Evaluating deep models for dynamic brazilian sign language recognition

Lucas Amaral[1], Givanildo L. N. Júnior[1], Tiago Vieira[1], and Thales Vieira[2](✉)

[1] Institute of Computing, Federal University of Alagoas, Maceió, Brazil
[2] Institute of Mathematics, Federal University of Alagoas, Maceió, Brazil
thales@pos.mat.ufal.br

Abstract. We propose and investigate the use of deep models for dynamic gesture recognition, focusing on the recognition of dynamic signs of the Brazilian Sign Language (Libras) from depth data. We evaluate variants and combinations of convolutional and recurrent neural networks, including LRCNs and 3D CNNs models. Experiments were performed with a novel depth dataset composed of dynamic signs representing letters of the alphabet and common words in Libras. An evaluation of the proposed models reveals that the best performing deep model achieves over 99% accuracy, and greatly outperforms a baseline method.

Keywords: Brazilian sign language · Dynamic sign language recognition · Deep learning.

1 Introduction

Sign languages are visual languages used by deaf communities to communicate. Differently from spoken languages, in which grammar is expressed through sound, sign languages employ hand gestures, movement, orientation of the fingers, arms or body, and facial expressions [2, 5]. However, sign languages are still very limited as a social inclusion tool, mainly because most hearing individuals are not knowledgeable in any sign language. Consequently, most deaf people can only communicate with a very limited part of the population, finding difficulties to interact in many daily activities.

In the last decade, with improvements on hardware and computer vision algorithms, real-time recognition of sign languages is becoming feasible. These novel technologies can enable the communication between deaf and hearing people and also potentialize interactions with machines through gestures. While recognizing simple static signs is an easy and well-established problem for many machine learning algorithms [4, 12, 1], recognizing words represented by complex dynamic interactions between hands, arms, body and facial expressions is rather a difficult task. An intermediate step to achieve a complete solution is developing robust methods to recognize dynamic signs, *i.e.* hand poses in movement.

However, very few studies have aimed to recognize dynamic signs of Libras [12, 3, 11], and none of them investigated deep models such as recurrent

neural networks. Pizzolato *et al.* [12] presented a two-layer ANN used for automatic finger spelling recognition of Libras signs from RGB images. However, the architecture of the network is a Multi Layer Perceptron (MLP), which is currently an outdated choice for image classification or dynamic gestures recognition. Moreira *et al.* [11] use depth data to recognize all letters of the Libras alphabet, including static and dynamic signs. They also employ neural networks, but as in the previous, only a general MLP is proposed, requiring the optimization of a huge number of weights. Recently, Cardenas and Camara-Chavez [3] proposed a method that fuses deep learning descriptors to recognize dynamic gestures, including experiments on Libras. Depth, RGB and skeleton data are fused in that hybrid approach. Deep features from images are extracted using straightforward 2D CNNs applied to both depth data and image flow. However, according to Tran *et al.* [14], 2D CNNs are not suitable for spatiotemporal feature learning as recent state-of-the-art architectures, which we investigate in this work. It is also worth mentioning that the authors of that work suggested the use of 3D CNNs as future work.

In this paper, we propose to investigate deep models to recognize dynamic signs of the Brazilian Sign Language (Libras) from depth data. To the best of our knowledge, deep models based on learning spatiotemporal features have not been investigated to recognize dynamic signs of Libras. More specifically, we investigate the use of variants of 3D CNNs, which are capable of learning local spatiotemporal features [9]; and LRCNs, which can learn long-term spatial features dependencies [7]. We present experiments to evaluate the ability of variants of both architetures to recognize segmented dynamic Libras signs, and compare them with a baseline method. Another relevant contribution is the acquisition and publication of a large depth dataset composed of dynamic signs of Libras that represent letters of the alphabet and common words in Libras. Although the complete set of dynamic gestures in the Libras language is much higher than the selected signs, they are all composed by combinations of 46 *key handshapes* in movement [13], sharing similarities among them. Thus, the selected set of dynamic signs is still valuable to evaluate the proposed architectures. By finding a very robust deep model to recognize dynamic libras signs, we expect, in a near future, to build more sophisticated methods that would employ such model to recognize more complex interactions.

