to define the upper level activation functions.

DLSTMs are architectures obtained by stacking multiple LSTM layers where the output sequence h_l of one layer l forms the input sequence for the next layer $l+1$ (Fig. 6). The memory cell of an $LSTM_l$ at time t, in addition to the classic x_t and $h_{l,t-1}$ vectors, takes in input the $h_{l-1,t}$, i.e., the hidden state at time t of the below $LSTM_{l-1}$. So, the activations of

the memory cells of $LSTM_l$ of the network higher levels (i.e., $l > 0$) are given by the following equations:

$$i_{l,t} = \sigma(W_{xi}x_t + W_{h_l i}h_{l,t-1} + W_{h_{l-1}i}h_{l-1,t} + W_{ci}c_{l,t-1} + b_i) \quad (9)$$

$$f_{l,t} = \sigma(W_{xf}x_t + W_{h_l f}h_{l,t-1} + W_{h_{l-1}f}h_{l-1,t} + W_{cf}c_{l,t-1} + b_f) \quad (10)$$

$$c_{l,t} = f_t \odot c_{t-1} + i_{l,t} \odot \tanh(W_{xc}x_t + W_{h_l c}h_{l,t-1} + W_{h_{l-1}c}h_{l-1,t} + b_c) \quad (11)$$

$$o_{l,t} = \sigma(W_{xo}x_t + W_{h_l o}h_{l,t-1} + W_{h_{l-1}o}h_{l-1,t} + W_{co}c_{l,t-1} + b_o) \quad (12)$$

$$h_{l,t} = o_{l,t} \odot \tanh(c_{l,t}) \quad (13)$$

The output of the DLSTM network, at time t, with N-layers, is defined as follows:

$$y_t = W_{h_{N-1},t}h_{N-1,t} + b_y \quad (14)$$

where b_y is a bias vector, $h_{N-1,t}$ is the hidden vector of the last layer, and $W_{h_{N-1},t}$ is the weight from the hidden layer $h_{N-1,t}$ to output layer. The output y_t defines a probability distribution over the K possible gesture classes, where y_t^k (i.e., the k^{th} element of y_t) is the estimated probability of a specific class C_k at time t for the acquired gesture X. Finally, all results y_t are collected and normalized into the softmax layer, through the following equations:

$$\hat{y} = \sum_{t=0}^{T-1} y_t \quad (15)$$

$$\tilde{y}^k = p(C_k|X) = \frac{e^{\hat{y}^k}}{\sum_{q=0}^{K-1} e^{\hat{y}^q}} \quad (16)$$

for each k, with $1 \leq k \leq K$. The classification of the gesture X will be given by the highest probability contained in $\tilde{y}$.

D. Network Training

Given a dataset D composed of M train gesture sequences, the goal is to minimize the following maximun-likelihood loss function:

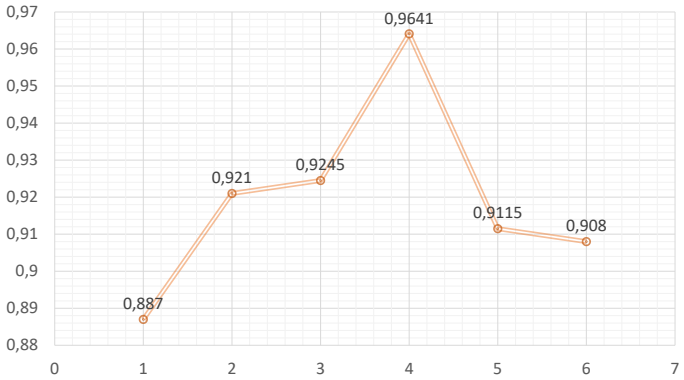
$$\mathcal{L}(D) = -\sum_{m=0}^{M-1} \ln \sum_{k=0}^{K-1} \delta(k,\tau)p(C_k|D_m) \quad (17)$$

where, D_m, $0 \leq m \leq M$, is an input sequence of the training dataset D, τ is the ground-truth label of D_m, and $\delta(\bullet,\bullet)$ is the Kronecker delta or delta function. This formulation is referred to the cross-entropy error proposed in [55]. The Back-Propagation Through Time (BPTT) algorithm [52] is used to obtain the objective function derived with respect to all the weights and to compute the minimization based on the stochastic gradient descent.

