Static Peruvian Sign Language Classifier Based on Manual Spelling Using a Convolutional Neural Network

Gerardo Portocarrero-Banda[1], Eveling Gloria Castro-Gutierrez[1], Abdel Alejandro Portocarrero-Banda[1,2], Claudia Acra-Despradel[3], David Rondon[4], Hugo Guillermo Jimenez-Pacheco[2] and Miguel Angel Ortiz-Esparza[5*]

[1] *Universidad Nacional de San Agustín de Arequipa, Peru*
[2] *Universidad Católica de Santa María, Urbanización San José s/n, Arequipa, 04013, Perú*
[3] *Universidad Nacional Pedro Henríquez Ureña, Santo Domingo, Dominican Republic.*
[4] *Universidad Continental, Arequipa, Perú*
[5] *Center for Research in Mathematics, Quantum Knowledge City, Zacatecas, Mexico*

Abstract

There are great difficulties for people who suffer from mixed language disorders, having only one means for interpretive communication, sign language. A great challenge is to efficiently recognize these static gestures in real environments, therefore, the present research presents a convolutional neural network model that allows recognizing and classifying Peruvian Sign Language (PSL) with a dataset of 3025 frames through 4 stages: a) Generation of a dataset that involves 11 gestural components per frame, which involve invariant characteristics, sign parameters and gestural space, which allows greater generalization of the model compared to samples from other research b) Image preprocessing through the application of techniques and computer vision algorithms, c) Application of a convolutional neural network (CNN) model architecture and d) Execution of the model on a web platform to support model testing. The proposed CNN model obtained an accuracy rate of 99% in training, 88% in validation, and 84% in PSL recognition in the testing stage. The present model is better prepared to recognize static signs of the PSL in real scenarios.

Keywords

Peruvian Sign Languages, Convolutional Neural Network, Fingerspelling;

1. Introduction

Currently, in the social environment, there are communication barriers, accessibility, and equal opportunities for people with mixed language disorders. There are approximately 70 million people with hearing disabilities around the world [1] causing difficulties in their interaction, teaching and under- standing. The main means of communication for people with hearing disabilities is sign language. Sign language uses gestures to imitate or illustrate an object, feeling, expression, or even an action. Like spoken languages, sign language is unfortunately not universal. There is no single language that is shared by deaf people around the world. Several different sign languages have evolved independently across countries and even regions [2]. The last Peruvian census, in 2017, identified approximately 230,000 people with speech and hearing disabilities using the PSL [3][4]. Fingerspelling is the process of spelling (one letter at a time) words that do not have an existing sign, such as proper nouns such as a person's name, cities, products, etc. This method is carried out using the hand shapes associated with the letters of the

JINIS 2023: XXX International Conference on Systems Engineering, October 03–05, 2023, Arequipa, Peru
*Corresponding autor
gportocarrerob@unsa.edu.pe (G. Portocarrero); ecastrog@gmail.com (E. Castro-Gutierrez); abdel.portocarrero@ucsm.edu.pe (A. Portocarrero); c.acra@unphu.edu.do (C. Acra-Despradel); drondon@continental.edu.pe (D. Rondon); hjimenez@ucsm.edu.pe (H. Jiménez-Pacheco); miguel.ortiz@cimat.mx (M. Ortiz-Esparza)
0000-0002-2539-5294 (G. Portocarrero); 0000-0002-0203-041X (E. Castro-Gutierrez); 0000-0002-1050-2093 (A. Portocarrero); 0000-0002-6429-5675 (C. Acra-Despradel); 000000-0003-3506-5309 (D. Rondon); 0009-0001-5177-1126 (H. Jiménez-Pacheco); 0000-0001-8762-5780 (M. Ortiz-Esparza)
© 2023 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3693/paper17.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

static alphabet. However, in addition to hand joints, non-manual aspects such as facial expressions, arms, head, body movements and positions play a crucial role in static sign language [5]. However, most of the time, the hand gesture made by the dominant hand carries most of the meaning of the [6] sign. Therefore, an automatic software tool that can recognize alphabetic sign language symbols or gestures could have a great impact on the communication of individuals with mixed language disorder in society. Much of the current research on sign language recognition (RLS or LS) examines various advanced artificial intelligence methods. Specifically, convolutional neural network (CNN) techniques are applied as in different research [7][8] and different computer vision algorithms are used, such as color model change [9], segmentation and identification of image depth, in the work of [10] and the application of activation functions in CNN to enable multi- class classification, as described in the work of [11]. Knowing these aspects, this work proposes to implement a CNN model that allows identifying the largest number of PSL alphabets possible, with a percentage of precision similar to or greater than that investigated in the state of the art. The present work is organized in 6 sections, which are detailed as follows, section II describes the related works, section III contains the applied methodology, section IV describes the experimentation, section V presents the results and the discussion, and finally section VI presents the conclusions of the research and future work.

