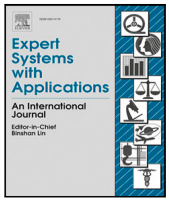
Expert Systems With Applications 215 (2023) 119394

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Automatic translation of sign language with multi-stream 3D CNN and generation of artificial depth maps

Giulia Zanon de Castro [a,b], Rúbia Reis Guerra [a], Frederico Gadelha Guimarães [a,*]

[a] Machine Intelligence and Data Science Laboratory (MINDS), Department of Electrical Engineering, Universidade Federal de Minas Gerais, Av. Antonio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil

[b] Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil

ARTICLE INFO

Keywords:
Sign language recognition
Gesture recognition
Deep learning
3D convolutional neural networks
Computer vision
Generative adversarial networks

ABSTRACT

Sign languages play an essential role in the cognitive and social development of the deaf, consisting of a natural form of communication and being a symbol of identity and culture. However, hearing loss has a severe social impact due to an existing communication barrier, preventing access to essential services such as education and health. A bi-directional sign language translation may be the solution to bridging the communication gap between the deaf and the listener, completing a two-way communication cycle. Virtual personal assistants can benefit from this technology by extending how users interact with the intelligent system. With this idea, in this work we develop a multi-stream deep learning model to recognize signs of Brazilian (BSL), Indian (ISL), and Korean (KSL) Sign Languages. We combine different types of information for the classification task, using single-stream and multi-stream 3D Convolutional Neural Networks. In addition, considering the largest source of sign data globally – the internet – we propose a depth sensor-free classification method, with depth maps artificially generated through Generative Adversarial Networks. In order to consider the main parameters that encode sign languages, the final architecture is composed of a multi-stream network that receives the segmented hands, the faces, the distances and speeds of the points of articulation, and the RGB frames associated with artificial depth maps. Finally, we provide a visual explanation to understand which regions were important for model decision-making. The best models were obtained using the multi-stream network, presenting an accuracy of 0.91 ± 0.07, and f1-score of 0.90 ± 0.08 on publicly available BSL data set. The results suggest that the multi-stream network with artificially generated depth maps is suitable for the task of sign recognition in different languages.

1. Introduction

Most software available today is designed around text-based or speech-based interfaces. Voice assistants, for one, have been around for the past decade, introducing the public to a novel way of interacting with technology. Today, a person can place an online order at a restaurant, request directions to a location, or even call a loved one without touching their devices during the process. A vast increase in smartphone's computing power and the extensive amounts of linguistic data have facilitated the development of highly successful machine learning models capable of interpreting human language (Hoy, 2018). However, similar advancements are yet to be seen regarding deaf and hard-of-hearing communities, whose primary communication channel relies on sign language.

Sign language is a form of communication composed of visual–motor cues to represent concepts and ideas. Each sign involves a combination of manual parameters — such as palm orientation and hand configuration – and non-manual parameters — such as facial expression and shoulder positioning. There are over 70 million deaf people worldwide and 300 sign languages, each with its own structure and lexicon.[1] Although there is currently no reliable estimate of how many people are fluent in sign language worldwide (Meulder, 2019), multiple studies have pointed out the impact of a communication barrier between deaf and hearing communities (Bai & Bruno, 2020; Barnett et al., 2011; Kushalnagar, Moreland, Simons, & Holcomb, 2018; Santos & Portes, 2019).

One solution to mitigate the communication barriers is developing automatic translation software. Ideally, a sign language recognition tool would enable effective means of communication and personal expression among deaf and hearing people, when third parties (for example,

* Corresponding author.
E-mail addresses: giuliaz@ufmg.br (G.Z. de Castro), rubiarg@cs.ubc.ca (R.R. Guerra), fredericoguimaraes@ufmg.br (F.G. Guimarães).

[1] World Federation of the Deaf. Available at https://wfdeaf.org/our-work/.

https://doi.org/10.1016/j.eswa.2022.119394
Received 19 January 2021; Received in revised form 17 August 2022; Accepted 30 November 2022
Available online 2 December 2022

sign language interpreters) may not always be present. However, translating signs to text poses several open issues. From a computational standpoint, capturing the visual–temporal nature of the languages is a complex task. Unlike other languages, signs are not clearly delineated in a sentence, binding into each other in a process denominated "coarticulation" (Tyrone & Mauk, 2010). Many sign languages around the world are not documented. The sheer amount of data required for training Machine Learning (ML) models hinders the development of more robust automatic translation techniques. Finally, some engineering solutions being researched fail to include deaf individuals in the development loop, creating end products that are centered on the communication needs of hearing individuals. This renders the conversation flow between signalers and non-signalers one-sided (Bragg et al., 2019; Forshay, Winter, & Bender, 2016).

Numerous sign language recognition models available today rely on wearables or RGB-D sensors (Bragg et al., 2019; Cheok, Omar, & Jaward, 2019), which hampers the application of the tool in settings where the user does not have access to such artifacts. Moreover, more affordable RGB-D sensors, such as Leap Motion Controller and Microsoft Kinect, suffer when providing a reliable and precise location of finger articulation points, which are helpful when distinguishing amongst signs with similar motions but different hand configurations (Guzsvinecz, Szucs, & Sik-Lanyi, 2019; Marin, Dominio, & Zanuttigh, 2014). From the algorithmic standpoint, promising advancements have been achieved using Hidden Markov Models (Kumar, Gauba, Roy, & Dogra, 2017; Zhang, Zhang, & Zheng, 2020) and Deep Learning techniques (Cui, Liu, & Zhang, 2017; Huang, Zhou, Zhang, Li, & Li, 2018). This paper explores the automatic translation of sign language to text by implementing a multi-stream architecture. We replace the RGB-D sensors and wearables by a combination of Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs) to generate depth and joint information from RGB channels. In the second stage, manual and non-manual features are processed in a 3D Convolutional Neural Network, which can capture the temporal components of sign recordings. Thus, our main contributions are:

- Developing a deep learning model for sign language recognition. We validated our method in four datasets: two Brazilian Sign Language (BSL), Indian Sign Language (ISL), and Korean Sign Language (KSL) datasets. The proposed method is independent of depth sensors, and we synthetically generate depth map information through GANs;
- We propose a multi-stream architecture that receives different input types for the model, including facial expressions, hands, and distances between joints. These units are essential for coding a meaning in the context of sign languages;
- We provide an analysis of the visual interpretability, which may be used to refine the models in future research in this area.

The remainder of this paper is organized as follows: Section 2 discusses related work in sign language recognition. Section 3 presents the implemented method. Sections 4 and 5 show the results obtained and the models' visual explanations and discussions. Finally, Sections 6 and 7 conclude this paper with suggestions of future works.

2. Related work

Gesture recognition problems are usually divided into two categories: approaches based on sensors and approaches based on computer vision (Cheok et al., 2019). The first makes use of devices such as gloves (Masood, Srivastava, Thuwal, & Ahmad, 2018), electromyography (EMG) equipment (Du, Jin, Wei, Hu, & Geng, 2017; Geng et al., 2016), radars, and WiFi (Ma, Zhou, Wang, Zhao, & Jung, 2018) to encode and track hand movement (Cheok et al., 2019). Although these approaches have a good result in terms of precision, the use of gloves or EMG can limit the spontaneous way in which a sign is performed. The second approach, which captures sign information from videos, allows the deaf to interact more naturally with the human–computer interaction system (HCI).

In methods based on computer vision, spatiotemporal characteristics can be obtained from video frames, although some studies have considered only static information (Das, Gawde, Suratwala, & Kalbande, 2018; Khari, Garg, Crespo, & Verdú, 2019). In this context, recognizing a sign involves extracting features from a single frame, while body movement is also informative in dynamic approaches (Escalera, Guyon, & Athitsos, 2017). For example, Hidden Markov Models (HMM) were applied to estimate the probability of an observed sequence (Vogler & Metaxas, 2003; Wang, Leu, & Oz, 2006). However, in video analysis, numerous factors must be considered, such as the variation of appearance, the position of a person, and the variation of illumination (Dalal, Triggs, & Schmid, 2006).

