Recibido: 17 de noviembre 2020 **Aceptado: 23 de febrero 2021**

Multilayer and convolutional neural networks for Bolivian Sign Language recognition: an empirical evaluation

Redes neuronales multicapa y convolucionales para el reconocimiento del lenguaje de señas boliviano: una evaluación empírica

Juan Pablo Rodríguez Villarroel, Nicolás Ponce de León Espinoza, Wendoline Arteaga Sabja

Departamento de Ciencias Exactas e Ingenierías, Universidad Católica Boliviana "San Pablo", Calle M. Márquez esquina Parque Jorge Trigo Andia Cochabamba, Bolivia

warteaga@ucb.edu.bo

Abstract: The deaf community is a social stratum with lots of struggles in daily life, chiefly cause for communication difficulties with the general public. Although each country has its sign language, which is the case of Bolivian Sign Language(BSL). However, only few people know it. Different approaches have been proposed to perform gesture recognitions and help people to translate sign language to a particular language, including neural networks. However, little is known about the effectiveness of the neural networks to detect Bolivian Sign Language (BSL).

This paper proposes and evaluates the use of two neural network techniques, multilayer (MLP) and convolutional(CNN), to recognize Bolivian Sign Language. Our approach takes as input the most significant frames from a video using a motion-based algorithm and applying a border detection algorithm in the selected frames. We present an experiment on which we evaluate these techniques using 60 videos of four basic BSL phrases. As a result, we found that MLP has an accuracy which ranges between 65% and 88%, and CNN ranges from 95% and 99%, depending of number of neurons and internal layers used.

Keywords: multilayer neural network, convolutional neural networks, computer vision, sign language recognition, BSL.

Resumen: La comunidad de sordos es un estrato social con muchas luchas en la vida diaria, principalmente causa de dificultades de comunicación con el público en general. Aunque cada país tiene su lengua de signos, como es el caso de la Lengua de Signos Boliviana (BSL). Sin embargo, pocas personas lo saben. Se han propuesto diferentes enfoques para realizar reconocimientos de gestos y ayudar a las personas a traducir el lenguaje de señas a un idioma en particular, incluidas las redes neuronales. Sin embargo, se sabe poco sobre la efectividad de las redes neuronales para detectar el lenguaje de señas boliviano (BSL).

Este artículo propone y evalúa el uso de dos técnicas de redes neuronales, multicapa (MLP) y convolucional (CNN), para reconocer el lenguaje de señas boliviano. Nuestro enfoque toma como entrada los fotogramas más significativos de un video utilizando un algoritmo basado en movimiento y aplicando un algoritmo de detección de bordes en los fotogramas seleccionados. Presentamos un experimento en el que evaluamos estas técnicas utilizando 60 videos de cuatro frases BSL básicas. Como resultado, encontramos que MLP tiene una precisión que varía entre 65% y 88%, y CNN varía entre 95% y 99%, dependiendo del número de neuronas y capas internas utilizadas.

Palabras clave: red neuronal multicapa, redes neuronales convolucionales, visión por computadora, reconocimiento de lenguaje de signos, BSL.

1 Introduction

Sign language is the main communication medium of deaf population. Sign language is a systematic language which includes fingerspelling, motions, lips reading, and another non-verbal expression. Sing language plays the important role in communication for deaf people, by tending to present the language visually with the use of signs (Ministerio de Educación Boliviano). Besides the existences of BSL sign language deaf people still have communication difficulties mainly because most of the people do not know sign language.

Diverse approaches have been proposed to improve this situation by providing people various mechanisms to help users: interpret sign language and/or translate a particular language to sign language. Such mechanisms may include particular hardware, such as gloves (Yang & Chen, 2020), bracelet (Pascual, 2014), or Kinect camera (PASTOR, 2013). On the other hand, the computer vision research community proposed a number of approaches to perform gesture recognition, which may be used to translate sign language taking to a particular language, for instance, English (Garcia & Alarcon). However, little is known about the effectiveness of neural networks to detect *Bolivian* Sign Language.

In this paper, we present an empirical investigation on the use of multi-layer and convolutional neural networks to translate Bolivian Sign Language to Spanish. Where we use a sample of image frames collected from a video as input. To select a sample of image frames of the video we use a movement-based selection technique. For this, we consider capturing each frame where the change of the hand direction is notorious on the video. The algorithm compares each frame and only selects when the movement of the person has a sudden change in direction. For all of these selected frames we apply a border detection algorithm proposed by canny etal. (CANNY, 1986). Finally, we analyze every representative gesture as a class for the neural network.

In our experiment, we evaluate these two neuronal networks to analyze 60 videos of four basic BSL phrases. Our experiment reveals MLP has an accuracy which ranges between 65% and 88%, and CNN ranges from 95% and 99%, for the signs we consider.

Structure. The following section of this paper as structured as follows: Section 2, describes the two neural networks techniques; Section 3, explains the video preprocessing technique used for the experiment; Section 4, describes the methodology; Section 5, are the results obtained using both of the neural networks; Section 6, the discussion and future work; Section 7, shows the related work; Section 8, represents the conclusions obtained based of the result.

2 Multi-layer and Convolutional Neural Networks

This section briefly describes the two neural networks under analysis.