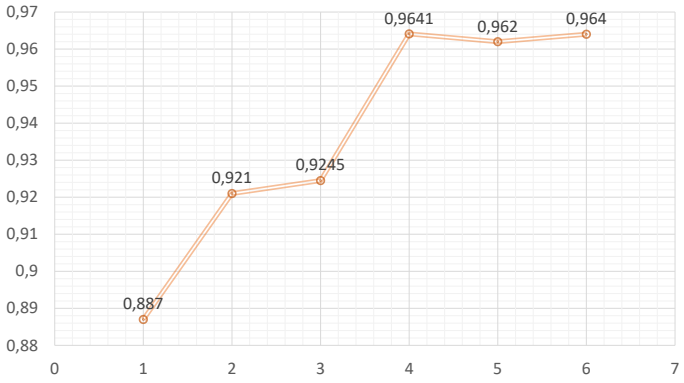
IV. Experimental Results

This section describes the experimental tests performed to evaluate the performance of the proposed approach. All the experiments were executed by using a LMC on an Intel i5 3.2GHz, 16GB RAM, with a GeForce GTX 1050ti graphics card. The DLSTM network and the BPTT algorithm, used to compute the minimization based on the stochastic gradient descent, were implemented by using the Keras[1] framework.

[1]https://keras.io/



(a)



(b)

Fig. 7: Accuracy results on the proposed dataset by varying the number of stacked LSTMs in the network architecture: (a) accuracy results using 800 epochs for each considered architecture and (b) accuracy results using 800 epochs for $1-LSTM$, $2-LSTM$, $3-LSTM$, and $4-LSTM$; for the $5-LSTM$ and $6-LSTM$ are used 1600 and 1800 epochs, respectively. The x-axis indicates the number of the stacked LSTMs, while the y-axis indicates the accuracy values.

The main aims of the experimental session were both the validation of the proposed method, including the assessment of the joint angles as salient features for the hand gesture recognition, and the outperforming of competing works of the current state-of-the-art. The achievement of the first goal was obtained by creating a challenging dataset based on the sign language (Section IV-A) on which the optimal number of stacked LSTMs (Section IV-B) and the effectiveness of the selected joint features (Section IV-C) were analysed. In addition, on the same dataset, a set of well-known metrics was computed to evaluate the overall performance of the approach (Section IV-D). Instead, the second goal was obtain by comparing the proposed method with other considerable works on the basis of the SHREC dataset (Section IV-E).

A. ASL Dataset

Currently, there are no public datasets, with a large number of classes and with information on the hand joints, that allow to test approaches like that we propose. For this reason, we created a new dataset composed of 30 hand gestures. In