2. Related works

In the work of [12] in 2010 they report the implementation of an optical CNN system for the recognition of image patterns using neural networks for the identification of the sign language alphabet, concentrating on its recognition and classification , without considering the cultural system or identity of any specific community, 2 freely accessible datasets are also generated, one a) Dataset made up of 23 digital images and a b) Dataset made up of 23 images, which was generated by the authors of the article, both containing the same static signs of the alphabet. After processing, they obtained an average performance of 99% in recognition; However, the Second level sectioning CNN has difficulties in image recognition when applying a digital transform correlator responsible for discriminating patterns such as rotation and translation compared to their original position, which causes it to not be prepared for images that have not been trained and that have multiple components such as rotation or translation of gestures.

Ali Karami, Bahman Zanj and Azadeh Kiani Sarkaleh in 2011 show a focus on the recognition of static gestures from Persian Sign Language alphabets (PSLa) using a multilayer perceptron network [13], using a generated dataset using a digital camera, made up of 640 images of the bare hand gesture without the use of other resources such as gloves or visual marking systems. Achieving that the CNN obtains an average recognition accuracy of 94.06% of the PSLa alphabet, developing a robust system for the selected PSLa images, which only share the same characteristics of their dataset.

In the research of Lionel Pigou, Sander Dieleman, Pieter- Jan Kindermans, Benjamin Schrauwen in 2015, they identified communication difficulties between the society of people with hearing disabilities and the hearing society, thus implementing a language recognition system of Italian signs using Microsoft Kinect, the use of a CNN and GPU acceleration [14] and make available the open dataset of ChaLearn Looking At People 2014 (CLAP14), Track 3 (Gesture Spotting), which consists of 20 generated gestures by 27 users, making up a total of 6600 samples, 4600 dedicated to training, 2000 to validation and 3543 in tests that can be included in the validation or training set, considering different aspects such as complex backgrounds, clothing, lighting and gestural movements. In this research, a recognition percentage of 91.7% was obtained in the validation data and 95.68% in the test data, emphasizing that the test data were used in the training of the model.

Salem Ameen and Sunil Vadera in 2017 identified the number of individuals with hearing disorders and their ad- versities; therefore, their proposal consists of implementing an automatic tool for the interpretation of the static alphabet of American Sign Language (ASL) using a CNN. of two entries dedicated to the intensity and depth of images, aimed at classifying gestures through

manual spelling of the alphabet [15]. Having a free dataset of 60,000 images of the static alphabet of the LSA excluding the letters 'Y' and 'Z' that require movement for their gesticulation [16] [17], this dataset was generated by 5 different users, it is like this that achieve an average percentage of 82% accuracy in recognizing the static LSA alphabet. The authors compare their work with the research of Rioux-Maldague and Giguere [17] obtaining a higher percentage of recognition, and identifying two types of errors through a confusion matrix: (a) symmetric errors, such as two letters that can be classified erroneously with each other and (b) asymmetric errors, in which one letter is misclassified as another but not vice versa.

In 2017, it is proposed to recognize 24 classes of the PSL static alphabet through the development of two different CNN architectures (CNN1 and CNN2) with different amounts of layers and parameters per layer, fed according to the use of digital image processing techniques. to remove or reduce noise, improve contrast under varying lighting, separate the hand from the image background and finally detect and crop the region containing the hand gesture [18]. The dataset is generated by the researchers, made up of 16,200 images from 25 different users, including invariant characteristics in different sets of frames such as: scale, rotation, translation, lighting, noise and complex background. The performance of each CNN architecture is different, and the following results are obtained: a) CNN1 achieves an average of 95.37% in the recognition of test data (87.33% in the data with scale features, 94% in the data with rotation features, 91% in the data with translation features, 92.67% in the data with illumination features, 89% in the data with noise features and 84.44% in the data with background features complex) and an average of 99.34% in the validation, b) CNN2 achieves an average of 96.2% in the recognition of test data (93.67% on the data with scale features, 94.83% on the data with of rotation, 91.66% in the data with translational features, 93% in the data with illumination features, 89.33% in the data with noise features and 85.34% in the data with complex background features) and a average 99.73% in validation.

The research proposed by [19] highlights the infeasibility and impracticality of using devices or equipment such as Microsoft Kinect, since they are usually expensive, and cannot be used in controlled environments or with very specific re- quirements, for language recognition. address. For this reason, it was proposed to recognize the static alphabet of the LSA using machine learning algorithms: a) Support Vector Machine (SVM), b) K-Nearest Neighbors (KNN) and c) Random Forest (RF). Which were fed by handcrafted features of the images (histogram, Gabor filter and discrete wavelet transform). And they also used a CNN as a deep learning algorithm to compare their results. The prepared dataset is made up of 2524 images (only the gestural hand and random objects are considered) that cover 24 static LSA alphabets gestured by 5 users. Applying the YCbCr color model, they carry out the preprocessing: a) Segmentation of the hand and b) Extraction of the hand region (erosion and dilation). Making their CNN model reach the highest accuracy of 97.

