Dynamic Sign Language Recognition Based on Convolutional Neural Networks and Texture Maps

Edwin Escobedo
Department of Computer Science
Federal University of Ouro Preto
Ouro Preto, MG, Brazil
Email: edu.escobedo88@gmail.com

Lourdes Ramirez
Department of Computer Science
National University of Trujillo
Trujillo, Peru
Email:lo.ramirezc89@gmail.com

Guillermo Camara
Department of Computer Science
Federal University of Ouro Preto
Ouro Preto, MG, Brazil
Email: gcamarac@gmail.com

***Abstract*—Sign language recognition (*SLR*) is a very challenging task due to the complexity of learning or developing descriptors to represent its primary parameters (location, movement, and hand configuration). In this paper, we propose a robust deep learning based method for sign language recognition. Our approach represents multimodal information (RGB-D) through texture maps to describe the hand location and movement. Moreover, we introduce an intuitive method to extract a representative frame that describes the hand shape. Next, we use this information as inputs to two three-stream and two-stream CNN models to learn robust features capable of recognizing a dynamic sign. We conduct our experiments on two sign language datasets, and the comparison with state-of-the-art *SLR* methods reveal the superiority of our approach which optimally combines texture maps and hand shape for SLR tasks.**

I. Introduction

Sign Language (*SL*) is a visual-gestural language used by hearing-impaired people. This language uses hand-shapes variations, body movement, and even facial expression to convey information or meaning. Also, sign languages provide a natural way of interaction by minimizing the barrier of communication between hearing-impaired and society. Currently, many translation services are human-based and are expensive due to the experienced staff required. For this reason, many researchers are interested on the development of different Sign Language Recognition (SLR) applications. (*SLR*).

Sign languages are not simple holistic gestures; they are well structure linguistic systems which decomposes into small units such as hand configuration, location, and movement [1], [2]. The *SLs* structure consists of primary and secondary parameters that are combined sequentially or simultaneously. According to [3], the primary parameters are:

- Hand Configuration: hands take different shapes to generate signs.
- Articulation Point or Location: is the space in front of the body (neutral space) or a region of the own body (head, waist, and shoulders), where signs are articulated.
- Movement: is a complex parameter that involves many forms and directions, *i.e.*, from pulse motion, movements of the finger joints, directional movements in the space or a set of movements in the same sign.

The secondary parameters are:

- Orientation of the palm of hands: is the direction of the palm during the sign: facing up, down, toward the body, forward, left, or right.
- Facial Expressions: many signs have a distinctive element such as the facial or body expression, giving more sense and feeling to the statement. So, they can express the difference between affirmative, interrogative, exclamatory, and negative sentences.

These primary and secondary parameters are used to complement each other. Depending on the expression context, some parameter may not be required to interpret a sign. Moreover, an individual parameter modifies the meaning of a sign, *e.g.*, the location of the hand in a different position of the body.

In recent years, the success of automatic Sign Language Recognition (*SLR*) systems has opened up a new way of Human-Computer-Interaction (HCI) that can convert sign gestures into text/speech [4]–[6]. Nevertheless, sign language recognition is still a very challenging task due to the complexity of exploiting the information from primary and secondary parameters. The challenge in sign language recognition is the learning or development of descriptors to represent hand configuration and movement; *e.g.*, hand configuration involves tracking hand regions in a video stream, segmenting hands, deleting blur images, etc. Initial approaches, proposed by different researchers, used RGB cameras to create controlled datasets to facilitate the tracking and segmentation of hands. The authors applied hand-crafted descriptors to extract features to describe the movement and hand shape. Next, a temporary statistical method that aligns signs and computes a likelihood of similarity was used to recognize signs. The most used methods were Dynamic Time Warping (DTW) [7], [8] and Hidden Markov Models (HMMs) [9]–[12].

Due to the introduction of low-cost depth cameras/sensors such as Microsoft Kinect [13] or Leap Motion [14], the sign language recognition paradigm has turned from RGB camera-based technique to an RGB-D sensor-based 3D environment and has opened up ways to extract and learn new features from multimodal input data. These devices/sensors are capable of delivering the 3D skeleton of the whole body and are not sensitive to illumination variations and cluttered backgrounds. Moreover, they present a signicant contribution on the robustness of *SLR* systems.

Different approaches proposed to convert the 3D skeleton trajectories into spherical coordinates to describe spatial and temporal information of signs [15]–[17]. Other methods as the proposed by Hernandez et al. [18] used the bag of visual word technique and Dynamic Time Warping. In Budiman et al. [19], the authors presented an On-line Sequential Extreme Machine Learning (*OS-ELM*) using upper body joints to compute three projected angles to each axis (x, y, z) and a K-means algorithm to generate hand features.

Currently, recent studies have demonstrated the power of deep convolutional neural networks (*ConvNets*), and it has become an effective strategy for extracting high-level features of data [20]–[22]. Moreover, many new architectures have been proposed for *SLRs* systems such as 3DCNN [23], or RNN [24] to process dynamic signs.

Other approaches proposed the generation of texture color maps to represent the 3D skeleton trajectory [25]–[29]. Following the same ideas, in [30]–[32], are proposed methods to encode video sequences into movement maps [30]–[32]. These approaches were presented to reduce the complexity of the CNN model used to process video sequences.