2.1 Multilayer perceptron neural network architecture (MLP)

Multilayer perceptron neural networks are feedforward networks, which mean it has one or more hidden layers, besides the input and output layer (Haykin). Each one of these layers has one or more processing units or neurons and every one of them is completely communicated with the ones in the previous layer; each link has an activation function. Each link j, i has an associated weight Wj, i that determines the strength and sign of the link and spreads the activation a_j. First, each neuron i calculates a weighted sum of its inputs in_i:

$$in_i = \sum_{j=0}^{n} W_{j,i} a_j$$

Equation 1: Weighted sum of inputs.

Source: (Russell)

Next, the activation function g is applied to the sum, and outputs a_i the output that will be spread to the neurons in the next layer:

$$a_i = g(in_i) = g\left(\sum_{j=0}^{n} W_{j,i} a_j\right)$$

Equation 2: activation function application

Source: (Russell)

At this point, it is important to clarify that during the weighted sum, a $W_{0,j}$ bias weight was added; this bias represents the real neuron threshold, which means, it will activate if the sum of the weight and the input is positive (Russell).

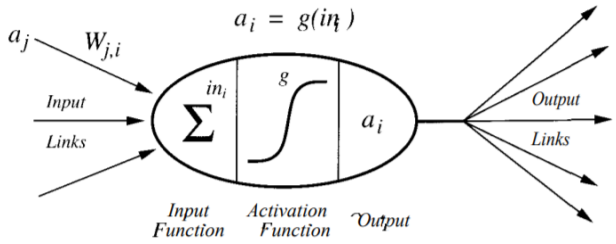

Figura 1 Artificial neuron of a MLP network. Source: (Russell)

2.2 Convolutional neural network architecture

Convolutional neural networks follow the same structure as a normal neural network, but they use a particular layer called convolution layer. (LeCun) Convolution is a mathematical operation and it is used to analyze the features of the image by parts and it's denoted as the equation below:

$$S(t) = \int (x * w)(t)$$

Equation 3: Convolution

Source: (Goodfellow)

In convolutional network terminology, the first argument x is often referred to as the input of the neural network and the second argument w is the kernel. The output is referred to as the feature map. This input usually is a multidimensional array of data and the kernel a multidimensional array of parameters. These arrays are known as tensors. (Goodfellow) The kernel is used on the data as the image below:

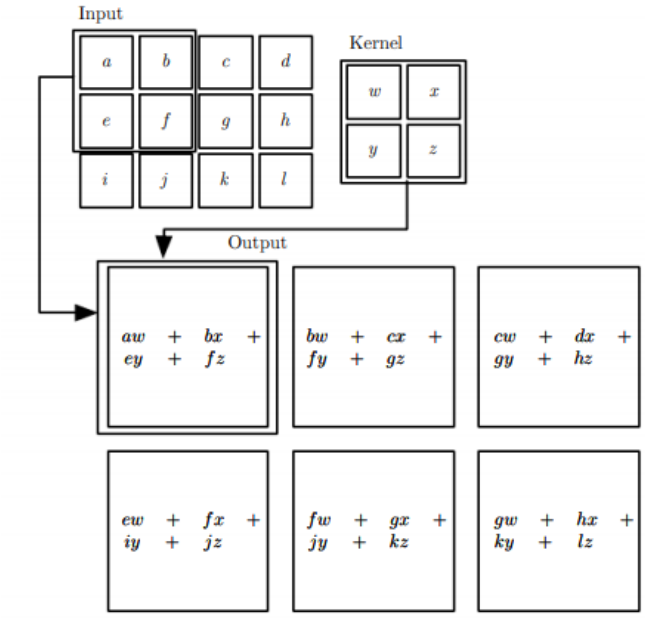

Figura 2 2D Convolution. Source: (Goodfellow)

In this example, the 2x2 kernel is applied to each sub matrix of the input, and the result is a new matrix with fewer elements called feature map. It is common to use a process after the convolution layer; this technique is called max-pooling and is represented as the image below:

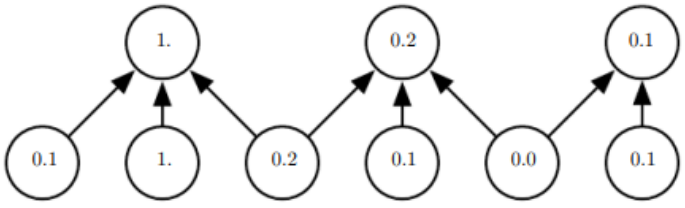

Figura 3 Max pooling. Source: (Goodfellow)

This process consists of selecting the maximum values of the matrix resulting from the convolution layer, this value is selected from a sub matrix of the feature map through a window, and these windows have a defined size and a pass for every group of pixels of the feature map. All these max values represent the features of the initial input and are a multidimensional array that is why at the end of this process a flatten activation is used to obtain a one-dimensional array that will be used as the input to a normal neural network.

2.3 The Backpropagation algorithm

A neural network can be represented as function $h(x, \emptyset)$, where X is an example represented as an integer array and to which we want to obtain the belonging class; meanwhile $\emptyset$ represents a parameter vector that the network will use, these are the weights. There are as many weights as links in the network, trying to define its correct value of each one of them by hand is quite a challenging job, maybe impossible. But here is where the Backpropagation joins the game (Buduma). Taking the next MLP network as an example:

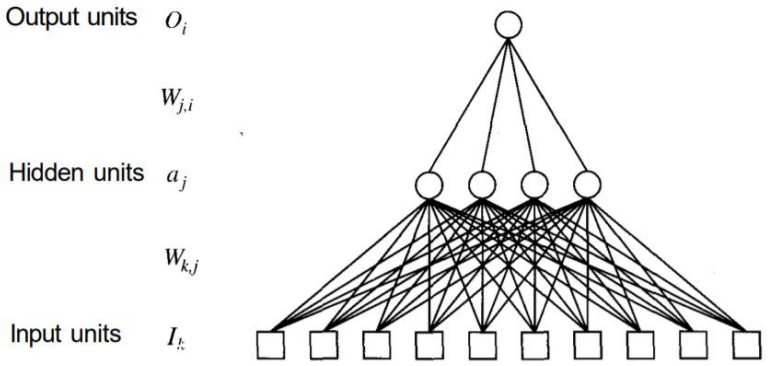

Figura 4 Multilayer Neural Network. Source: (Russell)

The algorithm has 2 phases: the first one takes place when the network is predicting the class of some example provided during the training, the data flows through each layer until we get a vector of predictions $h_w(x)$ in the output layer; it is at this point where a comparison is made, between this vector and the real value of the provided example, calculating the error E as follows:

$$E = \frac{1}{2}\left(y - h_w(x)\right)^2$$

Equation 4: Error calculation

Source: (Russell)

The higher the value of E is, the worse the network performs, but on the other hand, the closer this value is to 0, the better the network will perform. While the value of the error in this layer is representative and gives the necessary information for a correction, this does not happen in the rest of the hidden layers, because the error there is unknown. And is here where the second phase of the algorithm begins, with the weights update, starting with the ones in the output layer, as follows:

With $\Delta_i = E_i \, X \, g'(in_i)$ =>

$$W_{j,i} = W_{j,i} + \propto X\ a_i\ X\ \Delta_i$$

Equation 5: Output's weights update

Source: (Russell)

Where $\propto$ is the learning rate and g' is the activation function derived. Now the hidden neurons update is based in the idea that: the hidden neuron *j* is responsible of a fraction Δ_i in every neuron in the output layer to which is linked. By this way, the values of Δ_i are divided based on the link's strength and it backpropagates the Δ_j values, which is obtained as follows:

$$\Delta_j = g'(in_j) \sum_i \quad W_{j,i} \Delta_i$$

Equation 6: Delta j calculation

Source: (Russell)

And thus, the remaining hidden weights update begins:

$$W_{k,j} = W_{k,j} + \propto X\ a_k\ X\ \Delta_j$$

Equation 7: Hidden weights update

Source: (Russell)

This algorithm could summarize as follows:

- Obtain the Δ values for the output units, using the observed error.
- Starting in the output layer, repeat for every layer in the network until the first hidden layer is reached (Buduma; Russell):
    - Backpropagate the Δ values to the previous layer.
    - Update the weight between those two layers.

3 Video Preprocessing: Frame Sampling and Border Detection

The presented approach takes a video as input, which we preprocess using two steps: first, we take representatives frames of the video as a sample, and then we apply the border detection algorithm to these frames, to finally send the end result as input to the neural network.

3.1 Movement-based Frame Sampling.

The movement detection algorithm consists of comparing each frame of a video and comparing with its predecessors. If it is a huge change in motion, that instant is

captured and called a key point. When processing each sign, 1 or more key points will be obtained and each of these will be analyzed as a class in the neural network. For example preprocess the sign 'Cochabamba' will give the result shown on Figure 5.

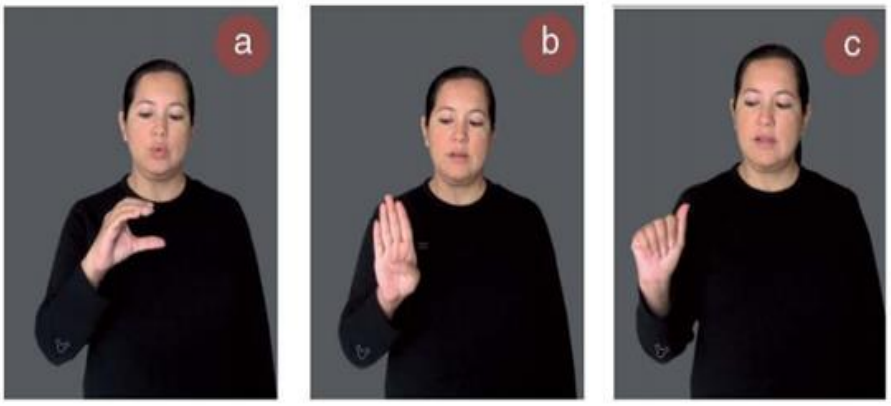

Figura 5 Movement detection algorithm result

Three frames were selected by the movement detection algorithm for this particular sign that is conformed by the letters: 'c', 'b' and 'a' on BSL. Each of these selected frames will continue with pre-process techniques to get the feature of the images.

3.2 Border Detection and Image Preprocessing.

For each selected frame from the video, we bring the image to grayscale, apply a blur filter, detect the borders using the Canny border detection algorithm, and rescale the image to a desired number of pixels high and wide, in this case 300 height and 300 wide. Bringing an image to grayscale greatly reduces unnecessary data, since RGB matrices are no longer used to represent colors, only a matrix with pixels representing 1 - 255 in black and white levels is needed. A blur filter is used to eliminate noise from the image and then the Canny filter which highlights only the borders that it finds in the image and in this way a matrix is obtained with only the values that interest us. Finally, a rescaling is performed so that all the images obtained have the same dimensions in height and width, in addition to seeking less processing in the neural network. (Hninn, 2009) Figure 6 illustrates the descriptor extraction process.