Thus, this information retrieval is a challenging task, and robust methods for feature extraction must be used for the problem to be generalized. First, pre-processing steps help to improve the image captured by the camera and reduce lighting variations (Amrutha & Prabu, 2021). The strategies for dealing with variations also involve combining different types of input data, such as RGB images, skeleton points, thermal data, or optical flow information. Other works also use a hybrid approach that combines handcrafted and automatically extracted features to achieve better performance (Rastgoo, Kiani, & Escalera, 2021b).

The recognition of sign languages is an extension of the gesture recognition problem. In addition to hand information – including movement and shape – facial and non-manual features are also considered. Because the hands compose the central phonological units that encode a sign language, one concern is how to segment the hand region efficiently. For this task, the Kinect and Leap Motion sensors are commonly used, allowing to capture depth information and the position of the joints. Other approaches extract the 3D coordinates of an image sequence for the pose estimation (Yan, Zhou, Pan, Yin, & Fang, 2022). Since the communication cycle is bidirectional, motion capture is also used in avatar modeling for the realization of signs by a virtual agent and interaction with the deaf (Dhanjal & Singh, 2022; Rastgoo, Kiani, Escalera and Sabokrou, 2021).

Almeida, Guimarães, and Ramírez (2014) explored the phonological structure of sign languages to build a sign recognition model in BSL. For this, seven features concerning these structures were obtained from RGB-D images, including the areas of the hands — related to their configuration – and the velocity for each sign — related to the type of movement. Raghuveera, Deepthi, Mangalashri, and Akshaya (2020) also used the Kinect sensor to obtain depth maps, which were used for the hand segmentation task, employing the k-means algorithm. They used handcrafted features, including Local Binary Patterns and Histogram of Oriented Gradient, as input to a Support Vector Machine (SVM) for the classification of 140 gestures in the Indian Sign Language (ISL) – numbers, alphabet, and whole words – and combined the predictions using the boosting method. However, no facial expression information was considered in this study, which may provide essential features for differentiating between signs. On the other hand, Rezende, Castro, and Almeida (2016) explored the importance of facial expressions in this context, but it is shown that most studies in the literature still use only single-handed signs (Wadhawan & Kumar, 2021).

Hand and finger movement characteristics were extracted by Chong and Lee (2018) and Katılmış and Karakuzu (2021), through the Leap Motion sensor. In the work of Chong and Lee (2018) the features included angles and distances between adjacent fingers and the radius of the sphere of palm, used for the classification of 26 letters and 10 digits of ASL by SVM and DNN. Similarly, Katılmış and Karakuzu (2021) extracted articulation characteristics by estimating distances and angles between points, generating 119 features that were selected using signal processing techniques. The distances between fingers are demonstrated to be important resources for the classification task since they are

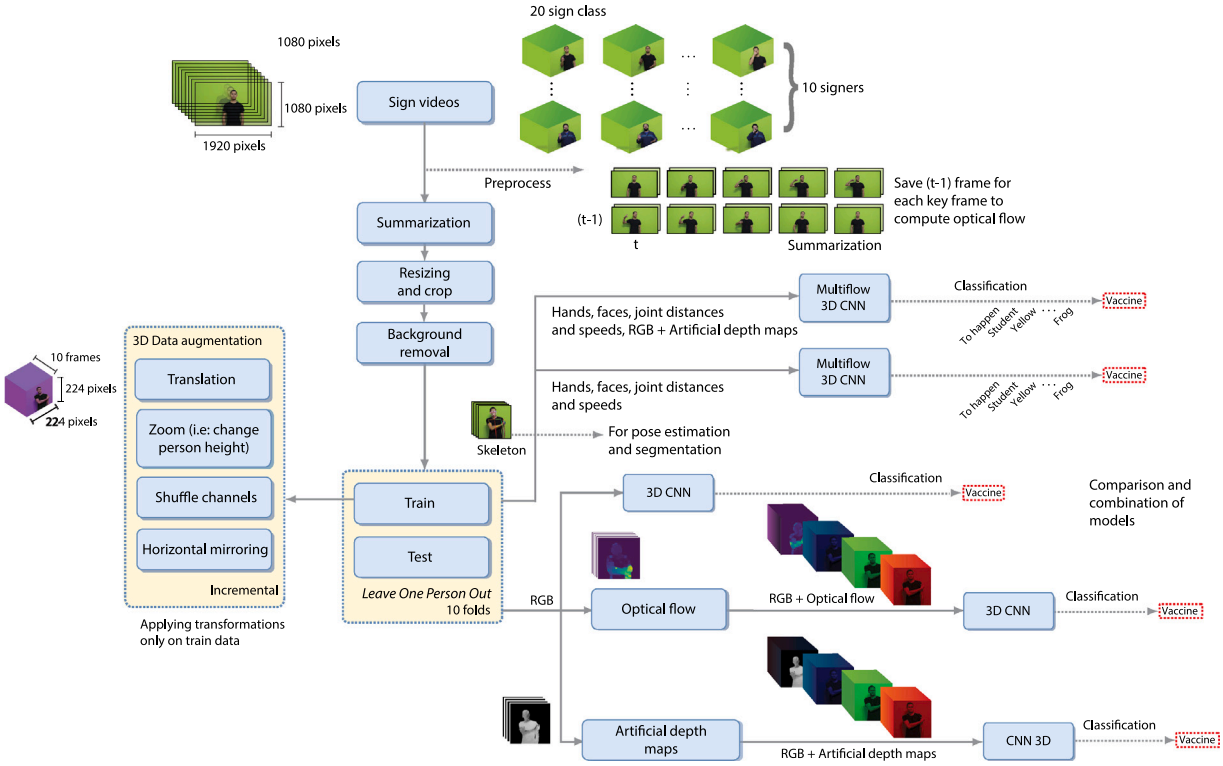


Fig. 1. Proposed method. Our final architecture is formed by a multi-stream network, which receives as input the summarized RGB frames, the segmented regions of the hands and face, the distances between joints, and the artificially generated depth data. The same pre-processing steps applied to training data are applied to test data. Background removal as a pre-processing step allows for dealing with complex backgrounds.

closely related to the shape assumed by the hand in a sign (Chong & Lee, 2018). However, the dependence on the Kinect sensor or Leap Motion makes it more difficult to expand the method.

The CNNs have been widely used to replace manual feature extraction methods. Sharma and Singh (2021) and Zeiler and Fergus (2013) used convolutional neural networks for the sign recognition task, but the temporal information of the frames was not considered in these studies. To consider temporal information, Huang, Zhou, Li, and Li (2015) used a 3D convolutional neural network to recognize hand gestures. They employed a Kinect sensor to track the person's body's position and to estimate the data in depth. Then, a combination of RGB data, the depth map, and skeleton information was used as input to a 3D CNN. Venugopalan and Reghunadhan (2021) propose a bidirectional LSTM architecture combined with a pre-trained GoogleNet network to recognize words commonly used in agriculture in India, such as rice, wheat, and insect. They used different camera orientations and different forms of lighting to capture signs used by deaf farmers in the agricultural context.

Lupinetti, Ranieri, Giannini, and Monti (2020) also extracted temporal information from dynamic gestures, using the Leap Motion sensor to acquire depth data, which is used to track the trajectory of the fingers and hand. The projections on the 2D plane of the positions of the 3D hand joints had temporal information encoded from a color representation. For recognizing hand gestures, these 2D projections were used to feed a convolutional neural network. Passos, Araujo, Gois, and de Lima (2021) uses the Spatio-temporal Gait Energy Image representation to encode the movements of body parts.

In another approach, Liang, Liao, and Hu (2018) proposed a classifier to predict the initial and final frames of each sign, considering that sign languages have three-time phases — preparation, core, and retraction. The first and last phases are irrelevant information to adjust the recognition models, since they are common to different gestures. The classifier inputs consisted of infrared data and contour frames, also acquired through the Kinect sensor. Santhalingam, Pathak, Košecká, Rangwala, et al. (2019) used RGB data from each hand as inputs to a multi-flow 3D convolutional neural network to classify ASL gestures, assuming that hand characteristics can differentiate signs with similar movement patterns. To add the body movement information, skeletal data captured through Kinect were used as input to an LSTM network, combined with the proposed 3D CNN model. Following this line, Rastgoo, Kiani, and Escalera (2021a) uses a multimodal network with two types of input, RGB and depth data, where spatial and temporal features were extracted using convolutional and LSTM networks, respectively. Optical Flow and Scene Flow information were also used as input to the multimodal model. A new approach uses textual descriptions for the sign recognition problem (Bilge, Cinbis, & Ikizler-Cinbis, 2022).

