Sign Language Recognition Based on 3D Convolutional Neural Networks

Geovane M. Ramos Neto(✉), Geraldo Braz Junior(✉), João Dallyson Sousa de Almeida(✉), and Anselmo Cardoso de Paiva(✉)

Computing Applied Group, Federal University of Maranhão, São Luís, Brazil
{geovane.menezes,geraldo,jdallyson,paiva}@nca.ufma.br

Abstract. The inclusion of disabled people is still a recurring problem throughout the world. For the hearing impaired, the barrier imposed by the sign language spoken by a small part of the population imposes limitations that interfere in the quality of life of these people. The popularization or even automation of sign language recognition can take their lives to a higher level. Understanding the importance of sign language recognition for the hearing impaired we propose a 3D CNN architecture for the recognition of 64 classes of gestures from Argentinian Sign Language (LSA64). We demonstrate the efficiency of the method when compared to traditional methods based on hand-crafted features and that its results outperform most deep learning-based work reaching 93.9% of accuracy.

Keywords: Sign language recognition · 3D CNN · Computer vision

1 Introduction

The inclusion of disabled people is still a recurring problem throughout the world. In the case of hearing impairment, the problem lies in the difficulty that they have in communicating using a visual language code, a Sign Language (LS). One of the major difficulties for the hearing impaired is the low number of people who are fluent in sign language, which makes learning difficult and the communication of these, especially in the early stages of the development of the individual [1,2].

Currently, solutions for the recognition of a sign language are given primarily by the use of human translators, and are therefore expensive solutions, given the necessary professional experience. Sign language recognition seeks to develop algorithms and methods that can correctly classify a sequence of signals to understand its meaning.

Given the complexity of obtain meaning for each of the elements that make up a sign language pose, many methodologies treat the recognition of sign language as a gesture recognition problem, that is, the solutions seek to identify optimal characteristics that satisfactorily represent a certain gesture and methods that can classify it correctly given a set of possible gestures.

A. Campilho et al. (Eds.): ICIAR 2018, LNCS 10882, pp. 399–407, 2018.
https://doi.org/10.1007/978-3-319-93000-8_45

The vast majority of gesture recognition studies seek to extract manually modeled features and then use them in a classifier for recognition. In this sense we find a work that uses features extracted from Kinect and Leap Motion, which based on the position and orientation of the fingers feeds a multi-class Support Vector Machine (SVM) to recognize 10 classes of American Manual Alphabet gestures present in a public dataset presented in this same paper [3].

Another work in the same direction was proposed by Ronchetti [4] through a Probabilistic SOM network (ProbSOM) [5] to recognize LSA64 [6] gestures, so that ProbSOM allows to infer statistically by grouping similar gesture classes and then determining which are the most important characteristics for the discrimination of each gesture.

Recent advances in processing capacity and the increase in the number of large databases are allowing the application of machine learning techniques that until then were almost impractical given their high computational costs and their needs for large amounts of data. Deep learning is one of the most promising techniques, being successfully used in automatic speech recognition [7–9], recommendation systems [10,11] and image recognition [12–15].

When it comes to the recognition of dynamic gestures, the work of Pigou et al. [16] employs two CNNs, whose inputs are the gray level and depth videos original from CLAP14 challenge dataset. The first network uses the original videos, while the second one cuts each frame so that the result contains only the user's hand. At the end, the characteristic vectors are combined and discriminated by the CNNs.

Also, Huang et al. [17] uses a CNN with convolutions in three dimensions (3D CNN). Each video was divided into 5, named: color-R, color-G, color-B, depth and body skeleton, each with 9 frames, the first 3 referring to the color channel of the original video. The database used in the work is private and contains 25 classes of gestures.

Although it does not deal with the recognition of Sign Language and gestures in general, it is worth mentioning the work of Molchanov et al. [18] which uses Augmentation and two 3D CNN networks in parallel. The first one has VIVA challenge's dataset [19] as input and the second uses the same images of the first but significantly reduces the dimensions of the images. The resulting probabilities of the final layer of each network are combined through conditional probability to then classify them into one of the 19 classes.

The objective of this work is to present an efficient computational methodology for the recognition of signals from the Sign Language of Argentina (LSA), through computer vision and machine learning techniques. We will present our 3D Convolutional Neural Network (3D CNN) [20] architecture and use it to represent and classify 64 LSA gestures present in the LSA64 [6] video dataset.

The main contribution of this work is to present a 3D CNN architecture specifically tuned for signal language recognition in order to produce generalist and efficient results in comparison to other works presented in the literature. We also intent to provide a method that could be applied to improve life quality, inclusion and communication over people that are no capable to use sign languages.