Thus, based on previous ideas, we propose a Sign Language Recognition System combining CNN models with texture maps or dynamic images. Our goal is to efciently summarize the primary parameters of a sign sequence in five texture images, especially the location and movement of the hand. We generate three dynamic images using the skeleton data (DXY, DXZ, DYZ), one using the color data (DC) and one using depth data (DD). We also propose a method to extract the most representative frame that describes the hand configuration. Finally, we present a new dataset of sign language with complete RGB-D and skeleton information with correctly labeled signs.

The remainder of this paper is organized as follows. In Sec. II, we describe our proposed approach. Experiments and results are presented in Sec. III. In Sec. IV, we discuss the conclusions and future works.

II. Method Overview

The proposed method is shown in Fig. 1 and consists of a set of phases explained in the following subsections.

A. Texture Maps Generation

We combine two main concepts to generate texture color maps: skeleton optical spectra and rank pooling. Our goal is to summarize a dynamic sign efficiently into single flow images to represent global and local information of hand movement.

1) Skeleton Optical Spectra: Similar to Hou et al. [27], we use the HSB color model to generate Skeleton Optical Spectra (SOS) images. These texture maps are capable of describing the hand movement and its location regarding the body (global movement). To further enhance the encoded spatiotemporal information, we encode the velocity of the joints into the saturation (S) and brightness (B) of the SOS images.

Let $S_l = \{s^1, s^2, \dots, s^n\}$ the skeleton sequence of a sign S_l, where n is the number of frames and $s^i = p_1^i, p_2^i, \dots, p_m^i$ where p_k^i is the coordinates of the joint k in the ith frame, i.e. $p_k^i = (x_k^i, y_k^i, z_k^i)$. We projected the skeleton joints of a sign video on three orthogonal Cartesian planes: *XY, YZ*, and *XZ*. Hence, we generate three texture color maps (*DXY, DYZ*, and *DXZ*). Fig. 2a shows an example of the texture color maps generated for a particular sign.

Firstly, we consider the relevant movement of each body part. Arms and legs often have more motion information, but different frequency. Therefore, it is not recommended to group these body parts in the same spectra. Compared to the method proposed in [27], we generate five spectral distributions (H) to encode five different body parts with its respective joints: left leg part K_1 = {*left hip, left knee, left ankle, left foot*}, right leg part K_2 = {*right hip, right knee, right ankle, right foot*}, left arm part K_3 = {*left shoulder, left elbow, left wrist, left hand*}, right arm part K_4 = {*right shoulder, right elbow, right wrist, right hand*}, and middle body part K_5 = {*head, neck, torso, hip center*}.

The spectrum, *i.e.* the range of hue (H) in Eq. 1, of the right arm part, is the inverted spectrum assigned to the left arm part. The range of hue of the right leg part is the inverted spectrum assigned to the left leg part. In the middle body part, we adopt a grayscale (we assign $hue = 0$) because of the subtle motion of these joints [27].

In general, the encoding and enhancement of hue (H), saturation (S), and brightness (B) can be expressed as follows:

$$H(j,i) = \begin{cases} \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2} + \frac{h_{\min}}{2}, & j \in K_1 \\ \frac{h_{\max}}{2} - \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2}, & j \in K_2 \\ h_{\max} - \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2}, & j \in K_3 \\ \frac{h_{\max}}{2} + \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2}, & j \in K_4 \\ 0, & j \in K_5 \end{cases}$$

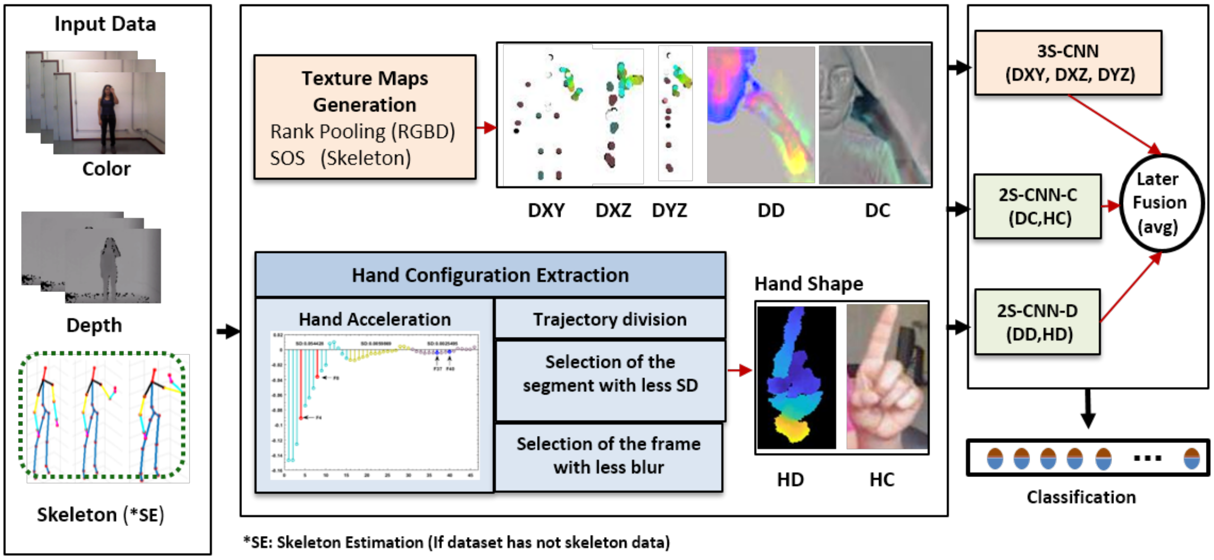
$$S(j,i) = \begin{cases} \frac{v_j^i}{\max\{v\}} \times (s_{\max} - s_{\min}) + s_{\min}, & j \in K_{1:4} \\ 0, & j \in K_5 \end{cases}$$