Nikhil Kasukurthi, Brij Rokad, Shiv Bidani and Aju Dennisan in 2019 proposed a deep learning model called Squeezenet for the recognition of the LSA alphabet so that it can be executed on mobile devices [20]. Considering a freely available dataset with a set of 43986 RGB (320x320 pixels) finger images from Surrey [21]. The arranged model achieves 87.47% accuracy in recognizing training data, and 83.29% in validation. Thus, obtaining a model that allows predicting sign language in real time, thus being a squeezenet architecture that is executed completely on a mobile device accessible to the end user. They are still investigating improving aspects such as lighting and the distance of the sign from the capturing device.

Amirhossein D, Alireza T, Maryam T and Majid M in 2019 propose a two-stage CNN architecture for robust hand gesture recognition, called HGR-Net, where the first stage performs precise semantic segmentation to determine the regions of the hands and the second stage identifies the gesture. Its experimentation is focused on using public datasets [22], achieving a recognition percentage of 81.1% with the HGR- Net model, occupying a size of only 2.4MB, which allows it to be flexible in its portability to any type of digital environment, and in its execution, achieving an average of 23ms with 43fps in the recognition of each gestural input. In the work of Ankita Wadhawan and Parteek Kumar in 2020, they propose a CNN model that allows the recognition of static gestures of Indian Sign Language (LSI) through in- depth experimentation

and comparison of 50 CNN models. Taking a free access dataset made up of 35,000 RGB images (350 images for each sign), which consist of various sizes, colors and taken in different environmental conditions to help in the best generalization of the classifier. Reaching 99.72% and 99.90% in the recognition of color and grayscale training images respectively. Demonstrating greater precision compared to other identifying systems: a) KNN (95.95%), b) SVM (97.9%) and c) ANN (98%) in the recognition of static signs of the LSI with training data [23].

3. Methodology

This section describes the stages of the proposed method, which is represented illustratively in Figure 1, these stages include the generation of a dataset of PSL alphabets, con- tinuing with the preprocessing of the dataset that is used as input data (static sign of the PSL), which allows adjusting and modifying each frame that makes up the dataset to correctly feed the CNN model through the following stages that include the training and validation of the architecture of the CNN model, continuing with the tests and adjustments thereof, to achieve correct recognition of the PSL in the execution stage of the model on a local web platform.

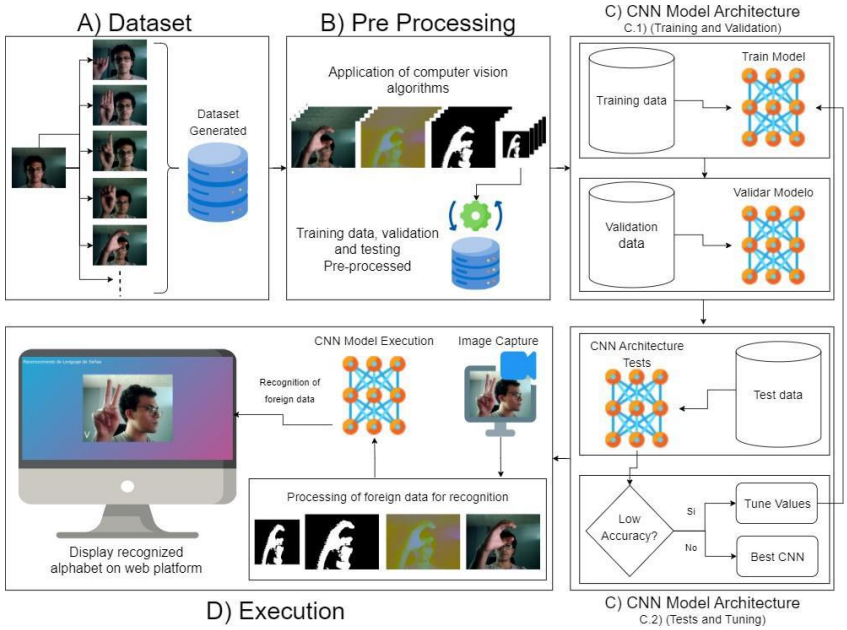

Figure 1: he phases and stages of the proposed method are detailed or specified: A) Generation of a Dataset, B) Preprocessing, C) CNN Model Architecture (C.1. Training-Validation, C.2. Testing-Adjustments), D. Execution.