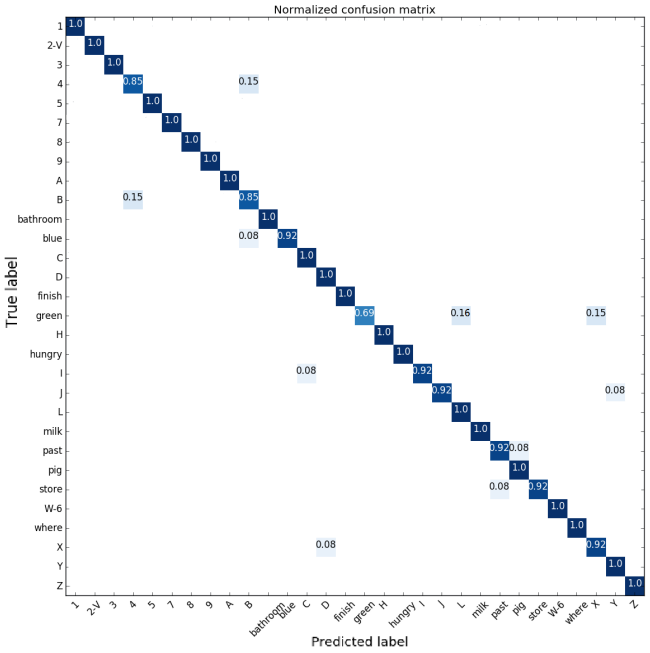


Fig. 8: The confusion matrix related to the proposed gesture dataset. The overall accuracy is 96.4102%.

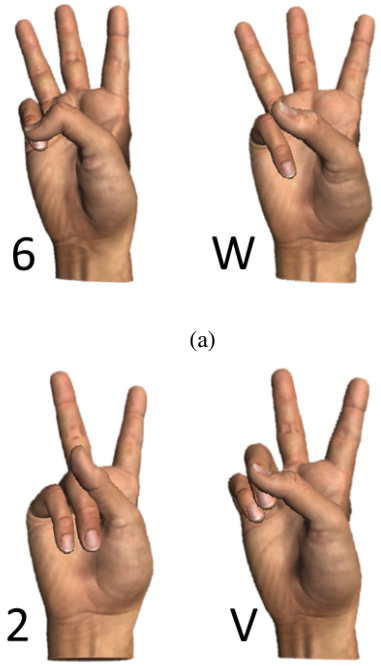


Fig. 9: Pairs of gestures joined into a common class: (a) 6 and W hand gestures, 2 and V hand gestures.

particular, the dataset consists of 12 dynamic gestures and 18 static gestures taken by the ASL. These gestures were chosen to stress the ability of the method in learning the variations of both joint angles and finger positions that occur when a hand performs a complex hand gesture. The static gestures are: 1, 2-V, 3, 4, 5, 6-W, 7, 8, 9, A, B, C, D, H, I, L, X, and Y. Instead, the dynamic gestures are: bathroom, blue, finish, green, hungry, milk, past, pig, store, and where.

The dataset is composed of 1200 hand gesture sequences, coming from 20 different people. Each gesture was collected by 15 males and 5 females, aged 20 to 28 years. Each person performed the 30 different hand gestures twice, once for each hand. The sequences from 13 people were used to create the training set, while the sequences of the remaining 7 people were used to form the test set. So, the 7 people used in the tests were never taken into consideration during the training phase. As previously described in Section III-B, each sequence was acquired according to a sampling process, with $T = 200$ and $\Theta = 5s$.

B. Selection of the Optimal Number of Stacked LSTMs

Several tests were conducted to choose the optimal number of stacked LSTMs needed to be used in the proposed architecture. The hidden units per LSTM were fixed to 200, i.e., the hidden units were fixed equal to the number of input time instances considered for each gesture (i.e., $T = 200$). Fig. 7a shows as an architecture composed by 4 levels provides the best accuracy results by using 800 epochs. In fact, although several levels of an LSTM allow to analyse complex time sequences by dividing them into multiple time scales, the $5 - LSTM$ and the $6 - LSTM$ require more epochs to be trained. Increasing the number of epochs needed to train the $5 - LSTM$ and $6 - LSTM$ architectures (i.e., 1600 epochs for the $5 - LSTM$ and 1800 for the $6 - LSTM$), the Fig. 7b shows how their results improves. We can notice how greater abstraction on input does not provide substantial benefits from a certain number of levels, and the accuracy gained by the network begins to converge to a fixed value.

In conclusion, 4 levels are appropriated for the proposed network and represent a good compromise between training time and accuracy. The choice of the learning rate influences the speed of the convergence of the cost function. If the learning rate is too small, the convergence is obtained slowly, while if the learning rate is too large, the cost function may not decrease in each iteration and therefore it could not converge. In the proposed method, the learning rate was set to 0.0001 through large empirical tests.