The problem of recognizing sign classes instances not seen during training was explored by Bilge et al. (2022), using zero-shot learning (Lampert, Nickisch, & Harmeling, 2009). This task is done from the semantic transfer of knowledge between classes. As information for zero-shot learning, they used descriptions of signs present in sign language dictionaries, which were combined with visual representations.

Table 1 compares methods, features used, and results of more recent research in this area. As seen in the works described in this section, sign recognition is still a field to be explored. What differentiates our work is recognizing a set of dynamic signs without any instrumented gloves and any depth sensors. This is a notable gap in the related literature, and it is essential for any practical sign language recognition system. As far as we know, no work used computationally generated depth information for sign recognition. The proposed method is general and makes room for research in other sign languages.

The generation of depth images consists of an image translation task, in which the objective is to learn a mapping function of an input image to an output image. Currently, adversarial generative networks (GANs) (Goodfellow et al., 2014) have been used for this type of task, being applied to the synthesis of photographs from semantic manipulation (Wang et al., 2018), to color images (Li et al., 2020), and segmentation (Fuhl, Geisler, Rosenstiel, & Kasneci, 2019).

3. Methods

A sign language usually is defined by five parameters that allow the encoding of a large set of meanings, that are (Brito, 2010):

1. Orientation of the hand: corresponds to the position of the palm concerning the body. For example, this position can be right or left;
2. Configuration: corresponds to the shape assumed by the hands and the finger, which may vary by language;
3. Movements: corresponds to the displacement over time of the hands, fingers, and arms;
4. Articulation point: consists in the location of the sign in the space, having the position of the body as a reference, for example, head, trunk, chest;
5. Non-manual expressions: like facial expressions are used to emphasize a sign, convey negative, affirmative, or interrogative ideas, or express a feeling.

These parameters alone do not allow encoding a sign's meaning, and the solution for a sign recognition system must involve the combination of these units. Therefore, to characterize a sign, we use different types of information as input to a 3D CNN, which allows capturing appearance and motion information from video frames.

For this, we built six sign recognition models: (i) we use the frames in RGB as input to the 3D CNN, as a baseline for the following methods; (ii) we add a fourth channel containing the magnitude information of the optical flow as input to the 3D CNN; (iii) we also test the use of artificially generated depth maps as an additional channel to RGB data; (iv) we combine the probabilities of the outputs generated in *i*, *ii* and *iii*; (v) we use a 3D multi-stream convolutional neural networks, containing the segments of the hands, the face, the distances and the speeds of articulation points; (vi) Finally, we add the RGB frames and the corresponding artificial depth maps in the multi-stream network defined in *v*, retraining the model. Fig. 1 shows the block diagram and each component of the proposed methods.

3.1. Datasets

For experimentation and definition of our method, we used the MINDS-Libras dataset (Almeida, Rezende, Almeida, Toffolo, & Guimarães, 2019; Rezende, Almeida, & Guimarães, 2021), collected by researchers from our laboratory and publicly available. We also use three public datasets to validate our method: the INCLUDE–50 (Sridhar, Ganesan, Kumar, & Khapra, 2020), the Korean Sign Language (Yang, Jung, Kang, & Kim, 2020), and the LIBRAS-UFOP datasets (Cerna, Cardenas, Miranda, Menotti, & Camara-Chavez, 2021), which are presented in the following sections.

3.1.1. MINDS-Libras

The dataset *MINDS-Libras* has 20 classes of signs reproduced five times by 12 signalers, making up about 1200 labeled videos.[2] The acquisition of the signs was carried out by Rezende et al. (2021), through a professional digital camera — Canon EOS Rebel t5i, at a rate of 30 fps and a resolution of 1080×1920. The signs were recorded by a group of volunteers with basic to advanced knowledge in BSL. The choice of gestures was based on the phonological parameters of the language: the orientation of the hand, its movements, its configuration, the point of articulation, and non-manual expressions. Fig. 2 represents a sample of signs grouped according to their similarities, using the T-distributed Stochastic Neighbor Embedding (t-SNE) to visualize the high dimensional data. We group the data using 10 frames per video, and a frame of each sign is displayed for a better view of the proximity between signs.

[2] Precisely 1170 videos since some videos were lost due to failures during the recording session. We use 1000 videos to keep the classes balanced.

Analyzing the lower right corner of Fig. 2, signs with close points of articulation present a high similarity in the two-dimensional space. This is the case of the signs "bank", "bad", "apple", "mirror", and "will". In all of these examples, the sign is performed around the head region. In addition, it is possible to identify a pattern in the upper left corner of the graph. The signs "vaccine", "corner", and "bathroom" are also shown nearby, being executed in regions of the arm. However, it is also noteworthy that the models created for the sign recognition will not necessarily induce these proximity patterns over unseen data since the model can extract specific patterns of the faces, movements, and hands.

3.1.2. INCLUDE dataset

The INCLUDE dataset (Sridhar et al., 2020) contains 263 classes of signs in Indian Sign Language (ISL) subdivided into 15 categories such as clothing, colors, adjectives, and pronouns. Each signaler performs each sign 2 to 6 times, totaling 4287 videos. The INCLUDE–50 for rapid evaluation (Sridhar et al., 2020) has 50 signs categories, with 958 videos. We employ this last dataset to validate our methodology.

3.1.3. Korean sign language dataset

The KSL dataset (Yang et al., 2020) contains 1540 videos in Korean Sign Language. The dataset presents 77 different sign classes, where 20 deaf signalers perform each one.

3.1.4. LIBRAS-UFOP

The LIBRAS-UFOP dataset (Cerna et al., 2021) contains 56 classes of dynamic signs built using the concept of minimal pairs, performed by ten signalers. Changing one of the language parameters in this dataset generates a difference in the sign's meaning with minimal pairs. There are four categories:

1. Signs that present the same movement and the same point of articulation but differ in the configuration of the hand;
2. Signs that have the same hand configuration and the same point of articulation but differ in terms of movement;
3. Signs that present the same hand configuration and similar movements but differ by the point of articulation;
4. Signs that have similar movements and vary in facial expressions. For example, "strong" and "weak", "run fast" and "run slow".

3.2. Preprocessing

First, we submit the videos to a preprocessing step, including the summarization process, background removal, and frame resizing. We perform the summarization step to eliminate redundant frames and transmit the information more quickly, decreasing the computational cost of training the model. However, the summarization process is not the focus of this work, but is used to select a more representative set when hardware requirements are limited, compared to a random sampling of data.

The summarization process consisted of extracting features from the video frames using a pre-trained VGG16 convolutional network (Jadon & Jasim, 2020), which was used to group the frames. We cluster the features extracted from each frame using the K-Means algorithm and PCA and define the frames closest to the centroid as key points. We choose the number of clusters equal to $(10 + 2)$ for selecting a predefined set of 10 frames per video. We exclude the two additional frames, extremes of the selected sequence, because they represent the preparation and retraction phases of the sign. The start of the preparation phase corresponds to the first frame in which the hand leaves the resting state, while the retraction phase corresponds to the return to that state.