2 3D CNN

3D convolution [20] is a mathematical operation where each voxel present in the input volume is multiplied by voxel in the equivalent position of the convolution kernel. At the end, the sum of the results is added to the output volume. In the Fig. 1 it is possible to observe the representation of the 3D convolution operation, where the voxels highlighted in the Input are multiplied with their respective voxels in the Kernel. After these calculations, the sum of them is added to the Output, generating the value of the highlighted voxel.

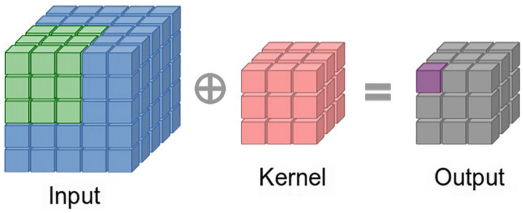

Fig. 1. Representation of a 3D convolution operation.

Since the coordinates of the input volume are given by (x, y, z) and the convolution kernel has size (P, Q, R) the 3D convolution operation can be defined mathematically as:

$$O_{xyz} = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} \sum_{r=0}^{R-1} K_{pqr} I_{(x+p)(y+q)(z+r)} \tag{1}$$

where O is a result of the convolution, I is the input volume, K is the convolution kernel and (p, q, r) are the coordinates of K.

Convolutional Neural Networks (CNNs) [21] are a specialized type of neural network for processing grid data (2D). The name of this type of neural network is closely linked to a mathematical operation called convolution, where matrix multiplications are replaced by the convolution operation 2D in at least one of its layers.

Its 3D extensions, called 3D CNN [20], differ from CNN networks because they use at least one 3D convolution. These convolutions can, in addition to extracting spatial information from matrices like 2D convolutions, extract information present between consecutive matrices. This fact allows us to map both spatial information of 3D objects and temporal information of a set of sequential images.

A layer that is not needed on a CNN 3D network however is very used is the pooling layer [16,17,22]. A typical pooling layer that calculates the maximum value of a neighborhood of a tensor is called MaxPooling. Specifically, MaxPooling 3D is the layer that calculates the maximum value of a 3D tensor [23,24].

The pooling of adjacent units can be done with the stride of more than one line, column or depth, reducing the size of the input and creating invariance to small displacements and distortions, which can increase the generalization power of the network [23].

These networks can be trained using the RMSProp (Root Mean Square Propagation) [25] optimizer. This is an adaptive step size method that scales the update of each weight through the average of its gradient norm. It is a strategy of minimizing the error, based on the robustness of the RProp [26], the efficiency of the mini-batches as well as an effective balancing of them.

Each kernel acts like filter, that is, when discriminant features are superimposed by the filter, these have a high output value (high neuron stimulus), indicating that they have found a pattern that can represent a certain feature.

3 Proposed Method

The method proposed in this work consists of configuring an architecture based on the 3D CNN concept for the recognition of Argentine Sign Language (ASL).

For all tests and model estimation, it was used the sign dataset LSA64 for Argentinian Sign Language includes 3200 videos where 10 non-expert subjects executed 5 repetitions of 64 different types of signs using one or both hands. Signs were selected among the most commonly used ones in the LSA lexicon, including both verbs and nouns [6]. Some examples of the dataset are presented in Fig. 2.

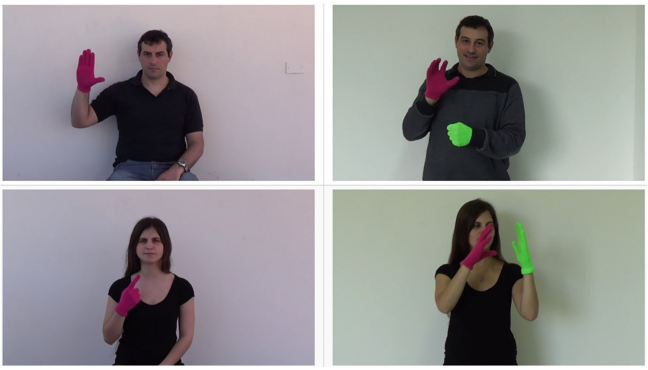

Fig. 2. Sample frames extracted from LSA64.

The dataset was collected under different luminosity conditions. Each sign was executed imposing few constraints on the subjects to increase diversity and realism in the database. The resolution of each video is 1920 by 1080, at 60 frames per second [6]. We used the LSA64 cut version, which is similar to the

raw version but each video has been temporally segmented so that the frames in the beginning or end of the video with no movement in the hands were removed.

Our Method consists in a pre-processing step and subsequent model fit and evaluation using our 3D CNN architecture. The next subsections presents the details of the proposed method.

3.1 Preprocessing

Each LSA64 video has a different sizes. To normalize video sizes to 30 frames we use the Nearest Neighborhood Interpolation (NNI) [27] strategy that removes or repeats frames as needed. For example, if the original video contains 40 frames, each multiple frame of 4 is removed and if the original video contains 20 frames, each multiple frame of 2 is repeated.