C. Effectiveness of the Selected Features

To verify the effectiveness of the features in classifying the set of gestures taken by the ASL dataset, various tests were carried out. The adopted network was composed by 4 stacked LSTMs (as explained in Section IV-B). Two different sets of hand gesture sequences of the ASL dataset (i.e., a set for the training and a separate set for the classification, respectively) were used to check the contribute of subsets of x_t as features.

The results in Table I shows that the combination of the features ω_i and β_i is sufficient to discriminate a high number of hand gestures. Notice that, this combination reaches better classification results with respect to the use of these two features separately. Although the single γ_j feature does not offer good performance, it greatly improves the classification when used with ω_i and β_i. Instead, the combination of features

	ω_i	β_i	γ_j	u_w, v_w, z_w	ω_i, β_i	$\omega_i, \gamma_j, \beta_i$
Accuracy%	62.70%	68.1204 %	46.67%	56.92%	79.74%	85.13%

TABLE I: Accuracy of the proposed solution obtained on the ASL dataset by varying the different features, where $0 \leq i \leq 4$, $0 \leq j \leq 3$, and $0 \leq w \leq 5$.

Accuracy	Precision	Recall	F1-Score
96.4102%	96.6434%	96.4102%	96.3717%

TABLE II: Performance of the method on the ASL dataset using Accuracy, Precision, Recall, and F1-Score metrics.

related to the hand movements (u_w, v_w, z_w) are unable by themselves to classify the hand gestures but, if combined with the features of the joint angles, allow the method to achieve high performance (as discussed below).

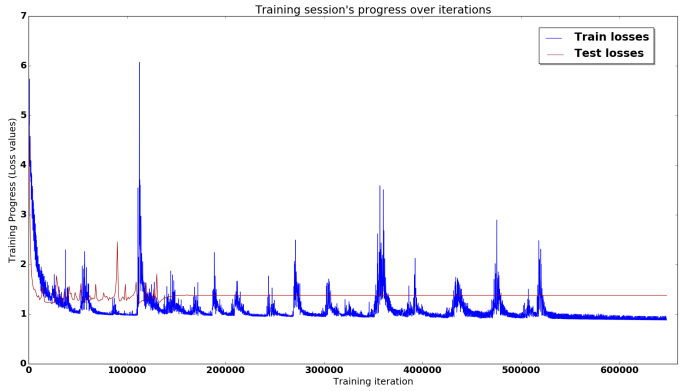
D. Hand Gesture Recognition on the ASL Dataset

To evaluate the method, we used very popular metrics: *Accuracy*, *Precision*, *Recall*, and *F1-score*. These metrics can be considered a de facto standard to measure the quality of this class of algorithms [56]. The results are presented in Table II.

According to the tests performed to recognize the different hand gestures, and to better analyse the proposed method, also the confusion matrix was computed (Fig. 8). Each column of the matrix represents the instances in a predicted gesture, instead each row represents the instances in a current gesture. The main diagonal of the matrix represents the instances correctly classified by the DLSTM. The elements below the diagonal are the false positives, i.e., the gestures that are incorrectly classified within a class of interest. The elements above the diagonal are the false negatives, i.e., the gestures incorrectly classified as not belonging to a class of interest. The distinction of some gestures is very hard, since they are very similar to other gestures in the dataset. Despite this, the proposed method does not suffer of ambiguity issues. The only exceptions are given by the gestures 6 with W (Fig. 9a) and 2 with V (Fig. 9b). The variations in their joint angles are minimal and difficult to see even to the human eye. Moreover, the LMC device fails to capture these variations. For this reason, these gestures have been grouped in the same class. To quantify the difficulty in recognizing these gestures, we have also performed tests without grouping these classes, thus obtaining an accuracy of 91.5178%. This decrease, with respect to the overall accuracy of 96.4102%, is due to the incorrect classifications related to the classes: 2, 6, v, and W.