Once summarized and resized, we get a representation of the videos, with the dimensions $(10, 224, 224, 3)$, and the next step was to remove

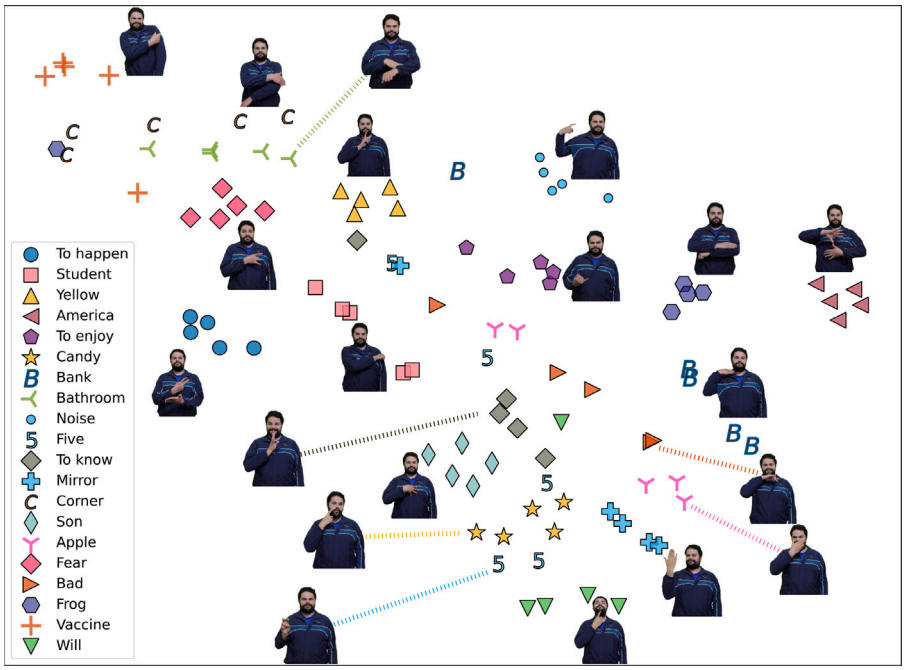


Fig. 2. Sign visualization using t–SNE. There are some patterns concerning the points of articulation for different classes of signs, which constitutes a challenge in sign recognition. Movement, hand configuration, and facial expressions are essential in this task.

the background from the frames. There are different ways to deal with the background variation problem to increase generalization capacity. The first way consists of training the models using diverse backgrounds, aiming that their characteristics are not perceived as important to the model. The second way is to remove the background from the videos before the training, applying the same background removal transformations to new data in the inference process. In this research, we use this second approach. We use the U2-Net architecture (Qin et al., 2020) to remove complex backgrounds. For the MINDS-LIBRAS dataset, which background is composed of a Chroma Key tissue, we define the upper and lower HSV boundaries to detect our color of interest (green color). Then, we specify the lower and upper H as equal to 36 and 70, respectively, and the saturation S at the limits 25 and 255, at the same for V. The next step was the task of data augmentation.

3.2.1. Data augmentation

Data augmentation is generally a beneficial step for deep learning models, reducing the model's bias by introducing further variability into the data. It artificially generates more observations by applying a set of transformations to the existing dataset.

We tested a video-wise online data augmentation technique in this work. In the online stage, i.e., during the training of the 3D CNN, every epoch, we create new video samples by horizontal mirroring, zooming, translating, and shuffling the RGB channels. The proposal to generate new observations with different color channels is to enforce a color invariant model, that is, the task of learning a particular sign is independent of any colors present in the video. Also, it is valuable to notice that horizontal mirroring captures the fact that signs may be executed in either direction, depending on a signaler's dominant hand. Finally, the different versions of the training data in each epoch ensure creating a model with greater generalization capacity for unseen data.

We perform the data augmentation task in a 3D way, that is, at a given epoch, we guarantee that the same transformations were applied for the set of frames within the same video. This ensures, for example, that a person could not change their size from one frame to another in the same video. Then, considering frames of different videos, we perform translation of up to 5% to the right or left; thinning or widening to a limit of 10%; decreasing and increasing the size by up to 5% and 1%, respectively; horizontal mirroring by 5% of the videos and shuffling channels with a 25% probability.

3.3. Models

3.3.1. 3D CNN with RGB frames

In this step, we use the frames in RGB as input to a sequential three-dimensional convolutional neural network that worked as a baseline for our experiments.

For model training, we subdivide the whole data into ten subsets, and we employ the Leave-One-Person-Out method (Seredin, Kopylov, Huang, & Rodionov, 2019; Tahir et al., 2015), allowing a signaler never to be part of a training and test set in the same experiment, in addition to ensuring mutually exclusive folds. If the data from the same signaler is used to train and test the model simultaneously, it can generate information leakage, called overfitting, making the model not general.

The network architecture consists of 3 blocks of 2 convolutional layers each, with 32, 64, and 64 kernels that moving in depth, each followed by a ReLU activation function and a 3D max pooling layer. The dimensions of the filters and max pooling layers are (3,3,3) and (2,3,3). We initialize each kernel with the He normal function (He, Zhang, Ren, & Sun, 2015).

We use 2 fully connected layers and a softmax activation function for the classification step. This activation function transforms the input of the previous layer into a set of 20 probabilities of correspondence to a class. We use mini-batches of size 32 and define the initial learning rate as equal to 0.001, with decay by a factor of 10 when a categorical cross-entropy loss of validation does not increase for 3 consecutive epochs. We use the Adam optimizer and define the number of epochs as equal to 100, with an early stopping if there is no improvements on 15 consecutive epochs.

As a regularization technique, we use the dropout after each max pooling layer, with a 25% probability of remove a node temporarily and randomly; and after the first fully connected layer, with a 50% of probability, every update. We use batch normalization after the first dense layer.

The final result is composed of the average of each iteration of the Leave-One-Out method, where we remove all the data from one signaler for testing. We adopt this same methodology in all experiments, except for the first, to show the importance of this procedure.

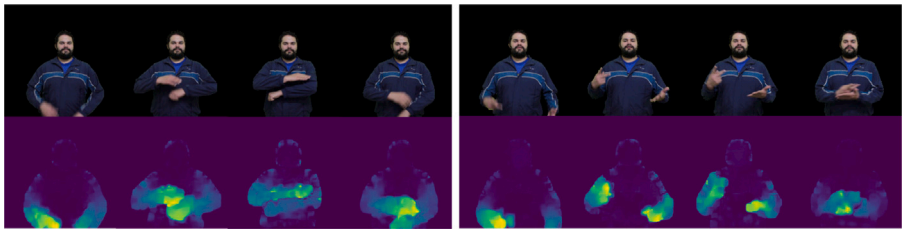


Fig. 3. Optical flow information is used as an additional channel to the RGB data, which feeds a 3D CNN network. We compare different types of input information.

3.3.2. 3D CNN with RGB + Optical Flow frames

In the next experiment, we use the magnitude information of the optical flow as an additional channel to the RGB data, which were used as inputs of the 3D CNN. Using the OpenCV, we estimate the dense optical flow by the Gunnar–Farneback algorithm (Farnebäck, 2003), describing the apparent speed of groups of pixels along the temporal sequences of the frames. We use a window of size 3; 3 pyramid layers, 10 iterations at each pyramid level and 5 pixels of neighborhood for the approximation of the windows by quadratic polynomials.[3] To reduce computational costs, we calculate the optical flow using only the resized and summarized video frames, represented by the time t, and the frames prior to the frames obtained by the summarization process, represented by the instant $(t-1)$. This made it possible to estimate the optical flow from consecutive frames, describing the movement more smoothly, without the need to use all the video frames, which would be computationally expensive. Fig. 3 shows a sequence of summarized frames, followed by the sequences of the magnitude information of the optical flow encoded in an HSV image.

In this experiment, the input of the 3D CNN consists in a volume composed of the 10 selected frames, with channels in RGB and optical flow magnitude, whose size are $n \times h \times w \times c = 10 \times 224 \times 224 \times 4$. Where n corresponds to the number of summarized frames, h and w to the height and width of the images, respectively, and c to the numbers of channels (RGB + magnitude of optical flow). We use the same architecture as defined in Section 3.3.1.

3.3.3. 3D CNN with RGB + Synthetic Depth Maps frames

Given a set of training image pairs containing an RGB frame and the corresponding depth frame, this step aims to create a model that learns a mapping function to generate a depth image from an RGB image. For training the in-depth data generation model through a Conditional Generative Adversarial Network (cGAN), we use the dataset of the Continuous Gesture Challenge (Wan et al., 2016), which includes 249 types of gestures performed by 21 different subjects, in RGB–D. We sample 25,000 images of this dataset for the training set and 5000 for the validation set for the generation of depth maps. We convert all images to gray-scale and apply histogram equalization. Our goal was to generate artificial depth images that represented depth information through a single channel, similar to RGB–D sensors.