$$B(j,i) = \begin{cases} \frac{v_j^i}{\max\{v\}} \times (b_{\max} - b_{\min}) + b_{\min}, & j \in K_{1:4} \\ b_{\max} - \frac{i}{n} \times (b_{\max} - b_{\min}), & j \in K_5 \end{cases} \tag{1}$$

where the joint velocity is calculated by

$$v_j^i = \left\| p_j^{i+1} - p_j^i \right\|_2. \tag{2}$$

Thus, for each joint p_j^i in the skeleton s^i of an sign S_l, we apply the Eq. 1 to compute its respective hue, saturation and



*SE: Skeleton Estimation (If dataset has not skeleton data)

Fig. 1. Overview of the proposed Sign Language Recognition system.

brightness values. Next, these values are plotted in each Cartesian plane to generate three dynamic SOS images. Finally, we convert the SOS images to RGB color model to process them in the training stage.

If the skeleton data is not available, we estimate the pose using the approach based on a nonparametric representation that encodes both the position and the orientation of human limbs proposed by Cao *et al.* [33]. In this case, we only compute the *DXY image*.

2) Rank Pooling: For RGB-D data, we apply the dynamic images generation based on rank pooling proposed in [31], [32]. The core idea is to represent a sign video through a single image that summarizes the hand's movement in the region where the sign is articulated (local movement). Let a sign video S_l, we can represent S_l as a ranking function for its frames $F_1, \ldots, F_T$. In more detail, let $\psi_{F_t} \in \mathrm{R}^d$ be a representation or feature vector extracted from each individual frame F_t in the video. Let $V_i = \frac{1}{i}\sum_{\tau=1}^{i}\psi(F_\tau)$ be time average of these features up to time t [32]. The ranking function associates each time t a score $S(t|\mathbf{d}) = \langle \mathbf{d}, V_t \rangle$, where $\mathbf{d} \in \mathbb{R}^d$ is a vector of parameters. The function parameters $\mathbf{d}$ are learned, so that the scores reflect the rank of the frames in the video. Therefore, later times are associated with larger scores, *i.e.* $q > t \implies S(q|\mathbf{d}) > S(t|\mathbf{d})$. Learning $\mathbf{d}$ is posed as a convex optimization problem using the *RankSVM* [34] formulation:

$$\mathbf{d}^* = \rho(F_1, \ldots, F_T; \psi) \tag{3}$$

Bilen et al. [31] presented an approximation to rank pooling which is faster and works well in practice, they derivated the approximate rank pooling by considering the first step in a gradient-based optimization reducing the Eq. 3 to:

$$\hat{\rho}(F_1,\ \ldots,\ F_T; \psi) = \sum_{t=1}^{T} \alpha_t \psi(F_t). \tag{4}$$

The coefficients α_t are given by:

$$\alpha_t = 2(T - t + 1) - (T + 1)(H_T - H_{t-1}). \tag{5}$$

where $H_t = \sum_{i=1}^{t} 1/t$ is the $t - th$ Harmonic number and $H_0 = 0$. Likewise, we can use individual video frames F_t directly by replacing $\psi(F_t)$.

Hence, the authors show that d^* can be interpreted as a standard RGB image, since, this image is obtained from rank pooling the video frames which summarizes information from the whole video sequence. The complete process is explained and detailed in [31], [32].

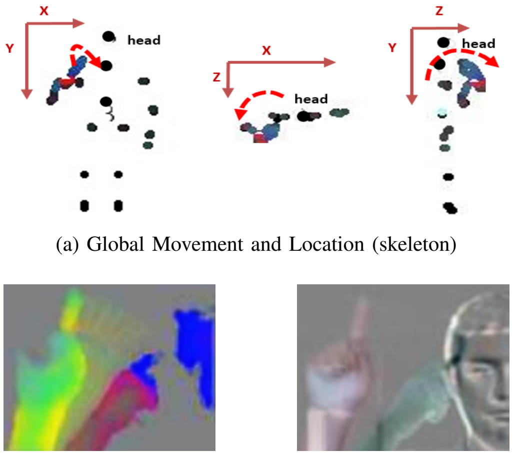
To generate texture images from depth data, we normalized each video frame F_d to the interval $[0 : 255]$ using the Min-Max normalization defined by $F_d = \frac{F_d - min(F_d)}{max(F_d) - min(F_d)} \times 255$.

Finally, we generate two dynamic images from RGB-D videos. Fig. 2b shows the dynamic depth image (DD) computed by the normalized depth video. Similarly, the Fig. 2c shows the dynamic color image (DC) computed by the RGB video. Notice that dynamic images tend to focus mainly on the active body part, such as the right hand in Fig. 2b. In contrast, background pixels and background motion patterns tend to average out. Hence, the pixels in dynamic images seem to focus on the appearance and motion of the user body, which indicates that they can contain the necessary information to recognize a sign.

B. Hand Configuration Extraction

Despite texture images generated from rank pooling describe the local hand movements, they miss information of the hand shape due to the short time duration and fast motion of a sign. Therefore, we add a process to extract the hand configuration of a sign. We follow these three basic ideas:

- Due to the short time duration of a sign, the hand moves at different speeds, showing variations in its acceleration.