In order to reduce the computational cost, the dimensions of the videos were reduced to 80 × 45 pixels. In this way we maintained the original aspect ratio so that there was no significant impact on the results.

3.2 Proposed 3D CNN Architecture

The proposed 3D CNN architecture has two stages, the first contains 3 layers of feature extraction, obtained through 3D convolutions and MaxPooling 3D. The second stage is where classification is done using the features previously extracted in 3 non-linear Fully Connected (FC) layers.

The architecture used in this work is visually represented in Fig. 3, we chose to represent the 3D convolution layers and MaxPooling 3D together in order to simplify the representation.

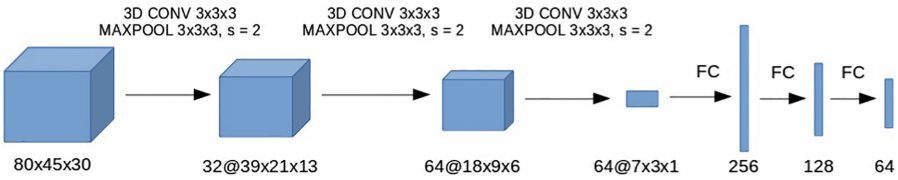

Fig. 3. Proposed 3D CNN architecture for Argentinian Sign Language recognition.

Full architecture has 3 3D convolutional layers, each followed by a 3D MaxPooling layer. The first convolution used 32 filters (3, 3, 3) and is followed by a MaxPooling 3D layer of kernel (3, 3, 3) and strides (2, 2, 2). The second and third contains 64 filters (3, 3, 3) and both are followed by a MaxPooling 3D kernel layer (3, 3, 3) and strides (2, 2, 2). All layers are activated by ReLu function.

The network architecture was adjusted through the optimization of a parameter search space, from the number of layers, to the quantity and size of the filters. The tests performed all combinations of filters (5, 5, 3) with filters (3, 3, 3) with

the amounts 32, 64, 64 filters, in that order. Each convolution layer was followed by a kernel MaxPooling layer (3, 3, 3) and strides (2, 2, 2). The motivation of the convolutions always to use kernel with size 3 in depth is that at the end of the extraction of characteristics we wanted to reduce the volume for only one image. Each setup quoted in this section has been tested 3 times and used the split ratio of the 80/20 video base, that is, 80% for training and 20% for testing.

After the last layer of MaxPooling, the architecture has 3 Fully Connected (FC) layers of 256, 128 and 64 neurons in this sequence. We did incremental tests starting with a single FC layer, until at 4 FCs the result remained practically the same as 3 layers, so we chose to leave only 3 layers.

The first two FC layers uses the ReLU activation function, while the last uses the softmax function in order to interpret the 64 outputs of the network as the probabilities of the input being one of the 64 classes present in the base.

Initially we put 300 epochs for training, but from 50 epochs the result was not increasing enough to justify the computation time, so we chose 50 epochs and batch size 32 as empirical values. As a loss function we used cross entropy, and the chosen optimizer was the RMSprop with suggested parameters [25] given its success in many studies [28,29] (learning rate of 0.001, ρ of 0.9 and decay equal to 0). To build the network we use the Keras [30] library and Theano [31] as backend.

4 Results

We evaluated the results through 5 tests, dividing the data set randomly using the proportion of 80% for training and 20% validation, resulting in 2560 videos for training and 640 videos for testing. Each set is created randomly and performing this procedure we seek to exempt the random factors present in the tests.

The mean values for accuracy, sensitivity and precision of the tests are given in Table 1. Proposed method reaches 93.9% of mean accuracy, with standard deviation of 1.4. We notice a very stable relation between the 5 tests evidenced by the low variation even when applied over random validation dataset generation.

Given the results, we compare them with works that use hand-crafted features and deep learning to extract characteristics. Table 2 contains the most relevant information in the comparisons between the results of the proposed architecture and related works.

When we compare the results with works that use hand-crafted features, we see that we have a slight advantage, reaching 2.6 percentage points more than [3]. And 2.2 percentage points over [4] that uses the same database. We can suggests

Table 1. Proposed method results.

	Accuracy (%)	Sensibility (%)	Precision (%)
Mean	93.9	93.7	94.9
σ	1.4	1.58	1.58

Table 2. Comparison with related works

Proposed	Method	Dataset	nClasses	Acc (%)
	3D CNN	**LSA64**	**64**	**93.9 ± 1.4**
[3]	Finger Position	MKLM	10	91.3
[4]	ProbSOM	LSA64	64	91.7
[16]	CNN	CLAP14	20	91.7
[17]	3D CNN	Private	25	94.2
[18]	3D CNN	VIVA challenge	19	77.5

that our approach, that capture information over spatial and temporal data using 3D CNN provide better information and outperforms these works.