The discriminator is used to classify each image patch of the depth maps as real or fake. The more the discriminator improves the performance, the more the generator must improve in the generation of depth maps, like in a race between a detective and a counterfeiter. The discriminator tries to maximize the estimated probability $D(x, y)$ that an instance of real data x is real and minimize the estimated probability that a fake instance is real $D(x, G(x, y))$ (maximize $(1 - D(x, G(x, y)))$). Thus, it should maximize $(\max_D)$ the loss function, ideally classifying actual images as 1 and fake images as 0. The function generator loss, in turn, aims to maximize the prediction error of the discriminator and produce images closer to the actual images (ie, $\min_G$). Eq. (1) expresses the objective function of cGAN ($\mathcal{L}_{cGAN}$) (Isola, Zhu, Zhou, & Efros, 2017).

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, y)))] \tag{1}$$

where x corresponds to the observed image, z represents the random noise vector and y corresponds to the mapping of the random noise vector from the observed image. $\log D(x, y)$ corresponds to the log likelihood of the image being real and $\log(1 - D(x, G(x, y)))$ to the log likelihood of the image from generator being fake. The $\mathcal{L}_{L1}$ loss function defined in Eq. (2) allows the generator to produce depth images closer to the actual images, being used as regularization and weighted by a λ factor. The weight λ was defined as being equal to 100, as adopted in Isola et al. (2017). Thus, the final loss function is given by Eq. (3).

$$\mathcal{L}_{L1}(G) = \mathbb{E}x, y, z[\|y - G(x, z)\|_1] \tag{2}$$

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{3}$$

We use the architecture of conditional GANs proposed by Isola et al. (2017), the Pix2Pix, where the convolution neural networks were employed as discriminator and generator models. We feed the discriminator with the real and generated depth data, with the additional information of the real gray-scale frames as input.

The generator consists of 16-layer U-net, an encoder–decoder architecture containing so-called jump connections between a k layer and an $(n - k)$ layer. The encoder block receives the gray-scale image as input, which is compressed through convolution operations. This stage is formed by 8 convolutional layers with 64, 128 and 256 filters in the first 3 layers, respectively, and 512 filters in each of the following 5 layers. Each kernel has a size $(3, 3)$ and a $(2, 2)$ strider. After the convolution layers, the batch normalization and the LeakyReLU transformation are applied, where $F_{LeakyReLU} = x$, if $x > 0$ and $F_{LeakyReLU} = \alpha x$, if $x < 0$, with $\alpha = 0.2$. The decoder block applies the inverse process for depth maps generation with the same dimensions as the input image. This stage comprises 8 upsampling layers, where the resource map is doubled concerning the previous layer. The final output corresponds to the generated image.

The discriminator, in turn, consists of a module called Patch GAN, since, instead of being analyzed the whole image as being true or false, a patch is analyzed. We use a (32×32) size patch. This stage comprises 4 convolutional layers, followed by batch normalization, except in the first layer, and LeakyReLU activation function. A single output is generated by a sigmoid transformation, with the probability that the image is true or false. We use the Adam optimizer with an initial learning rate of 0.001.

After training, we use the model to predict depth images from the summarized content of the sign videos, as can be seen in Fig. 4. Finally, to compare the addition of the magnitude information and the addition of artificially generated depth maps to the process, we use the same architecture defined in Section 3.3.1.

3.3.4. Combination of probabilities

In this step, we combine the output probabilities P of the models presented in Sections 3.3.1, 3.3.2, and 3.3.3, as shown in Eq. (4).

[3] These arguments are specified in the OpenCV documentation. Available at https://docs.opencv.org/3.4/dc/d6b/group_video_track.html.

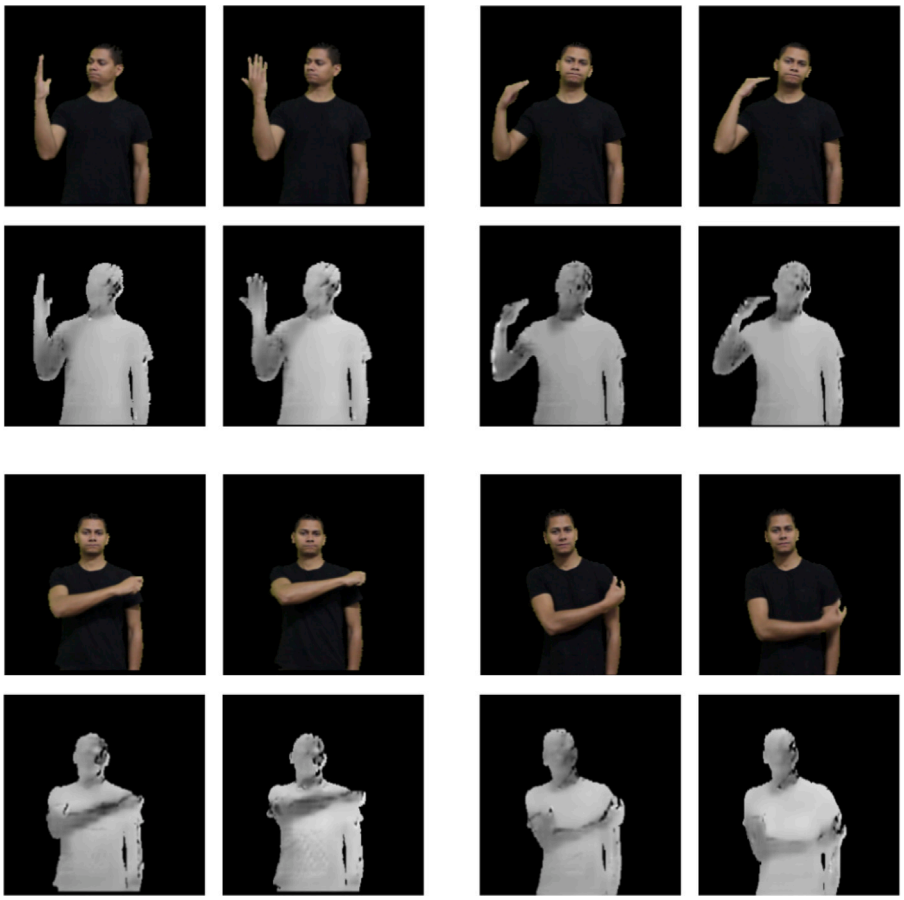


Fig. 4. Depth maps generation. The model is used to infer depth maps from frames of new videos without depth sensors.

$$P_{\text{final}} = \frac{1}{3}(P_{(\text{RGB model})} + P_{(\text{RGB + Optical Flow model})} + P_{(\text{RGB + Synthetic Depth Maps model})}) \quad (4)$$

3.3.5. Multi-stream 3D CNN with hands + head + distance and speed maps

In order to extract more specific features that represent a sign, we also built a multi-flow CNN model, with specific body parts used as input for each 3D CNN stream. In this configuration, one stream receives the hands' segments as input, another the faces segments, and the last the distance and speed maps. The distance maps are formed by the distances between the points of articulation of the hands and the center of the body; and between the upper body joints and this center. We also estimate the distances between the same point of articulation in consecutive frames, representing the speed of that point. In this multi-stream network, we concatenate the features extracted from the CNNs in a vector, and a fully connected layer receives this vector as input.

For hands and face segmentation, we use the *Open Pose* (Cao, Hidalgo Martinez, Simon, Wei, & Sheikh, 2019), an open-source library that performs the pose estimation using the so-called convolutional pose machines (Wei, Ramakrishna, Kanade, & Sheikh, 2016). This task consists of locating regions for estimating the joints of the human body, including joints of the shoulders, knees, hands, and elbow. A set of 137 points are detected for each frame: 25 for the body, 21 for each hand, and 70 for the face. For the segmentation of the hands, we consider the most extreme coordinates of the points of articulation detected to be the minimum and maximum coordinates of the ends of a bound box. Then, we take the $\max(height, width)$ value to create a square box. In the cases where the hand was not detected at the scene, we duplicate the segmentation hand at instant $(n-1)$, performing a similar procedure for segmenting the face.

To create distance maps, first, we assign the center of the signaler body as a reference point, and we calculate the distances from each joint of the right and left hands to this point. We did the same for upper body joints, concatenating this information and forming a (6×21) matrix — one row for the x coordinate and another for the y coordinate of the joints for the right hand, left hand and upper body joints. Next, we calculate the distances between frames for each corresponding point of the body and the hands, forming another (6×21) matrix representing vectors of speeds, i.e., the movement of a joint over time. Finally, we concatenate these distances and speeds maps, forming a new image used as a network input, for each sign video. To keep its dimension comparable with the other inputs, we duplicate the distance map, forming images with 3 channels.