(a) Global Movement and Location (skeleton)

(b) Local Hand Movement (Depth)

(c) Local Hand Movement (RGB)

Fig. 2. Sample of texture Images that represent global and local information of the hand movement.

- There are segments in the hand trajectory where the acceleration performs high variations. In these segments, the corresponding frames present a high level of blur.
- There are segments in the hand trajectory where the variation of the acceleration is smallest. In these segments, the frames present a low level of blur. Therefore, in one point of this segment, there is a frame that clearly shows the corresponding hand configuration of the sign performed.

Based on the ideas presented above, we propose a fast and effective method to extract the most representative frame with the hand shape belonging a sign.

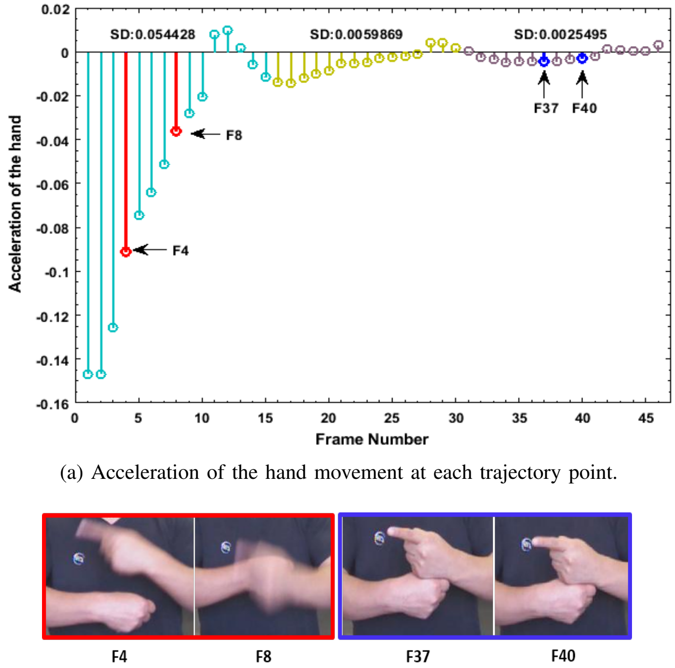
1) Let $p_h^i = (x_h^i, y_h^i, z_h^i)$ be the coordinates of the hand joint in the *ith* frame, where $i \in \{1, \ldots, n\}$ and n is the number of points. We compute the distances between each consecutive point: $dist^i = \sqrt{(p_h^{i+1} - p_h^i)^2}$, for $i = 1, 2, \ldots, n-1$.
2) Next, we compute the velocity V^i for each point, and extract the difference of the velocities to compute the vector A of accelerations:

$$dist_{cum}^i = \sum_{k=1}^{i} dist^k, \text{ for } i = 1, 2, \ldots, n-1 \quad (6)$$

$$V^i = dist_{cum}^i \times \frac{1}{i}, \text{ for } i = 1, 2, \ldots, n-1 \quad (7)$$

$$A = V_h^{i+1} - V_h^i, \text{ for } i = 1, 2, \ldots, n-1 \quad (8)$$

3) Finally, we divide the hand trajectory into M segments S^k, $k = 1 : M$, for each S^k segment, we calculate the corresponding standard deviation (*SD*) of its accelerations. Then, we select the S_{min}^k segment with the minimum value of *SD*.
4) For each frame corresponding to the S_{min}^k segment, we extract the hand area and calculate its relative degree of focus using the energy of Laplacian as the measure algorithm [35]. So, we select the frame with the maximum degree of focus.



(a) Acceleration of the hand movement at each trajectory point.

(b) Frames corresponding to four points in the trajectory.

Fig. 3. Illustration of our method to extract the hand shape inside a video sequence.

Fig. 3a shows the acceleration variations generated by the hand movement. The trajectory was divided into three segments. In Fig. 3b the frames F4 and F8 corresponding to the first segment with greater *SD* (0.054428) are displayed; the frames F37 and F40 corresponding to the segment with minimum *SD* (0.0025495) are also presented.

C. CNN training and fusion

Different approaches with two-stream (or N-stream) CNNs propose to combine the spatial and temporal information into a single network using an integration process [36], [37]. This integration process is known as fusion level and can be performed after any convolutional stage or fully-connected layer. The information fusion is important to recognize a sign language since a sign is composed of different parameters (hand configuration, location, and movement). Therefore, our strategy consists of grouping the spatiotemporal information extracted from each multimodal channel. Thus, we create three groups: *a)* global movement and location (DXY, DXZ, and DYZ), *b)* local movement and hand shape for RGB data (DC,HC) and *c)* local movement and hand shape for depth data (DD,HD).

We adopted the *imagenet-vgg-f* [38] model to design our proposed CNN architectures. This pre-trained model is composed of five convolutional stages and three fully-connected layers of size 4096. Also, this model takes 224×224×3 images size as input. According to the number of input images of each group, we create a copy of the first four convolutional stages using as initial training parameters the filters learned from the *imagenet-vgg-f* model. Next, we introduce a fusion layer to combine the feature maps calculated after the *Conv4* stage. Finally, we add three fully-connected layers and fix the output to the number of classes. For the first group, we create a 3S-CNN model with DXY, DXZ, and DYZ as inputs (Fig. 4a). For group *b*, we create the 2S-CNN-C model with DC and HC as inputs (Fig.4b). Similarly, we create the 2S-CNN-D model for the group *c* with DD and HD as inputs.