3.1. A) Generation of a Dataset

It is made up of the generation of 121 frames for each static alphabet of the PSL. Which is divided into 3 parts, training data, validation and test data (63%, 7% and 30% respectively). 24 PSL alphabets were considered plus a class labeled 'nothing' that represents the frames that do not correspond to any gesture of the PSL alphabet, with the purpose of differentiating any external gesture that does not belong to the PSL, as shown in Figure 2. A total of 3025 frames were produced across all PSL and non-PSL classes. Therefore, 1906 frames are allocated for model training, 211 frames for model validation and 908 frames for evaluating the percentage of recognition accuracy of test data.

The data itself maintains certain very important aspects to consider, gestural diversity is manifested in different perspectives and environments, and therefore it is optimal to cover all the components of a signing gesture. It contemplates: a) 11 components of the gesture, which make up 7 invariant characteristics (day, afternoon and night lighting, noise or atypical aspects,

complex background, translation and scaling) [18], b) 03 parameters of the sign (hand shape, hand orientation and location) [24], and c) the gestural space, which is the rectangular view of the user's upper torso. The shape of the hand is the configuration that the hand assumes when it begins to make the sign. Orientation is the direction in which the hand turns, and location is the formation of the sign near the signer's body. Around 75% of all signs are formed in the head and neck areas, as they can be identified better and are closer to reality.

3.2. B) Pre-processing

The preprocessing stage works on the generated dataset, its objective is to determine the most important characteristics of each frame. For this preprocessing task, different filters such as techniques and transformations are applied, which are mentioned below; 1) color model change, 2) skin color detection, 3) resolution change and 4) normalization.

1) Color model shifting: The color model (RGB to YCbCr) is modified in each frame to allow the application of the appropriate filters to identify skin color, as it is applicable to complex color images with uneven lighting [18][25], where 'Y' is the luminance, 'Cb' and 'Cr' mean blue and red respectively, are collectively called as color components.

The YCbCr color space has the characteristics of separating chromaticity and brightness [9]. When the frame includes different aspects, in addition to the person performing the gesture, we choose to use skin color segmentation using the YCbCr [25] color model for the segmentation of the bare hand as seen in Fig. 3.

Therefore, the RGB color space is separated into the luminance and chroma components, obtaining a YCbCr color space. The 'Y' component belongs to luminance. The components 'Cb' and 'Cr' belong to the color difference chroma. The components 'Y', 'Cb', 'Cr' can be obtained from the RGB values [19] using Eq. 1. The prime symbol indicated by the average gamma correction is used. Gamma correction controls the overall brightness of an image. R', G', and B' nominally vary from 0 to 1, where 0 represents the minimum intensity and 1 the maximum.

$$Y' = 16 + (65.481 \cdot R' + 128.553 \cdot G' + 24.966 \cdot B')$$

$$C_B = 128 + (-37.797 \cdot R' - 74.203 \cdot G' + 112.0 \cdot B')$$

Figure. 2: Dataset made up of frames of static gestures with bare hands according to the PSL alphabet: A) Considering 11 components of the gesture (7 invariant characteristics, 3 parameters of the sign and the gestural space), B) Frames that do not belong to any gesture of the alphabet of the PSL (ex: a pencil).

2) Skin color detection: Once the skin color range is defined according to the YCbCr model, all the sectors that coincide are highlighted so that only these characteristics can be observed within the environment that, in most cases, involve different complex aspects which can hinder the information. After obtaining the YCbCr color space values for each pixel of the image in Eq. 1, each pixel was classified as skin pixel or nonskin pixel. If the values of 'Y', 'Cb' and 'Cr' for a pixel in the image lie in the ranges mentioned in Eq. 2, then that pixel is a skin pixel, otherwise it is a non-skin pixel [19].

$$0 < Y < 255 \quad (2)$$
$$133 < CB < 173$$
$$77 < CR < 127$$

These operations are applied to all frames of the dataset after changing the color model to YCbCr, and in this case, highlighting the daylight invariant characteristics in Fig. 4

Figure 4: A) Frame with the YCbCr color model, B) Frame highlighting the sectors that have the skin color.

As observed in each of them, the necessary aspects can be noted to cover both the invariant characteristics, the parameters of the sign and the gestural space. Now, the reason for not maintaining skin color and maintaining black and white frames is precisely because of the types of components that were mentioned above, one of them is noise, which prevents recognition efficiently, if these types of aspects is found in the frames of the dataset, such as a user's glasses, or possibly rings, bracelets, or objects in the background, which are recognizable if they remain and thus spoil the recognition, because the only element What should be recognized are the static gestures of the PSL alphabet.

3) Resolution Change: All frames are modified and resized to have a resolution of 224x224 pixels [22][26][27][28], which further allows only the essential components of the gesture to be extracted as shown in Figure 5.

4) Normalization: All frames are normalized, dividing the value per pixel by 255, since it is the maximum value that a pixel can have, obtaining values in the range from 0 to 1, with the aim of having a lighter dataset that can be managed more easily, it is done using the following Eq. 3.

$$\text{Pixel Value} = \{0....1\} \quad (3)$$

$$R = 128 + (112.0 \cdot R' - 93.786 \cdot G\text{-}18.214 \cdot B')$$

Figure 5: A) Frame with resolution of 640x480 pixels. B) Frame output with224x224 pixels resolution change applied.