2 Methodology

We first describe the data acquisiton and pre-processing steps. We consider each dynamic sign as a finite sequence of depth images. In both training and recognition phases, the sequences are preprocessed before being given as input for the deep models. Depth data is acquired from a Intel RealSense sensor [8] at 640×480 spatial resolution, 8 bit depth resolution and 30 frames per second.

Let $f = \{f_1, f_2, \ldots, f_{|s|}\}$ be a dynamic sign, where $f_t[x, y]$ represents the depth (distance to the sensor) of the pixel at column x and row y of the tth image of the sequence. For each image f_t, we initially segment the hand from

the background by first identifying the closest distance d_t to the sensor (smallest value of f_t). We consider a depth threshold D ensuring that the hand of most individuals, performing any hand pose, is not incorrectly extracted or cropped from the image, even if some parts of the arm are not extracted from the image. The filtered image $\tilde{f}_t$ is given by

$$\tilde{f}_t[x, y] = \begin{cases} f_t[x, y], & f_t[x, y] \leq d_t + D \\ 0, & f_t[x, y] > d_t + D \,. \end{cases} \tag{1}$$

We achieve excellent results by empirically setting $D = 20$ cm. Note that, even if the hand is not perfectly segmented, we expect the neural networks to learn how to deal with instances of signs containing regions of the arm, for example.

After segmentation, each image is subsampled to 50×50 of spatial resolution. We empirically found that this resolution is sufficient for a human to identify any sign, and thus we expected the same capability from a neural network.

As the inputs of 3D CNNs are expected to have fixed size, we fix the time dimension T to match the longest example of our collected data set, which is $T = 40$. Instead of oversampling each shorter sequence, we pad them with zero images to fill up the remaining time steps. In what follows, we describe two different classes of deep architectures that are capable of learning features of different nature: 3D-CNNs and LRCNs. We formally describe both, highlighting the advantages of each one.

2.1 3D CNNs

3D Convolutional Neural Networks were initially proposed to recognize human actions in surveillance videos [9]. Later, it was found that such networks are more suitable for spatiotemporal feature learning compared to 2D CNNs, as the latter loses temporal information of the input video right after every convolution operation [14].

The input for 3D CNNs are videos where each frame has a single channel containing depth data with resolution $m \times n \times T$, which can be seen as a cube formed by stacking multiple contiguous frames together. Instead of applying 2D convolutions in each image, as in the classical 2D CNN approach, 3D convolutions are directly applied to the cube. Thus, the extracted features represent spatiotemporal information that combines data from different frames. In contrast, 2D convolutions may only capture spatial information.

In a 3D convolutional layer, a video (or 3D feature map) f^{l-1} from a previous layer $l - 1$ is given as input and K^l different 3D feature maps are computed by applying K^l different kernels w_k^l, jointly with a bias b_k^l and an activation σ, as

$$f_k^l[x, y, t] = \sigma((f^{l-1} * w_k^l)[x, y, t] + b_k^l) \,.$$

As shown in Fig. 1, 3D CNNs follow the same structure of 2D CNNs. In this work, given an instance of a dynamic sign of Libras, we give the whole depth video as input to a 3D CNN.

2.2 LRCN

Long-term Recurrent Convolutional Networks were recently proposed and experimented on three applications involving sequences: activity recognition, image captioning, and video description [7]. They combine the power of 2D CNNs to extract relevant spatial features, with LSTMs: a class of recurrent neural networks (RNNs) that can learn long-term temporal dependencies. Consequently, this class of networks is capable of learning dependencies of spatial features over the time. Advantages of recurrent models include the capability of modelling complex temporal dynamics; and directly mapping variable-length inputs, such as videos, to variable-length outputs.

As in classical RNNs, an LSTM unit also has a memory (state) which is updated at each time step of a sequence. However, differently from RNN, LSTM learns when to forget and when to update previous states by incorporating information from the current step. The final output of the network depends on the last updated state. For more details, see the work of Donahue *et al.* [7].