In Fig. 10, the Train/Test plots are shown. The first plot (Fig. 10a) shows the Train/Test accuracy over the iterations, instead the second plot (Fig. 10b) contains the loss curves that represent the sum of the errors provided for each training or test instance. In this work, the loss curves are calculated as maximum-likelihood loss function, described in Section III-D. Instead, the curves of accuracy represent the training or validation instances correctly recognized. After a certain number of iterations ($\sim$ 125000), the test accuracy curve converges.

(a)



(b)

Fig. 10: Train/Val Curves: the progress of training and testing over the iterations based on: (a) the accuracy and (b) the loss. After 100000 iterations, the test accuracy curve converges. The x-axis represents the progress of training/validation stage and the y-axis represents the number of training iterations.

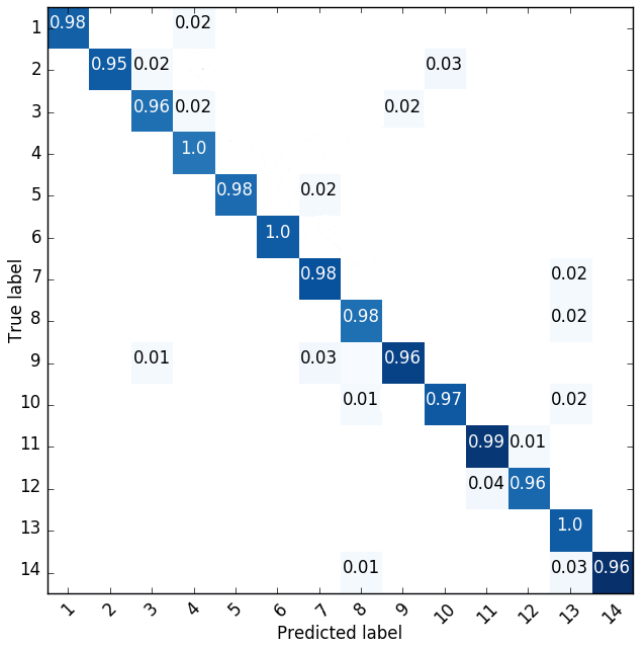
E. Comparisons

We compared the proposed method with significant works of the current state-of-the-art presented in [57], [59], [58], [48], [29] on the SHREC dataset [29]. The SHREC dataset was selected since: (a) it provides different types of data to allow comparisons between methods based on different acquisition sensors; (b) it allows the classification of hand gestures with different degrees of complexity; (c) it provides data that allow to extract all the features necessary for the proposed method. Notice that, the SHREC dataset contains very challenging semaphoric hand gesture sequences, and the evaluation of the method on this type of gestures is one of the main targets of the presented work.

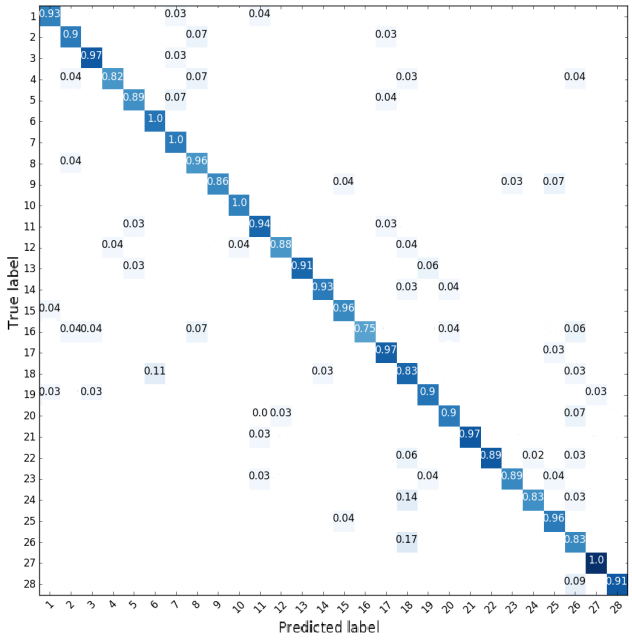
The SHREC dataset is composed by 14 dynamic hand gestures performed by 28 participants (all the participants were right-handed). The hand gestures were captured by the Intel RealSense short range depth camera. Each gesture was