3.3. C) CNN Model Architecture

The proposed system is designed to recognize static alphabets of the PSL, to achieve this the architecture of the CNN was defined according to the number of classes (alphabets) to be identified, for this different but related stages were established through training, validation, model testing and adjustments.

The CNN model has 4 convolution layers (32, 64, 128 and 256 filters respectively) applying a 3x3 kernel, and a ReLU activation function per convolution layer, 4 MaxPooling layers between the convolution layers applying a kernel of 2x2, adding a dropout layer with a random node exclusion rate of 30% [29] in the last convolution layer, to avoid overtraining.

A flattening layer is also defined, which allows the input layer to be delivered to the fully connected neural network. For the fully connected neural network, 1 input layer, 3 hidden layers and an output layer (256, 128, 64, 32 and 25 nodes respectively) and a ReLU activation function in the first 4 layers are defined, defining in the output layer the Softmax activation function to allow multi-classification, producing a probabilistic percentage of recognition for each class, giving the highest percentage to the class with the highest probability rate in the validation of the model, it works as an early stop and distributing the rest to the other classes with the least assertive probability, also adds a dropout layer with a random node exclusion rate of 30%, which can be seen in Figure 6.

The experimentation was carried out on a dedicated graphics card (GPU), as well as used in the different proposals [14][18][20][22][23][27][28][30], Nvidia GeForce GTX 1650Ti with 4GB VRAM capacity together with a 4-core CPU at 2.50 GHz to 4.50 GHz with 16 GB RAM, for the use of Different configurations of the GPU were made both in the basic entry system (BIOS) and in the operating system and the use of official resources under Cuda 11.8 with cuDNN 8.2 in TensorFlow 2.10 with a stable version of Python 3.8. 1) Training and Validation: The training is executed using the “categorical cross entropy” loss function and the “ADAM” optimizer with a learning rate of 0.001. The maximum number of batches for each training step varies depending on the amount of data, as well as the iterations per batch, and a callback was implemented that allows identifying the accuracy

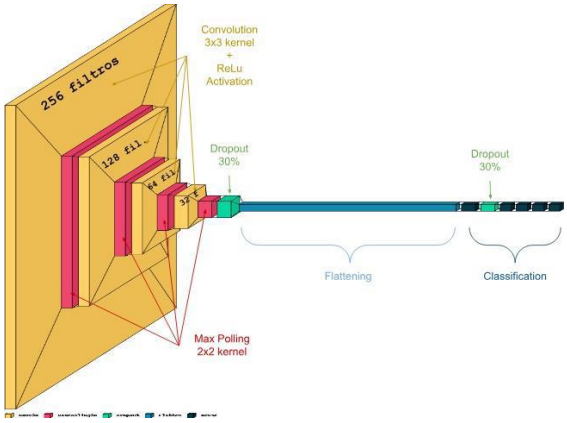

Figure 6: 4-layer CNN, a dropout layer, a flattening layer, a fully connected network with 1 input layer, a dropout layer, 3 hidden layers and 1 output layer with their respective kernels set.
which allows you to save the batch that was best trained and run the most optimal model [22].

2) Tests and Tuning: For the tests, the test dataset is being used; every time a modification is made to the dataset or to the model architecture, it is necessary to perform tests to obtain the percentage of accuracy of the model, and ensure its optimization through adjustments, which allows adjust certain model training values.

To determine the moment of need for these changes, an algorithm was implemented that functions as feedback and a key point to make adjustments, which applies the following Eq. 4, and returns the matches (a match is the output value of the recognition of each frame of the test dataset if correct) as a percentage between the recognition and the labeled classes for each frame, obtaining the indicator accuracy of the recognition percentage of the test data.

$$\sum_{i=1}^{n} \frac{\text{Matches} \cdot 100}{\text{LabeledClasses}} \tag{4}$$

D. Execution on a Web Platform

To improve testing, in a real-time environment, a web platform has been developed following an MVC (model-view- controller) architecture implemented in Django due to its capabilities for the execution of the previously trained, validated and tested CNN model, which will have the functionality to capture images through the user's device so that it can be delivered to the CNN model and perform the recognition, which will be displayed on the screen.

Its use is intended to be simple and understandable, precisely so that any user has the ability to use it, so it is only necessary to have a laptop or computer with a photo sensor or webcam and the project locally. As shown in Figure 7, a distinction is made between what the user observes, to which the platform is predestined, and what happens internally, each captured frame is sent for processing, as described in the previous stages, to be delivered to the model and perform the recognition, in this case, the identified alphabet is 'A' of the PSL, superimposing the sign in text at the bottom left of the frame, achieving a satisfactory recognition of the sign.

a) b)

Figure 7: A) Execution of CNN model recognizing the static alphabet 'A' of the PSL on the web platform, visualized by the user, B) Preprocessing of the frame delivered to the CNN model for recognition of the static alphabet 'A' of the PSL on the web platform, not visualized by the user.