To fuse data, we did the following *Conv Fusion* [36]: given k feature maps $x_r^n, r = 1:k$, as outputs of the *n-th* layer for each stream in a CNN architecture at the same pixel location $(i;j)$ and the same channel d. First, the x_i^n feature maps are concatenated at the same spatial location $(i;j)$ across channel d:

$$\boldsymbol{y}^{n,cat}(i,j,kd) = cat(x_r^n(i,j,d)) \qquad (9)$$

where $\boldsymbol{y}^{n,cat} \in R^{H'\times W'\times D'}, D' = kd, 1 \le i \le H', 1 \le j \le W', 1 \le d \le D'$.

Then, the stacked up feature map is convoluted with a bank of filters $\mathcal{F} \in R^{1\times 1\times D'\times D''}$, as follows:

$$\boldsymbol{y}^{n,conv} = \boldsymbol{y}^{n,cat} \; * \; \mathcal{F} + b, \qquad (10)$$

where $b \in R^{D''}$ is a bias term. The filters are used to learn weighted combinations of feature maps x_r^n.

Since the three architectures return a score vector of classification ($\boldsymbol{s}_{2SC}$, $\boldsymbol{s}_{2SD}$, $\boldsymbol{s}_{3S}$), we apply a later fusion to calculate the average value of the three vectors to obtain the final classification score $\boldsymbol{s}^{avg}$, which means the highest score represents the recognized sign.

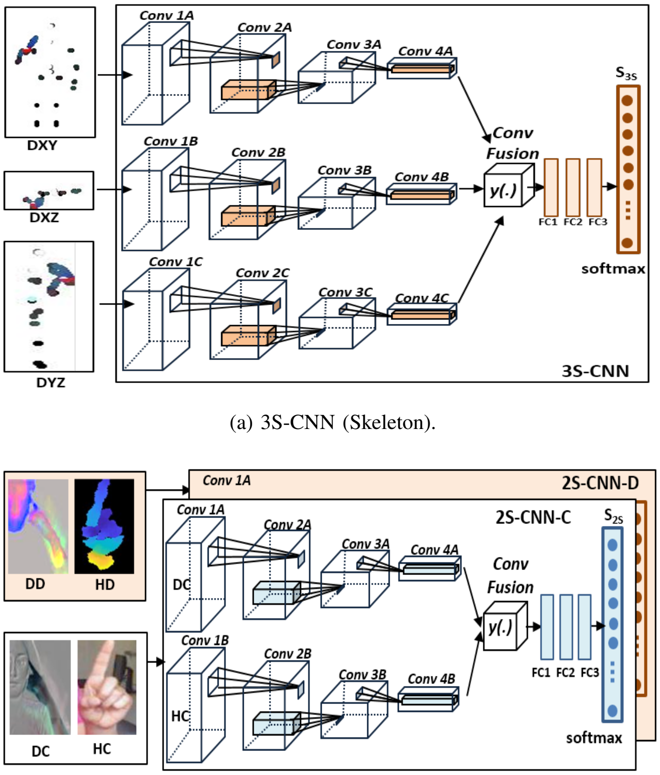
$$\boldsymbol{s}^{avg} = avg(\boldsymbol{s}_{2SC}, \boldsymbol{s}_{2SD}, \boldsymbol{s}_{3S}) \qquad (11)$$

III. Experimental Results

A. Datasets and Evaluation Protocol

1) The LSA64 dataset [39]: is a dictionary of Argentinian Sign Language; which includes 3200 videos; 10 subjects executed 5 repetitions of 64 different types of signs. Also, to simplify the problem of hand segmentation within an image, subjects wore fluorescent-colored gloves. These simplify the problem of recognizing and segmenting the position of the hand, also remove all issues associated with skin color variations; at the same time, preserve the difficulty of recognizing the hand shape. Each sign was executed by imposing few constraints on subjects to increase the diversity and realism in the database. All subjects were non-signers and right-handed; they learned how to perform signs during the recording session. Moreover, they watched a video of the signs at the same time performed by one of the authors; finally, they practiced each sign a few minutes before recording.

To conduct Experiments, the authors performed a subject-dependent classification by dividing the dataset randomly through 5 experiments using 80% of training and 20% of validation. These result in 2560 videos for training and 640 videos for testing. Each set is created randomly to seek to exempt the random factors present in the tests.



(a) 3S-CNN (Skeleton).

(b) 2S-CNN (RGB-D).

Fig. 4. The proposed CNN architectures for our Sign Language Recognition system.

2) The LIBRAS-BSL dataset: Many of the existing public sign language datasets lack effective and accurate labeling or are stored in a single data format [4], [5]. Thus, we present a public labeled Brazilian Sign Language dataset composed of 37 very similar signs to train deep neural networks for dynamic sign language recognition task. The dataset was performed by ten subjects: six women and four men. Each participant executed 12 repetitions of a sign. The dataset contains the complete RGB-D and skeleton data collected using a Kinect device. To facilitate a fair comparison with the state-of-the-art, we split the dataset into three subsets: training, validation, and testing, as listed in Table I.

TABLE I
Information of the LIBRAS–BSL Dataset

Sets	RGB	Depth	Skeleton	Sub. (ID)
Training	3552	3552	3552	8 (2, 4–10)
Validation	444	444	444	1 (1)
Testing	444	444	444	1 (3)

B. Training Details

1) Experimental environment: we conduct our experiments on a Notebook with Intel Core i7-6500U inside, CPU @ 3.16GHz, 12GB RAM and NVIDIA Geforce GTX 950M GPU. We train all networks using the Matconvnet toolbox [40].