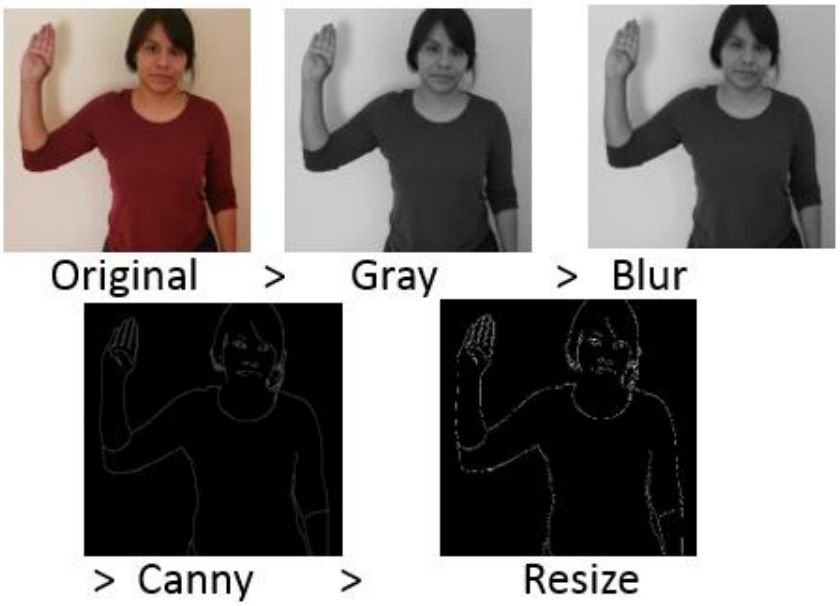

Figura 6 Descriptor Extraction

The techniques used in this step are common techniques in the field of computer vision, the use of that sequence of filters and processes guaranteed the feature extraction of an image. (Hninn, 2009)

We used some techniques to improve the quality of the dataset and avoid overfitting. We confirm that the best results were obtained for the datasets, where the data augmentation strategy was emplaced to generate the data. (Núñez, 2017)

With this idea the neural network now learns the components of a sign, for example, the three gestures that make up the sign 'Cochabamba' as shown in Figure 5. These components are found with the motion detection process, a change was made for it to save the individual frames in a directory. After the filter applied each of these gestures as a raw picture will conform to a class for the input of the neural network.

4 Experiment Setup

To validate the effectiveness of the proposed approaches. We designed a six step methodology and structure it in a workflow (Figure 7):

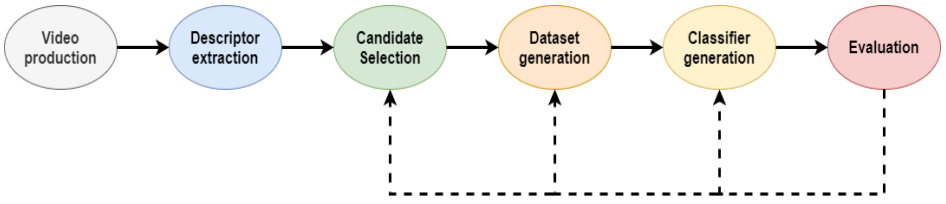

Figura 7 Research methodology.

A following we describe each one of the steps:

Data Set. In the face of the lack of an existing video dataset containing BSL sign examples for the networks learning process; we proceeded to produce it. So, the first step of the process was the dataset acquisition, whose elements are videos containing BSL. Not having a set of videos for the generation of descriptors that make up the dataset, we proceeded to establish it. For our experiment, we focus on ten signs: auto, coffee, Cochabamba, what?, thank you, hello, please, want and me. We choose these signs based on the first book-module of the Bolivian Sign Language, provided by the "Ministerio de Educación Boliviano". For each sign, we produce a sample of 15 videos, making a total of 150. These videos correspond to different persons doing each sign. Each of these videos follows, as far as possible, the following guidelines:

- Good lighting.
- Only one person in the video.
- High contrast between the hands, face and body of the person in the video.
- The person should only be focused from the waist up, since the legs are not used for any gesture.
- As far as possible, no objects in the background of the scene.

Video Preprocessing. We have used the preprocessing techniques described in section 3. We apply the motion detection algorithm on the videos to select the most representative gestures of the sign and then we use filters to highlight the image features (border detection).

Classifier generation. Machine learning techniques are used during this stage, to search a model that infers the rules for future examples classification. In this case, this generation is made by a multilayer and a convolutional neural network, setting a few parameters in order to find trends. The following parameter was used for the configurations:

MLP:

- Input classes. The number of gestures that represent the signs:

  - Word: Cochabamba (one-hand represented). 3 gestures/classes.
  - Words: Please, Coffee and hello (two-hand represented). 3 gestures/classes.
- Number of neurons per hidden layer. We use the following sets of ranges: [11 - 23] neurons on the hidden layer.
- Activation function. The activation function used was sigmoid.
- Number epochs. This experiment only used 5 epochs.

CNN:

- Input classes. The number of gestures that represent the signs:
  - Word: Cochabamba (one-hand represented). 3 gestures/classes.
  - Words: Please, coffee and hello (two-hand represented). 3 gestures/classes.
  - Words: Please, coffee, hello and want (two-hand represented). 4 gestures/classes. Where coffee and want use similar gestures.
- Number of neurons per hidden layer. We use the following sets of ranges:
  - [11 - 23] neuron on the hidden layer.
  - [15, 20, 25] neuron on the hidden layer.
- Activation function. The activation function used was sigmoid.
- Number epochs. We evaluated 1, 2, 3 and 5 epochs.
- Number of convolutional layers. The first experiments only used 1 convolutional layer, for the last one we used 1, 2 and 3 convolutional layers.