The CNN-hand input consists of the bounding box returned for the RGB hand right segmented with size $10 \times 32 \times 32 \times 3$, the CNN-head with size $10 \times 40 \times 40 \times 3$ and the distance–speed maps with size $10 \times 6 \times 21 \times 3$. Fig. 5 summarizes the process.

For the convolutional flow that receives the hand segments, we use six 3D convolutional layers, and we add a 3D max-pooling layer to every two of those layers. 3D convolutional layers contain 16, 16, 32, 32, 64 and 64 kernels, respectively, with dimensions (3,3,3). The first 3D max-pooling window, in turn, has dimensions (1,2,2) and the following ones have dimensions (2,2,2).

For the convolutional flow that receives the segmented face, we use the same number of 3D convolutional and 3D max-pooling layers and the same number of kernels, with max-polling windows with dimensions (2,2,2). Finally, for the flow that receives distance maps, we use two 3D convolutional layers followed by a 3D max-pooling layer. in such a way that the dimensions of the feature maps are comparable with the feature maps generated in the other convolutional flows. We also use a three-dimensional convolution in this step since we compute the distances for each frame and between frames, and their temporal information is also relevant. The loss function, optimizer,

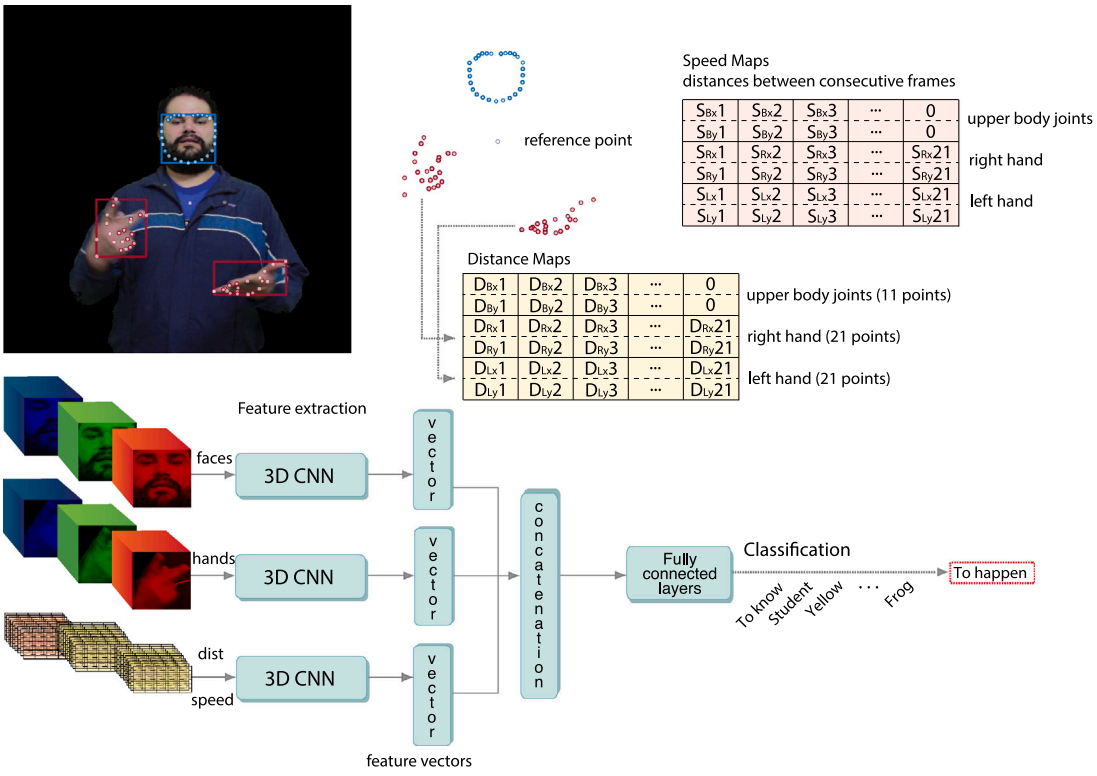


Fig. 5. The proposed method for the multi-stream network. The network receives the segmented hands, the faces, the distances and speeds of the points of articulation, considering the temporal information.

initial learning rate, kernel initializer and number of epochs, were the same as those defined in Section 3.3.1 and we also use dropout as a regularization technique.

Next, we concatenate the feature maps generated in the last 3D max-polling layers of each 3D CNN stream. Then, we use the vector formed as input to a fully connected layer with 32 outputs, followed by a fully connected layer with 20 outputs.

3.3.6. Multi-stream 3D CNN with hands + head + distance and speed maps + RGB and depth maps

Finally, we add the frames in RGB, and the corresponding depth maps in the system, creating a new branch in the multi-stream network described in Section 3.3.5. The 3D convolutional and 3D max pooling layers in this new flow are the same layers defined in Section 3.3.1. Thus, the final architecture is composed of four 3D convolutional flows, which receive as input the segmented hands of the frames, the faces, the distances + speeds of the joints, and the original frames in RGB with the addition of the artificial depth maps.

Table 2 presents the specifications for model training discussed throughout this section.

4. Results and discussions

This section shows the results obtained for the classification of 20 signs in the MINDS-Libras dataset (Section 3.1.1). To evaluate the performance of the models, we use the overall accuracy, the macro average f1-score, and the macro area under the precision–recall curve. Table 3 presents a definition of the experiments described in this work. The performance metrics of the models are shown in Table 4. We also provide the average of false negative rate and the most confusing sign per class for each experiment in Table 5.

We validated our method by comparing it with the results obtained for classifying 50 signs in the INCLUDE–50 dataset (Section 3.1.2), 77 signs in the KSL dataset (Section 3.1.3), and 56 signs in the LIBRAS–UFOP dataset (Section 3.1.4), shown in Table 8.

The results of each experiment consist of the averages of evaluation of the test set obtained in 10 iterations, using the approaches described in Table 3.

The first result of Table 4 shows that by dividing the data set into 10 exclusive folds, without the concern that the data of the same individual cannot be part of both training and test set in the same iteration, the model presents an accuracy of 0.96 ± 0.03, using only the RGB data as input from the network. However, by using the Leave One Person Out approach, removing the entire data for the same individual from the training set, the model's performance reduced to 0.57 ± 0.12. This means that the characteristics of an individual were learned to be important for the first model, presenting a good result for a specific dataset, but not having a good generalization capacity. For this reason, we do not consider the results of this first approach, and we used the leave subjects out method in the other experiments.

Although the accuracy and F1-score of the RGB DM HHDS experiment are 1 pp and 2 pp higher than the HHDS, the analysis of the RGB LOPO (Leave One Person Out) and RGB+DM experiments show a benefit when adding the generated depth maps artificially. Thus, we understand that the advance in research on Generative Adversarial Networks can produce information that impacts the performance of the models of sign recognition, and artificial depth map models can be refined.

As can be seen in Table 5 — in which false negative rates (FNR) are shown for each experiment, the RGB LOPO information has not yet been enough to classify the signs. In this experiment, the most confused signs (highest FNR) are "fear", where 24% was classified as "student"; "will", where 24% was classified as "to know"; and "corner", where 32% was classified as "frog". In both cases, the confused signs are performed at similar joint points. For the first two cases, this can be seen in Fig. 2. This suggests that using only RGB data as the network input, the movement features were more learned than the appearance features, such as the appearance of the hands or face.

To investigate this hypothesis, we also provide a visual explanation for the model, using the approach of Stergiou et al. (2019), which

Table 1
Recent papers that applied ML for sign language recognition, datasets and models used (the best model is reported), features analyzed, interpretability (Inter.), metric results in each paper; Depth (whether depth data is used or not. If yes, which sensor is used. LP: Leap Motion); CV (cross validation); CV Person (cross validation using data from different people in training, testing/validation); NM: Not mentioned.