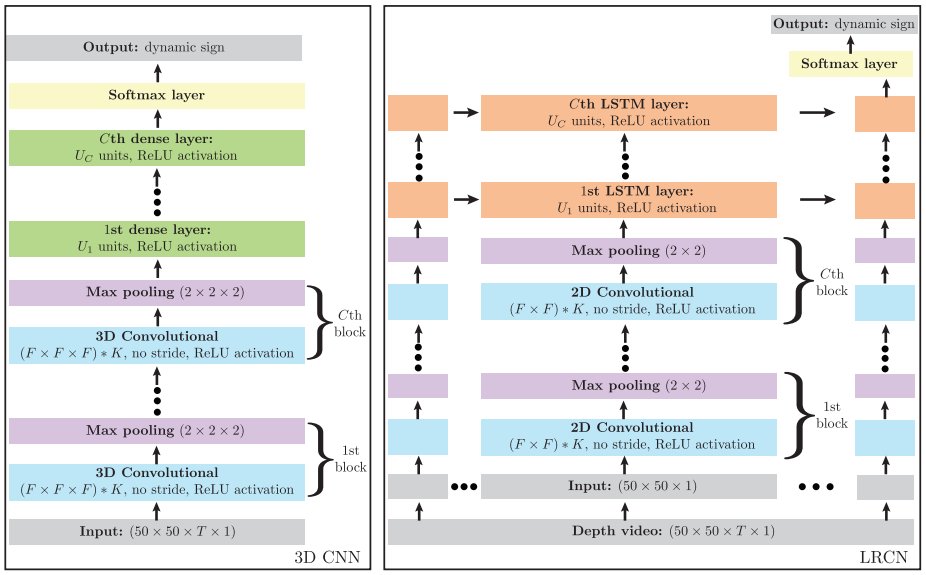


Fig. 1: 3D CNN architecture (left): the whole depth video is given as input to convolutional blocks composed of 3D convolutional layers (each one applying K filters of size $F \times F \times F$), and max pooling layers. Then, flattened features are given to dense layers that precede a softmax layer, which returns the final classification. LRCN architecture (right): Frames from the depth video are processed independently by 2D convolutional blocks. The extracted feature maps are flattened and given as input to the LSTM layers, that update their states. When the last frame is processed, the LSTM states, together with a softmax layer, are used to give the sign classification as output.

Differently from the 3D CNNs, LRCNs receive as input one image per step. Convolutional layers jointly with max pooling layers hierarchically learn features at different scales, similarly to 2D CNNs. Then, the extracted features are flattened and given as input to LSTM layers that update their states and compute partial (until the current time step) outputs. Finally, a dense softmax layer computes the partial probability.

To classify dynamic signs of Libras, we consider that, for each time step, a depth image is given as input and a partial probability is computed by the network. However, only the final probability, *i.e.* the probability distribution given by the softmax layer after the last image of the sign is processed, is considered for classification. The complete architecture is shown in Fig. 1.

3 Experiments

In our experiments we use a novel dataset containing dynamic Libras sign gestures, which is publicly available[3]. Assisted by Libras specialists, we designed a set of 10 classes representing dynamic signs of Libras: the letters H and J; and the words "day", "night", "enter", "all", "again", "start", "course" and "yours". The dynamic signs were chosen to satisfy the following requirements: 5 one-handed signs and 5 two-handed signs; all signs should be very distinct, to increase diversity, except for two pairs of similar signs which are harder to discriminate ("night" and "enter"), to better evaluate the robustness of the models. For each class, we collected a total of 300 examples of depth videos using the hardware and specifications described in Section 2. The number of frames per example varies, and the longest example contains 40 frames. Fig. 2 shows the first and last frames of one example of each sign.

3.1 Hyper-parameters Optimization and Evaluation

The proposed classes of architectures represent a large number of models defined by several hyper-parameters. Instead of empirically setting values as many previous works, we optimize hyper-parameters by searching over a uniform sample

[3] http://im.ufal.br/professor/thales/libras/

Fig. 2: Dynamic Libras sign dataset: images showing the first (top) and last (bottom) frames of each trained sign, after background extraction. From the left to right, the one-handed signs are H, "day", "again", "yours", J; and the two handed-signs are "night", "enter", "all", "start", "course".

of the search space. We consider the following hyper-parameters for 3D CNNs and LRCNs:

- Number of convolutional layers: $C_{\text{3DCNN}} = \{1, 2\}$ and $C_{\text{LRCN}} = \{1, 2, 3\}$;
- Filter size of convolutional layers: $F_{\text{3DCNN}} = F_{\text{LRCN}} = \{3, 5\}$;
- Number of filters of convolutional layers: $K_{\text{3DCNN}} = K_{\text{LRCN}} = \{16, 32, 64\}$;
- Number of dense/LSTM layers $L_{\text{3DCNN}} = L_{\text{LRCN}} = \{1, 2, 3\}$;
- Number of units of dense/LSTM layers $U_{\text{3DCNN}} = U_{\text{LRCN}} = \{100, 200, 300\}$.

Here, we consider both U_{3DCNN} and U_{LRCN} to be three-dimensional vectors containing the number of units up to 3 possible layers. If a layer does not exist, we set the corresponding number to zero. The same applies to F_{3DCNN} and F_{LRCN}. After each convolutional layer of all models, a max pooling layer with pool size $(2, 2)$ and no stride is applied. For all layers of both architectures, we use the ReLU activation function given by $\sigma(x) = \max(0, x)$, with a few exceptions, such as LSTM units and softmax layers.

For each combination of hyper-parameters, we perform a cross-validation experiment by randomly splitting the dataset examples into training and test sets, considering 70% of the examples for training and 30% for testing. The validation accuracy is evaluated after 10 training epochs for LRCNs and 5 training epochs for 3D CNNs. The mean validation accuracy after 10 repetitions is the value to be optimized, representing the robustness of the evaluated model.

We employed the Adam optimization algorithm [10] of the Keras library [6], with an initial learning rate of 0.001. The 5 best performing models of each class of architectures are shown in Table 1. Excellent results were obtained from the best configurations, achieving more than 99% of accuracy in all these cases.

To analyze the sensitivity of both classes of architectures to hyper-parameter calibration, we investigated the accuracies distribution among all performed experiments. Histograms of accuracies for both classes of architectures are exhibited in Fig. 3. Clearly, most accuracies are concentrated near the best results of Table 1. We conclude that the robustness of both classes of architectures is not very sensitive to hyperparameter calibration, with most evaluated configurations achieving more than 90% of accuracy. Consequently, the number of trainable parameters (weights) is a relevant criterion to select the most appropriate model. As shown in Table 1, the best performing LRCN is the best choice, due to the smaller number of weights.

We conclude that both classes of architectures can robustly be applied to the recognition of dynamic Libras signs. The excellent results can be explained by the relatively simplicity of dynamic Libras signs, and give evidence that such models are appropriate to more complex situations, including, larger dictionaries of signs and continuous conversations. However, in contrast to 3D CNNs, LRCNs have the ability to output partial probabilities for each time step, which may be very useful in such situations.

3.2 Comparison

In this section we compare the best performing deep model with a baseline method. The objective is to show that, although most dynamic Libras signs

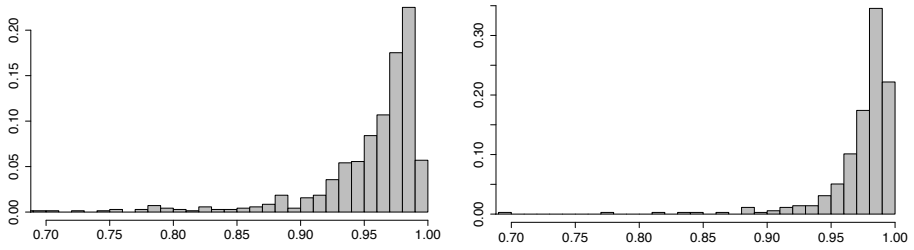
Table 1: Accuracy of the best evaluated 3DCNNs and LRCNs models.