Accuracy and recall. A performance analysis and comparison is made using the accuracy and recall metrics. The two techniques used in this experiment are (Buduma):

- **Accuracy:** The most common metric, which is in this case is the percentage of times that the neural network successfully classifies a sing. This metric is obtained as follows:

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions}$$

Equation 8: Accuracy

Source: (Goodfellow)

Recall: Is the fraction of the class elements that were correctly classified:

$$recall = \frac{Number\ of\ correct\ class\ predictions}{Total\ number\ of\ class\ elements}$$

Equation 9: Recall

Source: (Goodfellow)

5 Results

5.1 MLP

Figure 8 shows the results obtained by the multilayer neural network for the 'Cochabamba' sign, which consists of 3 gestures and only one hand is used and the signs 'please', 'coffee' and 'hello'. In the same way they consist of 3 gestures in total, but both hands are used to be represented. In this way, Figure 8 shows the comparison of the results obtained by a sign that only uses one hand to be represented with three signs that use both hands.

Accuracy. - In both cases, the MLP did not achieve cases greater than 88% and it seems that due to the similarity of the gestures it was more difficult for the 'Cochabamba' sign since it only changes the position of one hand. Given the results, it can be affirmed that the MLP does not classify satisfactorily either of the two sets of gestures and the tendency to increase neurons does not seem to improve accuracy. This is why it is not considered to increase more signals to the sets of gestures.

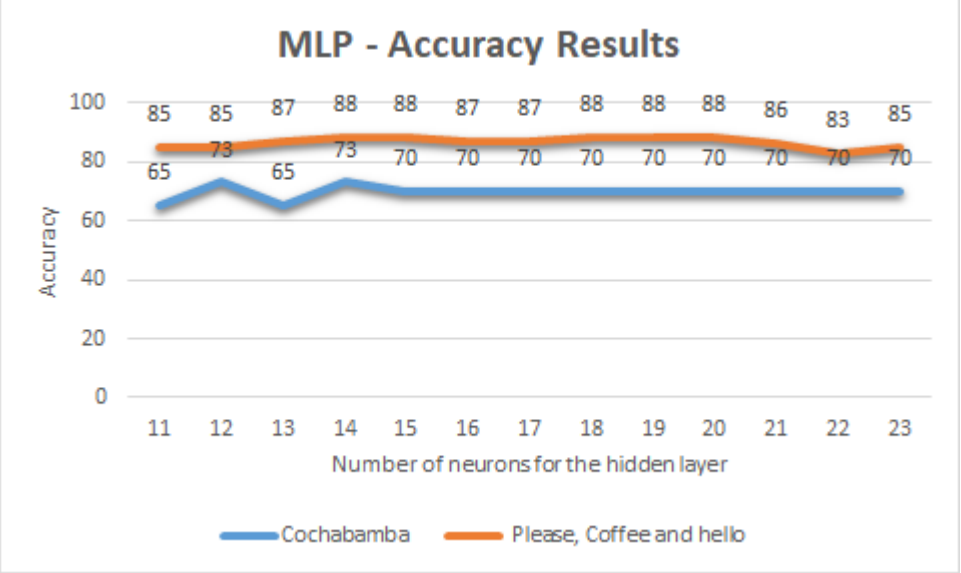

Figura 8 MLP accuracy results.

Observation 1. Our results show that MLP has values below 88% of accuracy for the signs under analysis, therefore it is not the appropriate machine learning technique for this problem.

5.2 CNN

The convolutional neural network follows a similar topology to MLP, but these contain one or more convolutional layers. The same tests as the MLP were performed for the set of gestures for 'Cochambamba' and 'please', 'coffee' and 'hello'. In this case we add a convolutional layer on the topology. Same as the previous test, in Figure 9 we can see the comparison of both types sets of signs.

Accuracy. - As we can see in Figure 9, all the results are above 95% accuracy. Compared to the results obtained by the MLP, we can confirm this type of network successfully learned the two sets of gestures. So we can affirm that CNN is very well suited to the problem of image classification.

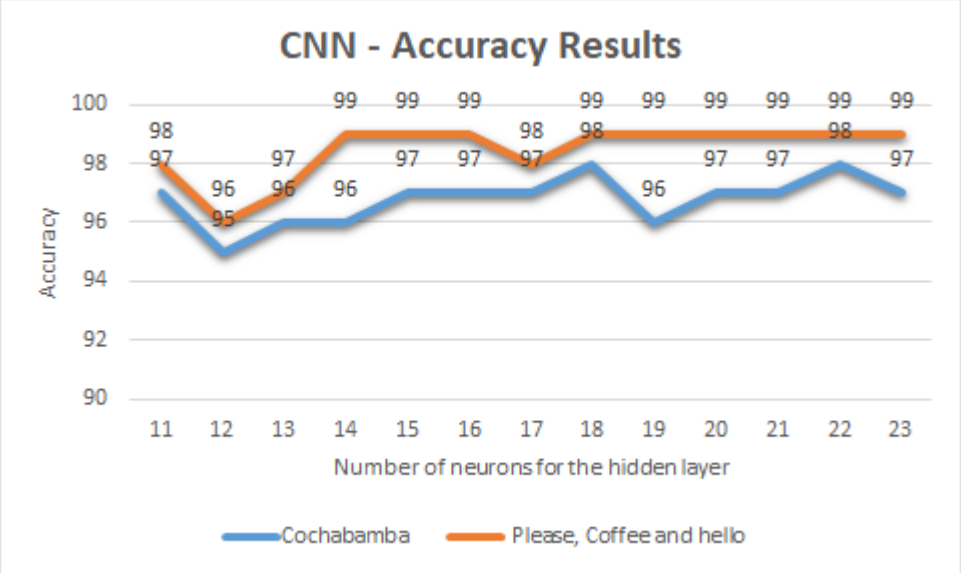

Figura 9 CNN accuracy results.