Features	Accuracy 14 Gestures	Accuracy 28 Gestures
Proposed Method	**97.62%**	**91.43%**
Skeleton-Based Dynamic Hand Gesture Recognition [57]	88.24%	81.90%
Key Frames with Convolutional Neural Network [29]	82.90%	71.90%
Joint Angles Similarities and HOG2 for Action Recognition [48]	83.85%	76.53%
HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences [58]	78.53%	74.03%
3-D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold [59]	79.61%	62.00%

TABLE III: Comparison of the accuracy measure among significant state-of-the-art approaches on the SHREC dataset.



(a)



(b)

Fig. 11: Confusion matrices obtained on the SHREC dataset: (a) with 14 hand gestures and (b) with 28 hand gestures.

performed between 1 and 10 times by each participant in two ways: using one finger and the whole hand. Therefore, the dataset is composed by 2800 sequences. The depth image, with a resolution of $640x480$, and the coordinates of 22 hand joints (both in the 2D depth image space and in the 3D world space) were saved for each frame of each sequence in the dataset. For the proposed method, we only needed of the 3D coordinates of the joints from which we derived the features of our interest. The depth images and hand skeletons were captured at 30 frames per second (fps) and the length of the sample gestures ranges from 20 to 170 frames. Since some sequences of the dataset are very short, to avoid a sampling with a very low T value, we used the padding technique to increase the length of these sequences to an acceptable value of T (i.e., $T = 100$). As shown in Table III, the proposed method outperforms the accuracy values of the other works both in the dataset divided into 14 hand gesture classes and in the dataset divided into 28 hand gesture classes.

The confusion matrices related to the tests are shown in Fig. 11. By analysing these matrices, it can be observed that the method can accurately classify the hand gestures made by using only one finger, instead, when these gestures are made by using the whole hand, some mismatches can occur. In detail, the gesture 16 (*SWIPE LEFT*) is, sometimes, erroneously classified, while, the gesture 18 (*SWIPE UP*) can be confused with the gesture 26 (*SWIPE V*). By carefully analysing the variations of the feature values, it is noticeable that the angles obtained from these instances are similar, moreover the movements of the hand in space are not substantial. In addition, some of these sequences are composed of few frames. Despite these isolated cases, we can state that the proposed method achieves excellent performance. This result demonstrates how the DLSTM and the selected features are a very powerful solution in recognizing different types of challenging hand gestures.

V. Conclusion

In this paper, an original hand gesture recognition method based on DLSTM is presented. In particular, an affective set of discriminative features based on both joint angles and fingertip positions is used in combination with an LSTM-RNN to obtain high accuracy results. At the time of writing the first version of the manuscript, there were no other similar approaches. The method we propose outperforms competing works on the SHREC dataset. The paper also provides a new dataset, based on a large subset of the ASL, to train and test the effectiveness of approaches similar to that we present. This dataset has been also used to analyse the robustness of the extracted features and the behaviour of the network when the number of stacked LSTMs change. As a next step, we intend to create another public dataset, always based on the ASL,

in which more hand gestures are inserted. We are planning to create this new dataset also including RGB frames, depth maps, and the whole hand skeleton model. This dataset should be able to support different ambiguities study cases (e.g., the recognition of hand gestures 6 and W).

ACKNOWLEDGMENT

This work was supported in part by the MIUR under grant "Departments of Excellence 2018-2022" of the Department of Computer Science of Sapienza University.