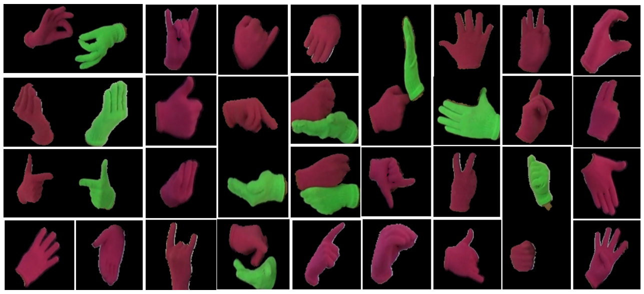


Fig. 5. Hand configurations detected on the LSA64 dataset.

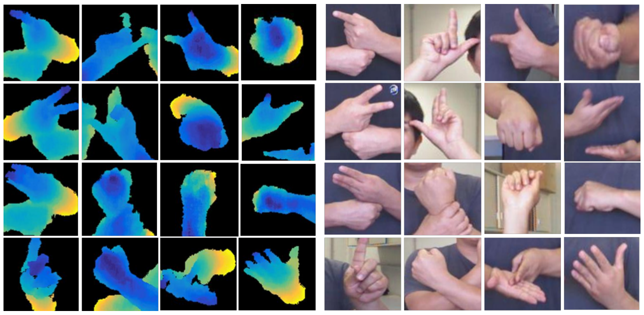


Fig. 6. Hand configurations detected on the LIBRAS-BSL dataset.

2) Parameter setting: we define the number of segments to divide a sign as $M = 3$ due to the short duration of a sign. We resize the input images to 224×224 for the CNN models. The training process stopped after 60 epochs. The learning rate starts from 0.01 in the first 20 epochs, 0.001 in other 20 epochs, and 0.0001 in the last 20 epochs. We set the dropout ratio of the fully-connected layers to 0.6. We use a batch size of 205 for both 2S-CNN models and 130 for the 3S-CNN model.

C. Results on the LSA64 dataset

In spite of the dataset does not provide depth and skeleton information, we select it owing to the special properties of deep learning. The LSA64 dataset is a large and balanced sign language dataset with several frames per video sequence. Also, it has a high number of similar signs (in motion and hand configuration).

Therefore, we apply the method mentioned at final of Sec. II-A1 to estimate body points [33]. Next, we apply our approach to detect the hand configuration. Fig. 5 shows some hand configurations detected for the LSA64 dataset.

Then, we conduct the follow experiments:

- We conduct a subject-dependent classification by dividing the dataset randomly through 5 experiments using 80% of training and 20% of validation, similar to the original experimental protocol.
- As we only generated the DXY texture map. We perform the training of the 3S-CNN model using the DXY image as input for the three streams channels.
- We generate the DC and HC textures images to perform the training of the 2S-CNN-C model.
- We perform a later fusion with the scores of the 3S-CNN and 2S-CNN-C models.

Table II summarizes the mean accuracy and mean standard deviation computed in five experiments. We also report the state-of-the-art results and the baseline results from [41], which employs different classifiers like HMM with Gaussian Mixture Models (ALL-HMM) to improve the initial results. Also, in Table II we note that the 3S-CNN model overcomes several proposed methods, which means that analyzing only the location and movement of the hand (the global movement), we can discriminate the majority of signs on LSA64 dataset (96.92 %). However, we still need local information, i.e., the hand configuration to improve the results. For this reason, the 2S-CNN model achieves a mean accuracy of 99.82%, since it has information of the hand configuration (HC) and local information from its movement (DC). The later fusion uses the mean operator and improves the final prediction stabilizing the predictions of the CNNs. We can observe this in the standard deviation that decreases to 0.33 in the later fusion.

Likewise, most of the works in the the-state-of-the-art use recurrent models or complex CNN architectures to compute spatiotemporal information of a sign. In our case, we use texture maps to encode the movement and location of the hand. As we see in Table II, texture maps help us to achieve satisfactory results on LSA64 dataset using a simple CNN model. However, there are some things to take into account in the methods that generate texture maps, *e.g.*, the overlapping between the same joints over different frames due to their slow motion, or the high level of noise in the skeleton that may generate a false hand movement.

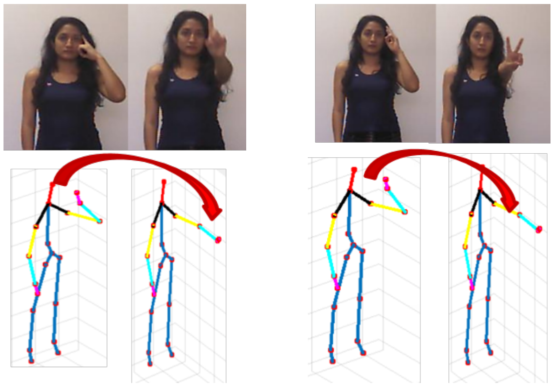
TABLE II
COMPARATIVE RESULTS OF THE STATE-OF-THE-ART ON LSA64 DATASET.