Observation 2. Our results show that CNN has values greater than 95% for the signs we consider, we can affirm that it satisfactorily classifies both sets of gestures and also CNN is better than MLP when in terms of image classification.

Recall. - In figure 10 we can see the results for the recall metric, the same previous tests were used. In the sign sets: 'Please', 'coffee' and hello. The first results, with fewer neurons in the hidden layer, were low. As the number of neurons increased they improved. Up to 99% was reached in recall, which is why we can affirm that the network is stable when making predictions using these sets of gestures as input.

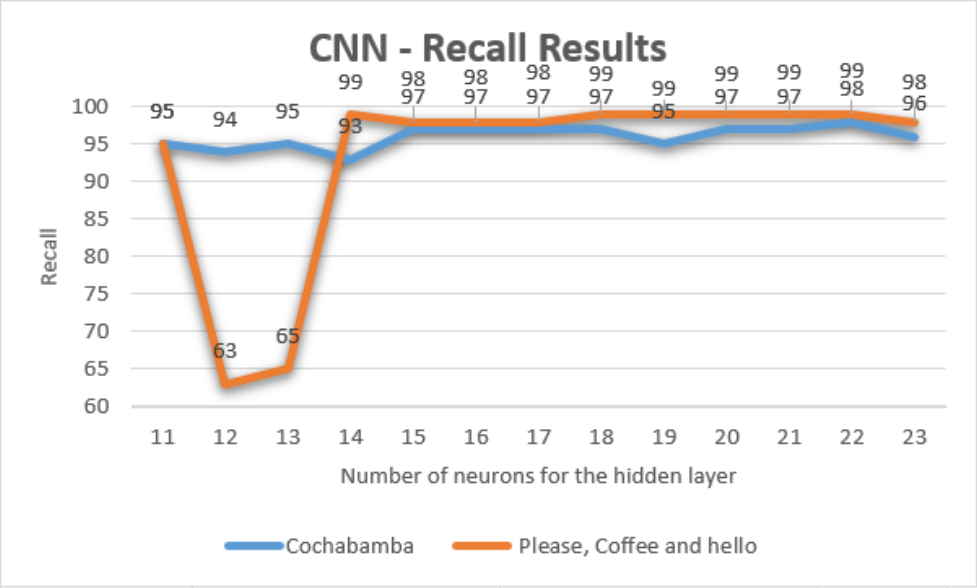

Figura 10 CNN recall results.

> Observation 3. Our results show that by increasing the number of neurons in the hidden layer the recall metric will improve using convolutional neural networks for the signs under analysis.

A new set of signs was organized with two objectives: To verify if CNN can learn two similar signs and to demonstrate that the increase in convolution layers improves the performance of the network. So, the signs were used: 'Please', 'coffee', 'hello' and 'want', where 'want' and 'coffee' have some similarities since in both the hands are on the torso. The results were the following:

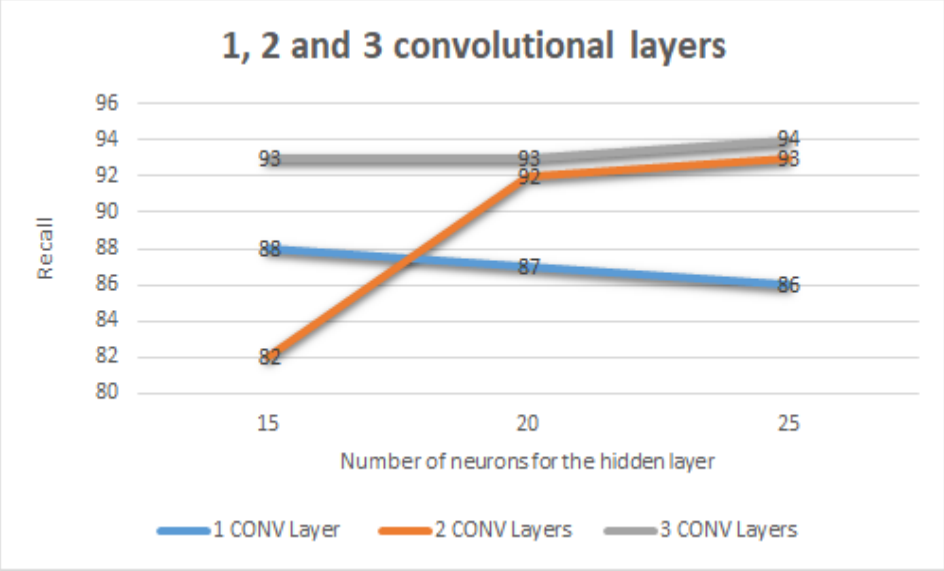

Figura 11 CNN recall results adding convolution layers.

As we can see in Figure 11, each of the tests, the one that gave the best results was a CNN with three convolution layers and a hidden layer of 25 neurons, so it can be stated that the increase in convolution layers can increase the performance of the network, adding more than three convolutional layers got the same results, so in this scenario the recall rate will get the best results with three convolutional layers. It is clear that CNN could classify different signs even if two of them have similar gestures, that is the case of 'want' and 'coffee', so we can confirm that this type of neural networks is efficient to solve these problems.