To mention additionally, observe in the figure 8 how the frame is captured and the trained model interprets when no PSL sign is being gestured and places the text "nothing" at the bottom left.

a) b)

Figure 8: A) Execution of CNN model recognizing no PSL alphabet on theweb platform, viewed by the user. B) Preprocessing of the frame deliveredto the CNN model for recognition of any PSL alphabet on the web platform,not viewed by the user.

4. Experimentation

Stage that allows establishing and adjusting the parameters and functions of both the preprocessing and the CNN model according to numerous executions. For this, 04 different configuration stages were required in which measurements were obtained according to the average percentage of validation and testing accuracy of the model, its effectiveness when performing gesture recognition.

4.1. A. First experimentation

Made when considering a dataset of 1200 frames, which make up the alphabets from 'A' to 'L', excluding the alphabet 'J' because it is not considered a static sign, and also consid- ering the creation of the class ' NADA' that represents any alphabet of the LSP, and in this way contrast when a sign is not being gestured, therefore, the different configurations for this dataset are found in the table I.

4.2. B. Second Experimentation

Performed when a dataset of 1465 frames was contemplated, which make up the alphabets 'A' through 'R', excluding the alphabet 'J' and 'N˜ ' because they are not considered as static signs, thus, the different configurations for this dataset are found in the table II.

Table 1
Use of various configurations for experimentation and evaluation of the model with a 1200 frame dataset.

Steps per epoch	% Training	% Validation	% Test
10	99.33%	79.76%	**79.44%**
30	**99.61%**	**83.33%**	77.22%
45	99.07%	77.38%	74.16%

Table 2
Use of various configurations for experimentation and evaluation of the model with a dataset of 1465 frames.

Steps per epoch	% Training	% Validation	% Tests
10	**99.13%**	84.47%	78.18%
30	98.69%	**89.32%**	**81.13%**
45	99.04%	86.40%	79.54%

4.3. C. Third Experimentation

Performed when 100 frames per alphabet were set to the dataset which are 'S', 'T', 'U', 'V', 'W', 'X', 'Y', excluding the alphabet 'Z' because it is not considered as a static sign, therefore, a dataset of 2500 frames are contemplated in its totality, these configurations are found in the table III.

Table 3
Use of various configurations for experimentation and evaluation of the model with a 2500 frame dataset.

Steps per epoch	% Training	% Validation	% Tests
15	**98.92%**	88.57%	82.26%
30	98.61%	87.99%	82.00%
45	98.15%	**89.14%**	**83.47%**

4.4. D. Fourth Experimentation

This was done by adding 21 additional frames per class to the dataset, for a total of 3025 frames in its entirety, but for this change to be viable, the number of epochs was adjusted from 30 to 75, and its other settings are found in the table IV.

All the experimentation was carried out using all the hard- ware resources (GPU, CPU, Storage) and it was possible to shorten the training time with an impressive difference using the GPU with respect to the CPU, having an average difference of 1500%, as shown in Table V.

Table 4
Use of various configurations for experimentation and evaluation of the model with a 2500 frame dataset.

Steps per epoch	% Training	% Validation	% Tests
30	**99.68%**	87.26%	83.59%
45	99.56%	**87.90%**	**84.40%**
60	99.42%	87.60%	83.48%

Table 5
Time spent training and experimenting with the CNN model.

GPU	CPU
113.1651 seconds (1.9 minutes)	1758.1821 seconds (29.3 minutes)

5. Results

In the present investigation, a dataset of 3025 frames was generated, and all the factors of the gesticulation of the signs in each image were incorporated. Each of the frames was processed to be input and trained to the CNN model. Thus, it can be observed that in the different compositions of the CNN model configurations, there are very narrow differences, how- ever, these disparities establish the most optimal CNN model possible. When 10 iterations per batch is defined, it delivers a (99, 84, 79)% average recognition accuracy of the PSL in the training, validation and testing stages, with 30 iterations per batch it is obtained (98, 88, 82) % precision, respectively, with 45 iterations per batch, (98, 89, 83)% precision respectively was obtained, and the last adjustment made, increasing the number of batches from 30 in the first 3 experiments, and 75 batches in the fourth experiment, obtaining (99, 88, 84)% respectively. It is taken into account that with fewer iterations a higher percentage of training precision is obtained, however, this factor is not critical like validation and testing.

Which indicates that the number of frames that have to be trained per batch should be as small as possible, so the 45 iterations per batch were maintained, and consequently a 99% accuracy in training was achieved. an average of 88% accuracy in validation and an average of 84% accuracy in recognition of test data.