class	C_*	F_*	K_*	L_*	U_*	mean accuracy	weights
LRCN	2	(3, 3, 0)	64	2	(100, 100)	99.8%	3,256,978
3DCNN	2	(3, 3, 0)	64	3	(300, 200, 100)	99.8%	18,779,658
3DCNN	2	(5, 3, 0)	32	3	(300, 200, 100)	99.6%	5,656,874
LRCN	1	(3, 0, 0)	64	2	(100, 100, 0)	99.7%	14,868,050
LRCN	1	(5, 0, 0)	64	2	(200, 200, 0)	99.7%	27,570,074
LRCN	1	(3, 0, 0)	64	3	(300, 200, 200)	99.7%	45,322,250
LRCN	1	(5, 0, 0)	64	3	(300, 200, 200)	99.6%	41,713,674
3DCNN	2	(3, 3, 0)	32	2	(200, 300, 0)	99.5%	6,287,286
3DCNN	2	(5, 3, 0)	64	2	(300, 100, 0)	99.5%	11,437,938
3DCNN	2	(3, 3, 0)	16	3	(100, 100, 200)	99.5%	6,287,286

are relatively easy to be recognized by deep models, a naive classifier will not be appropriate in this situation. In this experiment, we consider that the same inputs given to the 3D CNNs (raw depth video) are given to a multi-class one-versus-all SVM classifier. We adopt the popular rbf kernel, and calibrate both its radius γ and the regularization parameter C. Both parameters are calibrated by exhaustive grid search and cross-validation, using the same fraction of 70% of samples for training and 30% for validation. The best performing model was given by $\gamma = 0.1$ and $C = 1$, achieving only 63% of accuracy, in comparison with 99.8% of the best deep model of Table 1.

4 Conclusion

In this work we proposed and experimented deep models for the recognition of dynamic signs of Libras from depth data. Several variants of LRCNs and 3D CNNs revealed high robustness for the recognition of segmented signs. As future work, we expect the proposed models to be combined with other techniques, such as skeleton trackers, to recognize more complex conversations of Libras, including



(a) 3DCNNs accuracies distribution (b) LRCNs accuracies distribution

Fig. 3: Histograms of accuracies of the evaluated deep models. Horizontal axes represent relative accuracies, and vertical axes the relative frequencies.

arms and body movement, so that the machine would be capable of translating a Libras conversation in real-time. The collection of larger datasets would also be required to turn this objective feasible. In addition, research on online recognition methods would also be necessary, in such a way that unsegmented data could be given as input to the machine to identify phrases and words.

Acknowledgements. The authors would like to thank CNPq and FAPEAL for partially financing this research.

References

1. Bastos, I.L., Angelo, M.F., Loula, A.C.: Recognition of static gestures applied to brazilian sign language (libras). In: SIBGRAPI. pp. 305–312. IEEE (2015)
2. Brito, L.F., Langevin, R.: The sublexical structure of a sign language. Mathematiques Inform. et Sciences Humaines **125**, 17–40 (1994)
3. Cardenas, E.E., Camara-Chavez, G.: Fusion of deep learning descriptors for gesture recognition. In: Iberoamerican Congress on Pattern Recognition. pp. 212–219. Springer (2017)
4. Cardenas, E.J.E., Chávez, G.C.: Finger spelling recognition from depth data using direction cosines and histogram of cumulative magnitudes. In: SIBGRAPI. pp. 173–179. IEEE (2015)
5. Cheok, M.J., Omar, Z., Jaward, M.H.: A review of hand gesture and sign language recognition techniques. International Journal of Machine Learning and Cybernetics pp. 1–23 (2017)
6. Chollet, F., et al.: Keras. `https://keras.io` (2015)
7. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2625–2634. IEEE (2015)
8. Intel: Intel realsense technology (2018), `https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html`
9. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence **35**(1), 221–231 (2013)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Moreira, G.d.S.P., Matuck, G.R., Saotome, O., da Cunha, A.M.: Recognizing the brazilian signs language alphabet with neural networks over visual 3d data sensor. In: Ibero-American Conference on Artificial Intelligence. pp. 637–648. Springer (2014)
12. Pizzolato, E.B., dos Santos Anjo, M., Pedroso, G.C.: Automatic recognition of finger spelling for libras based on a two-layer architecture. In: Proceedings of the 2010 ACM Symposium on Applied Computing. pp. 969–973. ACM (2010)
13. Quadros, R.M.d., Karnopp, L.B.: Língua brasileira de sinais: estudos linguísticos. Artmed, Porto Alegre (2004)
14. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Computer Vision (ICCV), 2015 IEEE International Conference on. pp. 4489–4497. IEEE (2015)