REFERENCES

[1] A. Kendon, *Visible Action as Utterance*. Cambridge University Press, 2004.

[2] F. Quek, D. McNeill, R. Bryll, S. Duncan, X.-F. Ma, C. Kirbas, K. E. McCullough, and R. Ansari, "Multimodal human discourse: Gesture and speech," *ACM Trans. Comput.-Hum. Interact.*, vol. 9, no. 3, pp. 171–193, 2002.

[3] D.-H. Lee and K.-S. Hong, "Game interface using hand gesture recognition." International Conference on Computer Sciences and Convergence Information Technology (ICCIT), 2010, pp. 1092–1097.

[4] S. S. Rautaray and A. Agrawal, "Interaction with virtual game through hand gesture recognition." 2011 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), 2011, pp. 244–247.

[5] D. Avola, M. Spezialetti, and G. Placidi, "Design of an efficient framework for fast prototyping of customized human–computer interfaces and virtual environments for rehabilitation," *Computer Methods and Programs in Biomedicine*, vol. 110, no. 3, pp. 490 – 502, 2013.

[6] G. Placidi, D. Avola, D. Iacoviello, and L. Cinque, "Overall design and implementation of the virtual glove," *Computers in Biology and Medicine*, vol. 43, no. 11, pp. 1927–1940, 2013.

[7] T. Kim, G. Shakhnarovich, and K. Livescu, "Fingerspelling recognition with semi-markov conditional random fields." IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1521–1528.

[8] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," *IEEE Signal Processing Letters*, vol. 23, no. 9, pp. 1188–1192, 2016.

[9] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with jointly calibrated leap motion and depth sensor," *Multimedia Tools Appl.*, vol. 75, no. 22, pp. 14 991–15 015, 2016.

[10] M.-K. Sohn, S.-H. Lee, D.-J. Kim, B. Kim, and H. Kim, "A comparison of 3d hand gesture recognition using dynamic time warping." Proceedings of the Conference on Image and Vision Computing New Zealand, 2012, pp. 418–422.

[11] F. A. Barrientos and J. F. Canny, "Cursive: Controlling expressive avatar gesture using pen gesture." Proceedings of the International Conference on Collaborative Virtual Environments, 2002, pp. 113–119.

[12] A. Truong, H. Boujut, and T. Zaharia, "Laban descriptors for gesture recognition and emotional analysis," *The Visual Computer*, vol. 32, no. 1, pp. 83–98, 2016.

[13] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot." Proceedings of the ACM/IEEE International Conference on Human-robot Interaction, 2007, pp. 255–262.

[14] S. M. Goza, R. O. Ambrose, M. A. Diftler, and I. M. Spain, "Telepresence control of the nasa/darpa robonaut on a mobility platform." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2004, pp. 623–629.

[15] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2368–2377, 2014.

[16] J. S. Pierce and R. Pausch, "Comparing voodoo dolls and homer: Exploring the importance of feedback in virtual environments." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2002, pp. 105–112.

[17] M. J. Reale, S. Canavan, L. Yin, K. Hu, and T. Hung, "A multi-gesture interaction system using a 3-d iris disk model for gaze estimation and an active appearance model for 3-d hand pointing," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 474–486, 2011.

[18] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[19] H. Cheng, L. Yang, and Z. Liu, "Survey on 3d hand gesture recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1659–1673, 2016.

[20] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, pp. 2157–6904, 2013.

[21] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultiMedia*, vol. 19, pp. 4–12, 2012.

[22] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik, "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking," *Sensors*, vol. 14, no. 2, pp. 3702–3720, 2014.

[23] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1297–1304.

[24] A. Graves, A. r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 6645–6649.

[25] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent lstm neural networks for language modeling," *IEEE Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 3, pp. 517–529, Mar. 2015.

[26] A. Graves, N. Jaitly, and A. r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm." IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2013, pp. 273–278.