Observation 4. Our results show that by increasing convolution layers the metrics tend to improve and a convolutional neural network can learn different signs even if they are similar for the signs under analysis.

For the last test, we try to figure it out if the increase in convolution layers will increase the speed at which the network reaches its maximum point in terms of accuracy.

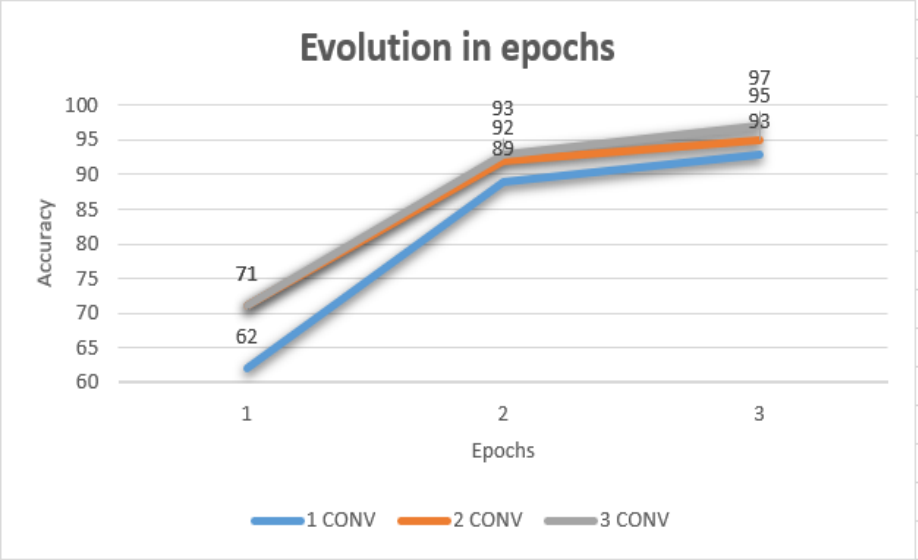

Figura 12 CNN evolution in epochs adding convolution layers.

Figure 12 shows that the greater the number of convolution layers the growth will be a little faster until reaching the maximum point, there could be the possibility in which the neural network got overstrained and the accuracy falls, although in this test low epochs were used to avoid that issue.

A pattern found was that the greater the number of convolution layers, the more neurons in the hidden layer were needed for the results to improve, otherwise if only one layer was used and more neurons were added than necessary, the results would begin to decrease. It is very possible that in case the trend is lowered, it will be necessary to increase convolution layers to change this.

Observation 5. Our results show that the increase in convolution layers make the neural network improve its performance and reach high numbers quickly in terms of accuracy, with a smaller number of neurons in the hidden layer.

The results of these last tests were entirely positive, demonstrating that the convolutional neural network manages to learn these signs using this type of preprocessing videos with the proposed descriptor, and new patterns were discovered that can be used in real-life projects.

6 Discussion and future work

Dataset. Although the videos filmed were used for the research, it is recommended that you take into account the following aspects when filming: Illuminated environments, the use of specialized tools such as professional cameras and tripods and clothing also backgrounds without textures or colors. Since this can influence the quality at the moment to use filters to obtain borders.

Descriptor. It necessary do more research about other possible descriptors that are more suited to real life problems such as: first option is emphasize points of interest such as arms, there is a classifier called Haar Cascade that is use to this type of task and the second option is the vectorization of the body, trying to analyze the key points if the movements, this would avoid the need for good conditions in the environment where the videos are filmed.

Machine learning. It is clear that CNN is a better option of artificial intelligence than MLP in this type of problems, but it's fair to say that it could be better to do more research and combine two types of neural networks to improve the results although with CNN positive results were obtained.

7 Related Work

American Sign Language recognition is not a new computer vision problem. Over the past two decades, researchers have used classifiers from a variety of categories that we can group roughly into linear classifiers, neural networks and Bayesian networks (Garcia & Alarcon).

While linear classifiers are easy to work with because they are relatively simple models, they require sophisticated feature extraction and preprocessing methods to be successful. For instance, Singha and Das obtained accuracy of 96% testing 10 different gestures that only need one hand using Karhunen-Loeve Transforms (Garcia & Alarcon). These types of transformations translate and rotate the axes to establish a new coordinate system based on the variance of the data. Karhunen-Loeve transformation is applied after using a skin filter, hand cropping and border detection on the images. This technique uses a linear classifier to distinguish between hand gestures including thumbs up, index finger pointing left and right, and numbers (no ASL). Sharma et al. use piecewise classifiers (Support Vector Machines and k-Nearest Neighbors) to characterize each color channel after background subtraction and noise removal. Their innovation comes from using a contour trace, which is an efficient representation of hand contours. They attain an accuracy of 62.3% using an SVM on the segmented color channel model (Sharma, Nemani, Kumar, & Kane, 2013).

Bayesian networks like Hidden Markov Models have also achieved high accuracies. These are particularly good at capturing temporal patterns, but they require clearly defined models that are defined prior to learning. Starner and Pentland used a Hidden Markov Model (HMM) and a 3-D glove that tracks hand movement. Since the glove is able to obtain 3-D information from the hand regardless of spatial orientation, they were able to achieve an impressive accuracy of 99.2% on the test set. Their HMM uses time series data to track hand movements and classify based on where the hand has been in recent frames (Starner & Pentland).