This research work is compared with the authors' research works, which are shown in Table VI. An example of the images that make up each dataset can be seen in Fig. 9.

Flores et al. [18] and Dadashzadeh et al. [22] worked with similar conditions in terms of lighting during the day, afternoon and night, also with noise, complex background, translation and scaling, achieving an accuracy percentage of 95.37% and 88.1% respectively in the stage of

evidence, it should be noted that Dadashzadeh et al. [22] added hand shape and location, adding these components of the gesture affects the accuracy percentage.

Table 6
Comparative results in the literature. where "A" is the present investigation.

Ref	Year	Neural Network Model	Components	Accuracy (Test Stage)
[18]	2017	CNN	7	95.37%
[11]	2018	CNN	6	92.88%
[10]	2018	CNN	9	73.40%
[22]	2019	CNN	8	88.10%
[31]	2020	CNN	7	96.20%
A	2023	CNN	11	84%

Figure 9: Comparison of frames that conform different elements of the gesture considered by authors and the present research.

[11], [10] and [31] worked between 6 and 9 frames with similar conditions in terms of translation and scaling, however [11] and [31] added the form of hand and [10] the complex background, under these conditions they obtained 92.88%, 96.2% and 73.4% precision in the testing stage, as can be seen by working with more frames in this case 9 for the investigation [10] also affects the percentage of precision, decreasing its value compared to other investigations.

The research presented worked with a dataset made up of 11 components of the gesture composed of: 07 invariant characteristics, 03 sign parameters, and 01 gestural space, also considering these additional components for each frame in the dataset, 84% precision is obtained in the testing stage.

From the analysis of Table V, we can establish that when more components are added to these frames, the recognition task becomes more complex and a trend of decreasing percent- age of accuracy is observed in the testing stage, therefore 84% The precision achieved in our proposal represents an advance compared to the closest research, a model that was trained with 9 components and the authors obtained 73.4%.

6. Conclusions and future work

There is a difficulty in finding systems that recognize PSL static sign gestures with high accuracy. The present research uses a dataset of 3025 frames that contain 11 gesture components, which consider different aspects of real environments, which was trained in the CNN model. The CNN model is configured as follows: 04 convolution layers, 04 max pooling layers, 01 dropout layer, 01 flattening layer, 01 fully connected network with 01 input layer, 01 dropout layer, 03 hidden layers and 01 output layer.

This configuration allowed us to achieve an accuracy of 84%. It was concluded that to have a greater range in real scenarios, precision has to be sacrificed. Although it is true, the accuracy percentage of our proposal is not the highest in the literature, but it is better prepared to recognize static signs of the PSL in real scenarios.

Digital image processing techniques were used, these techniques helped to better detect the region containing the hand gesture, minimizing the error caused by the gestural components.

As future work, we plan to continue training a model that is capable of recognizing static and continuous signs of the PSL in real time.

References

[1] World Federation of the Deaf. (2016). Advancing human rights and signlanguage worldwide.

[2] X Glottolog. (2022). Pseudo family: Lenguaje de sen˜aspara sordos. Retrieved 2022-10-31, fromhttps://glottolog.org/resource/languoid/id/deaf1237

[3] X Glottolog Peru. (2022). Pseudo family: Lenguaje desen˜as peruana. Retrieved 2022-10-31, fromhttps://glottolog.org/resource/languoid/id/peru1235

[4] DDHH. (n.d.). Defensoria del pueblo: debe facilitarse el aprendisaje dela lengua de sen˜as peruana y promover la identidad linguistica y culturalde las personas sordas. 2022.

[5] Crasborn, O. A. (2006). Nonmanual Structures in Sign Language. InEncyclopedia of Language & Linguistics (pp. 668–672). Elsevier.https://doi.org/10.1016/b0-08-044854-2/04216-4

[6] X Ding, L., & Martinez, A. M. (2007). Recovering the linguisticcomponents of the manual signs in american sign language. In 2007ieee conference on advanced video and signal based surveillance (pp. 447– 452).

[7] Ashrafuzzaman, M., Saha, S., & Nur, K. (2022). Prediction of Stroke Dis-ease Using Deep CNN Based Approach. Journal of Advances in Informa-tion Technology, 13(6), 604–613. https://doi.org/10.12720/jait.13.6.604-613

[8] Al-Dmour, H., Tareef, A., Alkalbani, A. M., Hammouri, A., & Alrahmani,B. (2023). Masked Face Detection and Recognition System Based on DeepLearning Algorithms. Journal of Advances in Information Technol- ogy,14(2), 224–232. https://doi.org/10.12720/jait.14.2.224-232

[9] Jiang, X., Satapathy, S. C., Yang, L., Wang, S. H., & Zhang, Y. D. (2020).A Survey on Artificial Intelligence in Chinese Sign Language Recogni-tion. Arabian Journal for Science and Engineering, 45(12), 9859–9894.https://doi.org/10.1007/s13369-020-04758-2

[10] Li, Y., Wang, X., Liu, W., & Feng, B. (2018). Deep atten- tionnetwork for joint hand gesture localization and recognition using staticRGB-D images. Information Sciences, 441, 66–78. https://doi.org/10.1016/j.ins.2018.02.024

[11] Rao, G. A., Syamala, K., Kishore, P. V. V, & Sastry, A. S. C. S. (2018).Deep Convolutional Neural Networks for Sign Language Recognition.