[27] M. Hermans and B. Schrauwen, "Training and analyzing deep recurrent neural networks." Proceedings of the International Conference on Neural Information Processing Systems, 2013, pp. 190–198.

[28] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, R. Stefan, Q. Yuan, and A. Thangali, "The american sign language lexicon video dataset." IEEE Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis (CVPR4HB), 2008, pp. 1–8.

[29] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat, "Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset." Eurographics Workshop on 3D Object Retrieval, 2017, pp. 1–7.

[30] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[31] C. Zhang and Y. Tian, "Histogram of 3d facets: A depth descriptor for human action and hand gesture recognition," *Computer Vision and Image Understanding*, vol. 139, no. Supplement C, pp. 29 – 39, 2015.

[32] C. Wang, Z. Liu, and S. C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 29–39, 2015.

[33] "Combining multiple depth-based descriptors for hand gesture recognition," *Pattern Recognition Letters*, vol. 50, no. Supplement C, pp. 101 – 111, 2014.

[34] S. Vikram, L. Li, and S. Russell, "Writing and sketching in the air, recognizing and controlling on the fly." Extended Abstracts on Human Factors in Computing Systems, 2013, pp. 1179–1184.

[35] E. Ohn-Bar and M. M. Trivedi, "The power is in your hands: 3d analysis of hand gestures in naturalistic video." IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2013, pp. 912–917.

[36] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns." Proceedings of the European Conference on Computer Vision, 2012, pp. 872–885.

[37] S. B. Wang, A. Quattoni, L. P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," vol. 2. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006, pp. 1521–1527.

[38] P. Li, H. Ling, X. Li, and C. Liao, "3d hand pose estimation using randomized decision forest with segmentation index points." IEEE International Conference on Computer Vision (ICCV), 2015, pp. 819–827.

[39] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Moddrop: adaptive multi-modal gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1692–1706, 2016.

[40] P. Suryanarayan, A. Subramanian, and D. Mandalapu, "Dynamic hand pose recognition using depth data." International Conference on Pattern Recognition (ICPR), 2010, pp. 3105–3108.

[41] H. M. Zhu and C. M. Pun, "Real-time hand gesture recognition from depth image sequences." International Conference on Computer Graphics, Imaging and Visualization (CGIV), 2012, pp. 49–52.
[42] R. Schramm, C. R. Jung, and E. R. Miranda, "Dynamic time warping for music conducting gestures evaluation," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 243–255, 2015.
[43] C. R. Naguri and R. C. Bunescu, "Recognition of dynamic hand gestures from 3d motion data using lstm and cnn architectures," in *IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 1130–1133.
[44] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1110–1118.
[45] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.
[46] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition." *Neurocomputing: Algorithms, Architectures and Applications*, pp. 227–236, 1990.
[47] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 24–38, 2014.
[48] E. Ohn-Bar and M. M. Trivedi, "Joint angles similarities and hog2 for action recognition." IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 465–470.
[49] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3d discriminative skeletal features for human action recognition." IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 471–478.
[50] A. Savitzky and M. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical chemistry*, vol. 36, pp. 1627–1639, 1964.
[51] A. M. Schaefer, S. Udluft, and H.-G. Zimmermann, "Learning long-term dependencies with recurrent neural networks," *Neurocomput.*, vol. 71, no. 13-15, pp. 2481–2488, 2008.
[52] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012.
[53] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using long short-term memory recurrent neural networks." Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH), 2014, pp. 2155–2159.
[54] F. Eyben, M. Wöllmer, B. Schuller, and A. Graves, "From speech to letters - using a novel neural network architecture for grapheme based asr." IEEE Workshop on Automatic Speech Recognition Understanding, 2009, pp. 376–380.
[55] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[56] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009.
[57] Q. D. Smedt, H. Wannous, and J. P. Vandeborre, "Skeleton-based dynamic hand gesture recognition." IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016, pp. 1206–1214.
[58] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences." IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 716–723.
[59] 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.