Method	**Accuracy (mean ± std)**
ProbSOM [42]	91.70
3DCNN [23]	93.90 ± 1.40
ALL-BF-SVM [41]	95.08 ± 0.69
ALL (sequence agnostic) [41]	97.44 ± 0.59
ALL-HMM [41]	95.92 ± 0.95
Deep Network [43]	98.09 ± 0.59
skeleton + LSTMs [24]	99.84 ± 0.19
3S-CNN	96.92 ± 0.56
2S-CNN-C	99.82 ± 0.48
Later Fusion (3S + 2SC)	99.91 ± 0.33

D. Results on the LIBRAS–BSL dataset

This dataset provides complete RGB-D and skeleton information. Although our proposed dataset has fewer signals (37), one of the advantages is the high degree of similarity presented in its primary parameters (location, and hand configuration).

Similar to LSA64 dataset, we apply our approach to detect the hand configuration. Fig. 6 shows some hand configurations detected on LIBRAS-BSL dataset.



(a) One day

(b) two days

Fig. 7. Sample of two signs with the same location and hand movement but different hand configuration.

Table III summarizes the results after applying different experimental schemes on the LIBRAS-BSL dataset. When we analyze the results, we assume that at the individual level, the 2S-CNN-D model is the most descriptive because depth data are invariant to light variations. Thus, this model achieves a result greater than 81% in the test data, when combining the local information of hand movement (DD) with its configuration (HD). Otherwise, the 3S-CNN model correctly describes globally the position and movement of the body. Since this model does not present the hand shape information, its performance decreases due to the confusion with other signs with similar movements as in Fig. 7, where two signs present two similar primary parameters (location and movement).

Also, we can observe that the combination of the 3S-CNN model with the 2S-CNN-C and 2S-CNN-D models significantly increases the result of our proposed method (87.02 %) due to, the fact that in the end, we are integrating our CNN models trained with the information of the primary parameters of a sign language (location, movement and hand configuration).

Finally, in Fig. 8, we present the final confusion matrix obtained with the test data. When we analyze the confusion matrix, there is still a confusion between certain groups of signs, *e.g.*, one year sign (ID: 1), two years sign (ID: 2) and three years sign (ID: 3); they have high levels of confusion. As LIBRAS-BSL dataset contains fewer signs than LSA64 dataset, presents a high degree of complexity because it contains similar signs differentiated only by a primary parameter.

IV. Conclusion

In this paper, we propose a robust deep learning based method for sign language recognition. Our approach encodes the location and movement of the hands in texture maps. Then, we extract a representative frame that describes the hand configuration in a sign video. Next, we use this information as inputs to two three-stream and two-stream CNN models to learn robust features capable of recognizing a sign. We conduct our experiments on two SLR datasets. Results showed that our proposed method outperform state-of-the-art methods and reveal the superiority of our approach which optimally combines texture maps and hand shape for SLR tasks. Also, we have presented a challenging dataset of Brazilian Sign Language. As future work, we will explore new fusion schemes of 2S-CNN and 3S-CNN models to improve the performance of our proposed model and try to decrease the error between very similar signs.

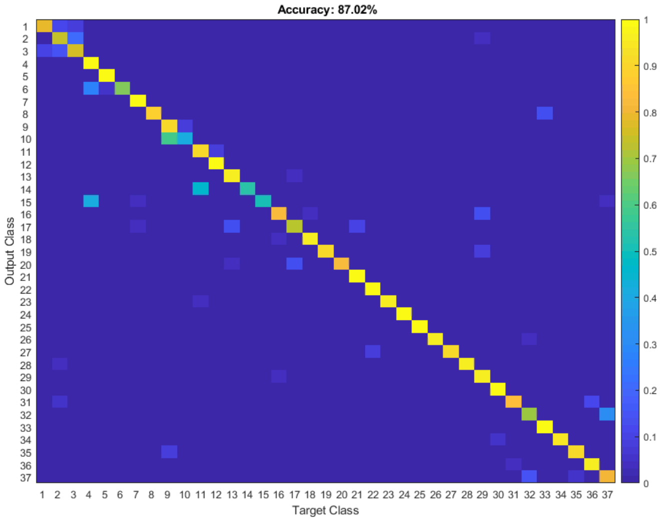


Fig. 8. Confusion Matrix obtained in the testing set on the LIBRAS-BSL dataset.

TABLE III
Results of the experimental Schemes on the LIBRAS–BSL dataset

Method	Validation	Testing
3S-CNN	79.19	79.25
2S-CNN-C	81.18	82.94
2S-CNN-D	81.19	83.20
Later Fusion (3S, 2SC)	82.86	84.10
Later Fusion (3S, 2SD)	83.75	84.68
Later Fusion (2SC, 2SD)	84.25	85.86
Later Fusion (3S, 2SC, 2SD)	85.25	87.02

Acknowledgment

The authors thank the Graduate Program in Computer Science (PPGCC) at the Federal University of Ouro Preto (UFOP), the Coordination for the Improvement of Higher Level Personneland (CAPES) and the funding Brazilian agency (CNPq).

References

[1] W. C. Stokoe Jr, "Sign language structure: An outline of the visual communication systems of the american deaf," *Journal of deaf studies and deaf education*, vol. 10, no. 1, pp. 3–37, 2005.

[2] J. F. Lichtenauer, E. A. Hendriks, and M. J. Reinders, "Sign language recognition by combining statistical dtw and independent classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2040–2046, 2008.

[3] L. F. Brito, *Por uma gramática de línguas de sinais*. Tempo Brasileiro, 1995.

[4] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "Coupled hmm-based multi-sensor data fusion for sign language recognition," *Pattern Recognition Letters*, vol. 86, pp. 1–8, 2017.