The architecture was also promising when compared to other works that use deep learning. The work of [17] was the only result of related work whose result was better than the one proposed. However, we must point that the amount of information provided to the network was a great differential in this work, using 5 videos per individual while we used only one video in gray level. Finally, [16] despite using two networks for recognition, the results are pretty close and we still get 2.2 percentage points up using only a 3D CNN.

Taking into consideration the related works, there is a great indication that the proposed method is promising for the recognition of sign language, surpassing most of the comparative ones.

5 Conclusion

In this work we present a 3D CNN network architecture for the recognition of 64 classes of gesture of the Argentinian Sign Language. The results reach an accuracy rate of 93.9%, which indicates that deep learning can be adequately applied to the problem.

The great differential is the application of a 3D convolutional network to a database substantially larger than the related works, showing that its application is feasible for bases with a greater variety of classes. In view of the above, the proposed architecture has potential when compared to other architectures that also aim to recognize sign language gestures.

Some points still need to be improved such as broadening the base of gestures and diverse situations. We will also conduct studies on the use of 2D convolutions in order to provide a comparative analysis of the techniques and their real employability, in addition to testing the architecture with larger test sets.

References

1. Moeller, M.P.: Early intervention and language development in children who are deaf and hard of hearing. Pediatrics **106**(3), e43 (2000)
2. Dalton, D.S., Cruickshanks, K.J., Klein, B.E., Klein, R., Wiley, T.L., Nondahl, D.M.: The impact of hearing loss on quality of life in older adults. Gerontologist **43**(5), 661–668 (2003)
3. Marin, G., Dominio, F., Zanuttigh, P.: Hand gesture recognition with leap motion and kinect devices. In: IEEE International Conference on Image Processing (ICIP), pp. 1565–1569. IEEE (2014)
4. Ronchetti, F.: Reconocimiento de gestos dinámicos y su aplicación al lenguaje de señas. Ph.D. thesis, Facultad de Informática (2017)
5. Estrebou, C., Lanzarini, L., Hasperué, W.: Voice recognition based on probabilistic SOM. In: Proceedings of the Conference: XXXVI Conferencia Latinoamericana en Informática, At Asunción, Paraguay (2010)
6. Ronchetti, F., Quiroga, F., Estrebou, C., Lanzarini, L., Rosete, A.: LSA64: a dataset of Argentinian sign language. In: XX II Congreso Argentino de Ciencias de la Computación (CACIC) (2016)
7. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
8. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
9. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio Speech Lang. Process. **20**(1), 30–42 (2012)
10. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1235–1244. ACM (2015)
11. Wang, X., Wang, Y.: Improving content-based and hybrid music recommendation using deep learning. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 627–636. ACM (2014)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
14. Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8150, pp. 411–418. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40763-5_51
15. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649. IEEE (2012)
16. Pigou, L., Dieleman, S., Kindermans, P.-J., Schrauwen, B.: Sign language recognition using convolutional neural networks. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8925, pp. 572–578. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16178-5_40

17. Huang, J., Zhou, W., Li, H., Li, W.: Sign language recognition using 3D convolutional neural networks. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2015)
18. Molchanov, P., Gupta, S., Kim, K., Kautz, J.: Hand gesture recognition with 3D convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–7 (2015)
19. Ohn-Bar, E., Trivedi, M.M.: Hand gesture recognition in real time for automotive interfaces: a multimodal vision-based approach and evaluations. IEEE Trans. Intell. Transp. Syst. **15**(6), 2368–2377 (2014)
20. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013)
21. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
22. Oyedotun, O.K., Khashman, A.: Deep learning in vision-based static hand gesture recognition. Neural Comput. Appl. **28**(12), 3941–3951 (2017)
23. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
24. Giusti, A., Ciresan, D.C., Masci, J., Gambardella, L.M., Schmidhuber, J.: Fast image scanning with deep max-pooling convolutional neural networks. In: 20th IEEE International Conference on Image Processing (ICIP), pp. 4034–4038. IEEE (2013)
25. Tieleman, T., Hinton, G.: Lecture 6.5-RmsProp: divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw. Mach. Learn. **4**(2), 26–31 (2012)
26. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RProp algorithm. In: IEEE International Conference on Neural Networks, pp. 586–591. IEEE (1993)
27. Molchanov, P., Gupta, S., Kim, K., Pulli, K.: Multi-sensor system for driver's hand-gesture recognition. In: 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), vol. 1, pp. 1–8. IEEE (2015)
28. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016)
29. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128–3137 (2015)
30. Chollet, F., et al.: Keras (2015). https://github.com/keras-team/keras
31. Theano Development Team: Theano: a Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688, May 2016