Reference	Lang.	Methods	Dataset	Data pub.	Features	Depth	Interp.	CV	CV person	Result
Rastgoo et al. (2021a)	[a]Italian (Gestures)	Two stream CNN + LSTM	Four datasets on gesture recognition and action recognition problems	Yes	Optical Flow + RGB/ Scene Flow + Depth videos. Use of pixel-level, hands, flow features, and hand pose features	Kinect	No	Yes	NM	98.97% acc (Italian gestures)/ 99.08% with fine-tuning
Lee et al. (2021)	ASL	LSTM + KNN	26 alphabets for gaming environment (ASL learning)	No	30 features extracted from Leap Motion, such as sphere radius, angles between fingers and distance between finger positions	LM	No	Yes	No	91.82% acc
Venugopalan and Reghunadhan (2021)	Indian (ISL)	BiLSTM + GoogleNet	Signs commonly used in agriculture in India	Yes	Hands in RGB	No	No	Yes	NM	76.21% acc
Sharma and Singh (2021)	Indian (ISL)	VGG CNN	43 classes of static gestures in ISL	No	Hand in RGB	No	No	Yes	NM	99.96 acc%
Katılmış and Karakuzu (2021)	Turkish (TSL)	PCA + LDA + ELM	50 classes of dynamic signs	No	Features such as direction, and angles between articulations	LM	No	Yes	NM	98% acc
Cerna et al. (2021)	Libras (BSL)	2 stream CNN + Rank Pooling. Extract dynamic images	56 classes of signs/four categories	Yes	RGB-D data and skeleton	Kinect	No	Yes	Yes	74.25% acc
Passos et al. (2021)	Libras (BSL)	SVD + SVM	24 dynamic signs		Head, arms, hands, and Gait Energy Image to encode the movement	No	No	Yes	No for CEFET /RJ Libras dataset	85.40% acc
Bilge et al. (2022)	ASL	Zero-Shot Learning	3 datasets with textual descriptions	Yes	Embeddings from SL dictionaries, hands, body	No	Yes	Yes	–	26.9% acc (overall seen/unseen classes)
Vahdani, Jing, Huenerfauth, and Tian (2022)	ASL	Multi-stream CNN	100 ASL sign glosses		RGB-D hands, faces and body poses	Kinect	Yes	No	Different subj. train/test	92.88% acc

[a]Rastgoo et al. (2021a) successfully uses a multi-stream network in a gesture recognition task, which is a subtask of sign recognition problem.

Table 2
Specifications for model training.

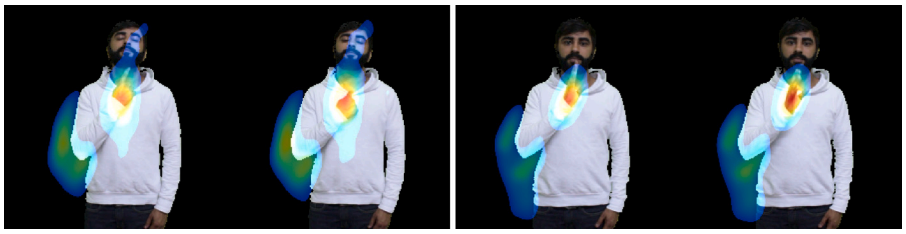
Hyperparameter	Specifications
Batch size	32
Initial learning rate	0.001
Learning rate decay	Factor of 10
Number of epochs	100
Regularization	*Dropout*. 25% after each layer of 3D *max–pooling* and 50% after the first fully connected layer
Early stop	Yes, if there is no improvement in the loss function after 15 consecutive epochs
Optimizer	Adam
Batch normalization	Yes

shows the main activated temporal regions along the frames. We chose this approach to encode the meaning of a sign since both appearance and movement are important sources of information. Fig. 6 shows the temporal activation for the two signs "will" and "to know".

In both, similar regions were activated over time. However, in the "will" sign, facial features were also important, but the activated regions were not enough to differentiate between these signs. This can be explained because some signalers also modify their facial expressions for the sign "to know", showing an expression of happiness, for example. The network would need to focus more on appearance features in this respect.

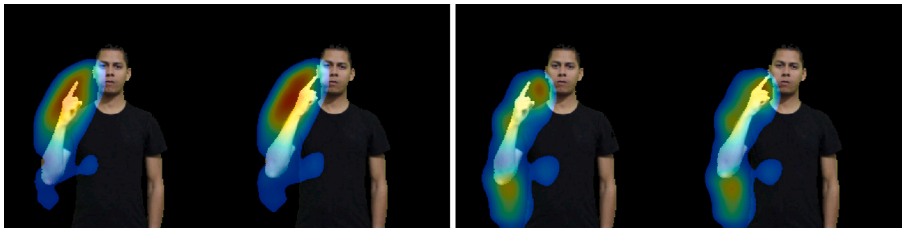
By adding the magnitude information of the optical flow, the model's performance improved from 0.57 ± 0.12 to 0.60 ± 0.09. Of the 20 signs considered, 13 had a performance improvement when compared to the baseline, as highlighted in Table 5. The signs that showed the best improvement in the false negative rate, in percentage, were the signs "candy", "noise" and "corner". Thus, based on these results, we investigate the issue: why did the classification of these signs improve after adding the magnitude information of the optical flow?

Looking at the visual explanations, this addition may have caused the network to focus even more on the signs' movement information. For the "candy" sign (Fig. 7), for example, the movement of the elbows



(a) "Will" sign for a test data (b) "To know" sign for a test data

Fig. 6. Visual explanations for the 3D CNN model that receives RGB images as input. The "will" and "to know" signs have similar regions activated over time.



(a) Sign "candy" for the RGB model (b) Sign "candy" for the RGB + OF model

Fig. 7. Visual explanations for the 3D CNN model that receives RGB images as input (Fig. 7(a)) and RGB + Optical Flow images (Fig. 7(b)). The addition of the optical flow activated the elbow region more strongly, but other information was lost.

Table 3
Description of the experiments.

Experiment	Description
RGB	3D CNN model that receives RGB frames as input. Model created by dividing the dataset into 10 unique folds. Nine of these are used for training and 1 for testing at each iteration. However, the signalers can participate in the training and test set in the same iteration, with different videos. This approach generates overfitting and is used to justify the subsequent experiments.
RGB LOPO	3D CNN model that receives RGB frames as input. A baseline for model comparison using Leave-One-Person-Out (LOPO) approach. At each iteration, the entire dataset for a signaler is left out for testing. All the following experiments are carried out following this methodology of LOPO.
RGB+OF	Sequential 3D CNN model that receives RGB frames as input and the magnitude information of the optical flow as an additional channel.
RGB+DM	Sequential 3D CNN model that receives RGB frames as input and artificially generated depth maps as an additional channel.
ENS	Combination of the probabilities of the outputs generated by the RGB LOPO, RGB + OF and RGB + DM models.
HHDS	Multi-stream 3D CNN network, which receives as input the segmented hands, the segmented heads, the distances from the points of articulation to the center of the body and the speeds of each point.
RGB+DM HHDS	Multi-stream 3D CNN network, which receives the same inputs of the HHDS model and the RGB frames + Depth Maps.

Table 4
Classification report.

Experiment\Eval.	Accuracy	F1-score	Prec.–Recall AUC
RGB	0.96 ± 0.03	0.96 ± 0.03	–
RGB LOPO	0.57 ± 0.12	0.53 ± 0.13	0.69 ± 0.12
RGB+OF	0.60 ± 0.09	0.56 ± 0.10	0.71 ± 0.08
RGB+DM	0.64 ± 0.06	0.61 ± 0.07	0.76 ± 0.07
ENS	0.65 ± 0.08	0.62 ± 0.08	0.80 ± 0.09
HHDS	0.90 ± 0.07	0.88 ± 0.08	0.95 ± 0.05
RGB DM HHDS	0.91 ± 0.07	0.90 ± 0.08	0.96 ± 0.04

showed a greater activation after the addition of magnitude of the optical flow in the network, but the regions around the index finger were less activated.

Thus, although the model's performance has improved, it could incorrectly classify signs that have similar displacements over time. Thus, as the amount of data increased, that would not be good.

Adding the artificially generated depth maps, it is observed that 18 of the signs had their performance improved, considering the FNR. The most significant improvements observed concerning false negative rates are in the signs "yellow", "to know", "corner", and "fear". We observe that the regions of the hands are more activated and also of the faces, when the latter is essential to characterize the sign (Fig. 8). The result did not improve much by making the ensemble of these three models.