[5] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3d convolutional neural networks," in *2015 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2015, pp. 1–6.

[6] W. Ahmed, K. Chanda, and S. Mitra, "Vision based hand gesture recognition using dynamic time warping for indian sign language," in *2016 International Conference on Information Science (ICIS)*. IEEE, 2016, pp. 120–125.

[7] W. Gao, G. Fang, D. Zhao, and Y. Chen, "Transition movement models for large vocabulary continuous sign language recognition," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.* IEEE, 2004, pp. 553–558.

[8] R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 462–477, 2009.

[9] C. Wang, W. Gao, and S. Shan, "An approach based on phonemes to large vocabulary chinese sign language recognition," in *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*. IEEE, 2002, pp. 411–416.

[10] R.-H. Liang and M. Ouhyoung, "A real-time continuous gesture recognition system for sign language," in *Proceedings third IEEE international conference on automatic face and gesture recognition*. IEEE, 1998, pp. 558–567.

[11] C. Vogler and D. Metaxas, "Handshapes and movements: Multiple-channel american sign language recognition," in *International Gesture Workshop*. Springer, 2003, pp. 247–258.

[12] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden markov models," in *Motion-Based Recognition*. Springer, 1997, pp. 227–243.

[13] Z. Zhang, "Microsoft kinect sensor and its effect," *MultiMedia, IEEE*, vol. 19, no. 2, pp. 4–10, 2012.

[14] 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.

[15] L. Geng, X. Ma, B. Xue, H. Wu, J. Gu, and Y. Li, "Combining features for chinese sign language recognition with kinect," in *Control & Automation (ICCA), 11th IEEE International Conference on*. IEEE, 2014, pp. 1393–1398.

[16] E. Escobedo-Cardenas and G. Camara-Chavez, "A robust gesture recognition using hand local data and skeleton trajectory," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1240–1244.

[17] Y. Zhao, Y. Liu, M. Dong, and S. Bi, "Multi-feature gesture recognition based on kinect," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2016 IEEE International Conference on*. IEEE, 2016, pp. 392–396.

[18] A. Hernández-Vela, M. Á. Bautista, X. Perez-Sala, V. Ponce-López, S. Escalera, X. Baró, O. Pujol, and C. Angulo, "Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d," *Pattern Recognition Letters*, 2013.

[19] A. Budiman, M. I. Fanany, and C. Basaruddin, "Constructive, robust and adaptive os-elm in human action recognition," in *Industrial Automation, Information and Communications Technology (IAICT), 2014 International Conference on*. IEEE, 2014, pp. 39–45.

[20] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[21] L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *International Journal of Computer Vision*, pp. 1–10, 2015.

[22] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, X. Cao, Z. Liu, X. Chai, Z. Liu *et al.*, "Multimodal gesture recognition based on the resc3d network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3047–3055.

[23] G. M. R. Neto, G. B. Junior, J. D. S. de Almeida, and A. C. de Paiva, "Sign language recognition based on 3d convolutional neural networks," in *International Conference Image Analysis and Recognition*. Springer, 2018, pp. 399–407.

[24] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "A deep learning approach for analyzing video and skeletal features in sign language recognition," in *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*. IEEE, 2018, pp. 1–6.

[25] E. K. Kumar, P. Kishore, A. Sastry, M. T. K. Kumar, and D. A. Kumar, "Training cnns for 3-d sign language recognition with color texture coded joint angular displacement maps," *IEEE Signal Processing Letters*, vol. 25, no. 5, pp. 645–649, 2018.

[26] P. Wang, Z. Li, Y. Hou, and W. Li, "Action recognition based on joint trajectory maps using convolutional neural networks," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 102–106.

[27] Y. Hou, Z. Li, P. Wang, and W. Li, "Skeleton optical spectra based action recognition using convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[28] Z. Ding, P. Wang, P. O. Ogunbona, and W. Li, "Investigation of different skeleton features for cnn-based 3d action recognition," in *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*. IEEE, 2017, pp. 617–622.

[29] C. Li, Y. Hou, P. Wang, and W. Li, "Joint distance maps based action recognition with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 624–628, 2017.

[30] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars, "Rank pooling for action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 773–787, 2017.

[31] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[32] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3034–3042.

[33] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," 2017.

[34] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[35] S. Murali, T.-S. Choi, and A. Nikzad, "Focusing techniques," *Applications in Optical Science and Engineering. International Society for Optics and Photonics*, 1992.

[36] J. Chen, J. Wu, J. Konrad, and P. Ishwar, "Semi-coupled two-stream fusion convnets for action recognition at extremely low resolutions," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 139–147.

[37] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[38] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014, pp. 1–12.

[39] F. Ronchetti, F. Quiroga, C. Estrebou, L. Lanzarini, and A. Rosete, "Lsa64: A dataset of argentinian sign language," *XX II Congreso Argentino de Ciencias de la Computacin (CACIC)*, 2016.

[40] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[41] F. Ronchetti, F. Quiroga, C. Estrebou, L. Lanzarini, and A. Rosete, "Sign languague recognition without frame-sequencing constraints: A proof of concept on the argentinian sign language," in *Ibero-American Conference on Artificial Intelligence*. Springer, 2016, pp. 338–349.

[42] F. Ronchetti, "Reconocimiento de gestos dinámicos y su aplicación al lenguaje de señas," in *XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste).*, 2018.

[43] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "Sign language recognition based on hand and body skeletal data," in *2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. IEEE, 2018, pp. 1–4.