[12] Vargas, L., Barba, L., & Mattos, L. (2010). Identification System of theSign Language Using Artificial Neural Networks.

[13] Karami, A., Zanj, B., & Sarkaleh, A. K. (2011). Persian sign language(PSL) recognition using wavelet transform and neural networks. ExpertSystems with Applications, 38(3), 2661–2667. https://doi.org/10.1016/j.eswa.2010.08.056

[14] Bronstein, M. M., Agapito, L., & Rother, C. (2015). Sign LanguageRecognition Using Convolutional Neural Networks. In Lecture Notesin Computer Science (including subseries Lecture Notes in ArtificialIntelligence and Lecture Notes in Bioinformatics) (Vol. 8927, p. VI). Springer Verlag. https://doi.org/10.1007/978-3-319-16178-5

[15] Ameen, S., & Vadera, S. (2017). A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images. Expert Systems, 34(3). https://doi.org/10.1111/exsy.12197

[16] X Pugeault, N., & Bowden, R. (2011). Spelling it out: Real- time ASL fingerspelling recognition. 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). doi:10.1109/iccvw.2011.6130290

[17] Rioux-Maldague, L., & Giguere, P. (2014). Sign language fingerspelling classification from depth and color images using a deep belief network. Proceedings - Conference on Computer and Robot Vision, CRV 2014, 92– 97. https://doi.org/10.1109/CRV.2014.20

[18] Flores, J. L., Cutipa, G., & Enciso, L. (2017). Application of Convo- lutional Neural Networks for Static Hand Gestures Recognition Under Different Invariant Features.

[19] Ranga, V., Yadav, N., & Garg, P. (2018). AMERICAN SIGN LAN- GUAGE FINGERSPELLING USING HYBRID DISCRETE WAVELET TRANSFORM-GABOR FILTER AND CONVOLUTIONAL NEURAL NETWORK. In Journal of Engineering Science and Technology (Vol. 13, Issue 9).
[20] Kasukurthi, N., Rokad, B., Bidani, S., & Dennisan, A. (2019). American Sign Language Alphabet Recognition using Deep Learning.
[21] Zimmermann, C., & Brox, T. (2018). Learning to Estimate 3D Hand Pose from Single RGB Images. https://lmb.informatik.uni- freiburg.de/projects/hand3d/
[22] Dadashzadeh, A., Targhi, A. T., Tahmasbi, M., & Mirmehdi, M. (2019). HGR-Net: A fusion network for hand gesture segmentation and recog- nition. IET Computer Vision, 13(8), 700–707. https://doi.org/10.1049/iet- cvi.2018.5796
[23] Wadhawan, A., & Kumar, P. (2020). Deep learning-based sign language recognition system for static signs. Neural Computing and Applications, 32(12), 7957–7968. https://doi.org/10.1007/s00521-019-04691-y
[24] Tennant, R. A., Brown, M. G., & Nelson-Metlay, V. (1998). The American Sign Language handshape dictionary.
[25] Shaik, K. B., Ganesan, P., Kalist, V., Sathish, B. S., & Jenitha, J. M. M. (2015). Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space. Procedia Computer Science, 57, 41–48. https://doi.org/10.1016/j.procs.2015.07.362
[26] Slimane, F. Ben. (2020). SIGN LANGUAGE RECOGNITION AND TRANSLATION.
[27] Rathi, P., Gupta, R. K., Agarwal, S., Shukla, A., & Tiwari, R. (2020). Next Generation Computing Technologies. https://ssrn.com/abstract=3545064
[28] Camgoz, N. C., Hadfield, S., Koller, O., & Bowden, R. (2017). Sub- UNets: End-to-End Hand Shape and Continuous Sign Language Recogni- tion. Proceedings of the IEEE International Conference on Computer Vi- sion, 2017-October, 3075–3084. https://doi.org/10.1109/ICCV.2017.332
[29] X Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhut- dinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929–1958.
[30] X Bantupalli, K., & Xie, Y. (2018). American sign language recognition using deep learningand computer vision. In 2018 ieee international conference on big data (big data) (pp. 4896–4899).
[31] Neethu, P. S., Suguna, R., & Sathish, D. (2020). An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks. Soft Computing, 24(20), 15239–15248. https://doi.org/10.1007/s00500-020-04860-5