Suk et al. propose a method for recognizing hand gestures in a continuous video stream using a dynamic Bayesian network or DBN model (Suk, Sin, & Lee, 2010). They attempt to classify moving hand gestures, such as making a circle around the body or waving. They achieve an accuracy of over 99%, but it is worth noting that all gestures are markedly different from each other and that they are not American Sign Language.

Some neural networks have been used to tackle ASL translation (Garcia & Alarcon). Arguably, the most significant advantage of neural networks is that they learn the most important classification features. However, they require considerably more time and data to train. To date, most have been relatively shallow. Mekala et al. classified video of ASL letters into text using advanced feature extraction and a 3-layer Neural Network. They extracted features in two categories: hand position and movement. Prior to ASL classification, they identify the presence and location of 6 "points of interest" in the hand: each of the fingertips and the center of the palm. Mekala et al. also take Fourier Transforms of the images and identify what section of the frame the hand is located in. While they claim to be able to correctly classify 100% of images with this framework, there is no mention of whether this result was achieved in the training, validation or test set (Mekala, Gao, Fan, & Davari, 2011).

Admasu and Raimond classified Ethiopian Sign Language correctly in 98.5% of cases using a feed-forward Neural Network (Admasu & Raimond, 2010). They use a significant amount of image preprocessing, including image size normalization, image background subtraction, contrast adjustment, and image segmentation. Admasu and Raimond extracted features with a Gabor Filter and Principal Component Analysis.

The most relevant work to date is L. Pigou et al's application of CNN's to classify 20 Italian gestures from the ChaLearn 2014 Looking at People gesture spotting competition (Garcia & Alarcon). They use a Microsoft Kinect on full body images of people performing the gestures and achieve a cross-validation accuracy of 91.7%. As in the case with the aforementioned 3-D glove, the Kinect allows capture of depth features, which aids significantly in classifying ASL signs.

The difference between the experiment exposed with the related work is that we analyzed the Bolivian Sign Language and we include signs that use one and both

hands to be represented. We also chose to make a comparison between the effectiveness of the multilayer neural network with the convolutional neural network as machine learning techniques applied on this problem.

8 Conclusions

The obtained video set of people performing BSL was used for the generation of the different training and validation datasets for both neural networks, MLP and CNN. This video set contains four different signs and fifteen videos per sign, making a total of 60 samples. During the video production, a set of conditions was followed as much as possible: good scene illumination, a solid background without any objects or texture, just one person in the scene and this person has to be well-focused and centered in the video.

The combination of the border detection as descriptor and the preprocessing image techniques related to the motion detection in conjunction with artificial intelligence techniques such as CNN and MLP gave an accuracy which ranges between 65% and 88%, and CNN ranges from 95% and 99%, depending of the number of neurons and internal layers. Making this combination a good option for the image classification problems and, in consequence, for the BSL image-based recognition. As future work, we plan to evaluate these techniques using a bigger data set, including more complex signs.

Bibliography

[1] Admasu, & Raimond. (2010). Ethiopian sign language recognition using Artificial Neural Network.

[2] Buduma, N. *Fundamentals of Deep Learning, designing the next-generation machine intelligence algorithms*. Retrieved from http://perso.ens-lyon.fr/jacques.jayez/Cours/Implicite/Fundamentals_of_Deep_Learning.pdf

[3] Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986

[4] Garcia, & Alarcon. (2016). Real-time American Sign Language Recognition with Convolutional Neural. *Stanford University*.

[5] Goodfellow. (2016). *Deep Learning*. The MIT Press.

[6] Haykin, S. (2009.). Neural Networks and Learning Machines

[7] Hninn. (2009). Real-Time Hand Tracking and Gesture Recognition System Using Neural Networks.

[8] LeCun. (2019.). Quand la machine apprend: La révolution des neurones artificiels et de l'apprentissage profond.

[9] Mekala, Gao, Fan, & Davari. (2011). Real-time sign language recognition based on neural network architecture. *IEEE 43rd Southeastern Symposium on System Theory*. Auburn, AL, USA: IEEE.

[10] Ministerio de educacion boliviano. (2010). Modulo I - Curso de enseñanza de la lengua de señas boliviana.

[11] Núñez, C. P. (2017). Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition. Madrid, Spain: Universidad Rey Juan Carlos.

[12] Russell, N. *Inteligencia Artificial, un enfoque moderno*. Retrieved from https://luismejias21.files.wordpress.com/2017/09/inteligencia-artificial-un-enfoque-moderno-stuart-j-russell.pdf

[13] Sharma, Nemani, Kumar, & Kane. (2013). Recognition of Single Handed Sign Language Gestures using Contour Tracing Descriptor. *Proceedings of the World Congress on Engineering*. London, UK.

[14] Starner, & Pentland. (1996). Real-Time American Sign Language Recognition from Video Using Hidden Markov Models. *Massachusetts Institute of Technology* .

[15] Suk, Sin, & Lee. (2010). Hand gesture recognition based on dynamic Bayesian network framework. *ELSEVIER*.

[16] Yang, & Chen. (2020, June 29). Sign-to-speech translation using machine-learning-assisted stretchable sensor arrays. *Nature Electronics*.

[17] Pascual, J. A. (2014, June 21). Google Gesture, la voz de las personas sordomudas. *computerhoy*.

[18] PASTOR, J. (2013, 10 31). Kinect traduce el lenguaje de signos a lenguaje hablado. *Xataka*.