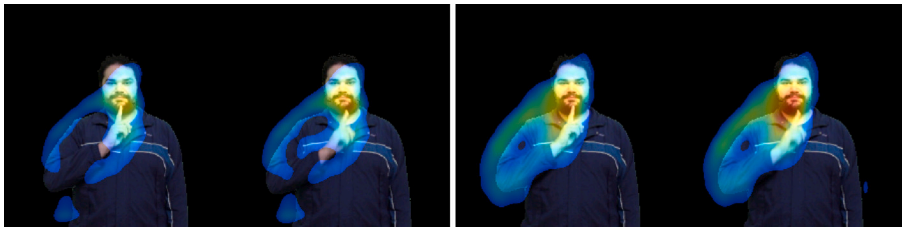
In addition, Fig. 9 shows the temporal activation for the signs "bad" and "corner", where facial information is important for characterizing the first sign, but not for the second. The activation regions are consistent for both cases.

When we add the hand and face information and the distances and speed maps in the multi-stream network, the performance of the model improved for the signs, as shown in the confusion matrix of Fig. 10. This suggests that the main characteristics learned by the three-dimensional convolutional network consisted of representations related to the movement, while the multi-flow network allowed to recover appearance information more efficiently, allowing a better distinction between signs.

However, the sign "fear" still presented the worst performance in terms of true positive rate. As shown in the confusion matrix, 10% of the "fear" signs were classified as "noise", 8% as "son" and 8% as "student". As seen in Fig. 2, the signs "fear" and "student" have a high proximity, with nearby points of articulation. Besides that, although the "fear" and "son" signs have different sequences of movements, in some time steps their hand configurations are similar. The hand configurations on that specific sign also presented a great divergence between signalers, which constitutes a challenging problem of gesture

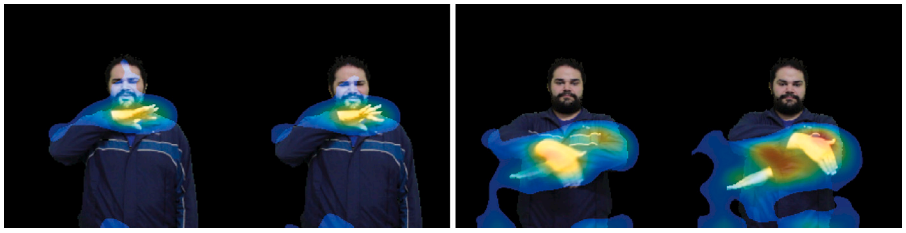
Table 5
False Negative Rate (FNR) and Most Confused Sign (MCS), for each class. We compare The RGB OF, RGB DM and ENS experiments with the RGB LOPO baseline. In these experiments, when a sign shows an improvement in the FPR rate over RGB LOPO, the values are highlighted. Considering the entire table, the signs confused at a rate greater than or equal to 15% are also highlighted.

Sign	RGB LOPO		RGB OF		RGB DM		ENS		HHDS		RGB DMHHDS	
	FNR	MCS	FNR	MCS	FNR	MCS	FNR	MCS	FNR	MCS	FNR	MCS
To happen	0.46	Bathroom	0.42	Corner	0.36	Son	0.40	Corner	0.08	Bathroom	**0.04**	Five, Son
Student	0.48	Bad	0.50	Bad	0.42	Bad, To happen, Vaccine	0.46	Bad	0.20	Vaccine	**0.18**	Vaccine
Yellow	0.18	Will	0.26	To know	0.12	Will	**0.06**	Will	0.10	To know	**0.06**	To know
America	0.30	Bathroom, Son	0.28	Frog	0.22	Frog	0.20	Bathroom, Frog	**0**	–	**0**	–
To enjoy	0.34	Bank	0.32	Bank	0.28	Yellow	0.28	Bank	0.20	Five	**0.06**	Fear, Five
Candy	0.26	Bank	0.18	Bank	0.28	Bank	0.22	Bank	0.04	Bank, Fear	**0.02**	Five
Bank	0.38	Candy	0.36	Candy, To enjoy	0.22	Candy	0.26	Candy	**0.12**	Noise	0.16	Noise
Bathroom	0.44	Frog	0.46	America	0.50	To happen	0.40	To happen	**0.10**	Frog	0.14	Frog
Noise	0.42	Five	0.29	Bank	0.37	Bank	0.31	Five	**0.02**	Five	0.04	Bank
Five	0.36	Mirror	0.52	Mirror	0.48	Mirror	0.44	Mirror	**0.04**	Noise, To know	0.08	To enjoy
To know	0.42	Will	0.34	Will	0.30	Will	0.28	Will	**0.02**	To enjoy	0.06	Yellow
Mirror	0.26	Five	0.42	Five	0.24	Five	0.20	Bank	0.10	To enjoy	**0.06**	Five
Corner	0.68	Frog	0.46	Student	0.40	Frog	0.44	Frog	**0.02**	Bathroom	**0.02**	Frog
Son	0.50	Bad	0.40	Bad	0.44	Bad	0.40	Bad	**0.17**	Fear	**0.17**	Bank
Apple	0.52	Will, Yellow	0.56	To know	0.50	Will, Yellow	0.52	Bad, Will, Yellow	0.12	Yellow	**0.04**	Bad, Yellow
Fear	0.74	Student	0.66	Student, To know	0.50	Vaccine	0.60	Student	0.44	Noise	**0.34**	Son
Bad	0.32	Son	0.36	Apple	0.30	Student	0.26	Son	0.04	Corner, Vaccine	**0.02**	Corner
Frog	0.50	Corner	0.48	America	0.36	Corner	0.46	Corner, To happen	0.04	Bathroom, Corner	**0.02**	Corner
Vaccine	0.40	Student	0.34	Student	0.30	Student	0.34	Fear, Student	**0.12**	Student	0.20	Student
Will	0.56	To know	0.46	To know	0.54	To know	0.48	To know	**0.12**	To know	0.14	To know



(a) Importance of facial expression. (b)

Fig. 8. Visual explanations for the 3D CNN model that receives RGB images as input (Fig. 8(a)) and RGB + Depth Maps images (Fig. 8(b)).



(a) Sign "bad". Facial expressions are important to characterize the sign (b) Sign "corner". In this case, facial expressions are not important

Fig. 9. Visual Explanations for the signs "bad" (Fig. 9(a)) and "corner" (Fig. 9(b)). The activation regions are consistent.

recognition. Regarding the noise sign, both are expressed with a facial expression that approaches disgust.

When adding artificial depth maps as input of the multi-stream network, the true positive rate for the "fear" sign showed an improvement of 10 pp, compared to the multi-stream model without the addition of depth maps. Finally, the difference between the use of the sequential network that receives RGB frames as input and the multi-stream network can be seen in Fig. 11, that shows the macro averages of the precision and recall curves for the RGB LOPO and RGB+DM HHDS experiment. This curve provides a measure of the ability to discriminate between signs and the closer its value to 1, the better the classifier. The average area under this curve, for each experiment, is shown in Table 4. For the experiment using the multi-stream network with depth maps, the macro average for the 10 folds showed a performance of 96%.

The time consumed in each step of the process with a Tesla V100-SXM2-16 GB GPU is shown in Table 6.

Finally, we apply our method to public benchmarks. Table 7 shows the description of the data division in each experiment, with the results shown in Table 8.

Compared with the method that does not depend on depth sensors proposed by Passos et al. (2021) and the experiments that do not use this sensor in Cerna et al. (2021), our method achieved better results in BSL recognition considering two different datasets.

We also show the results for each category of the LIBRAS-UFOP dataset in Table 9. Notably, in Category 4 (Section 3.1.4) of the LIBRAS-UFOP dataset, which presents signs with similar movements and variations in facial expressions (such as "run fast" and "run slow"), we reached an accuracy of 83%. This performance is superior to other methods, even those using depth sensors.

Furthermore, considering the ISL dataset, we achieved an accuracy of 94%. Fig. 12 shows the precision and recall curve for the LIBRAS-UFOP and INCLUDE-50 datasets, showing that sign languages formed by parameters similar to those described in this work can also benefit from our proposed method.

We noticed that it was possible to classify dynamic and static signs using a single network using our method, considering both manual and non-manual representations. This allows the method to be expanded

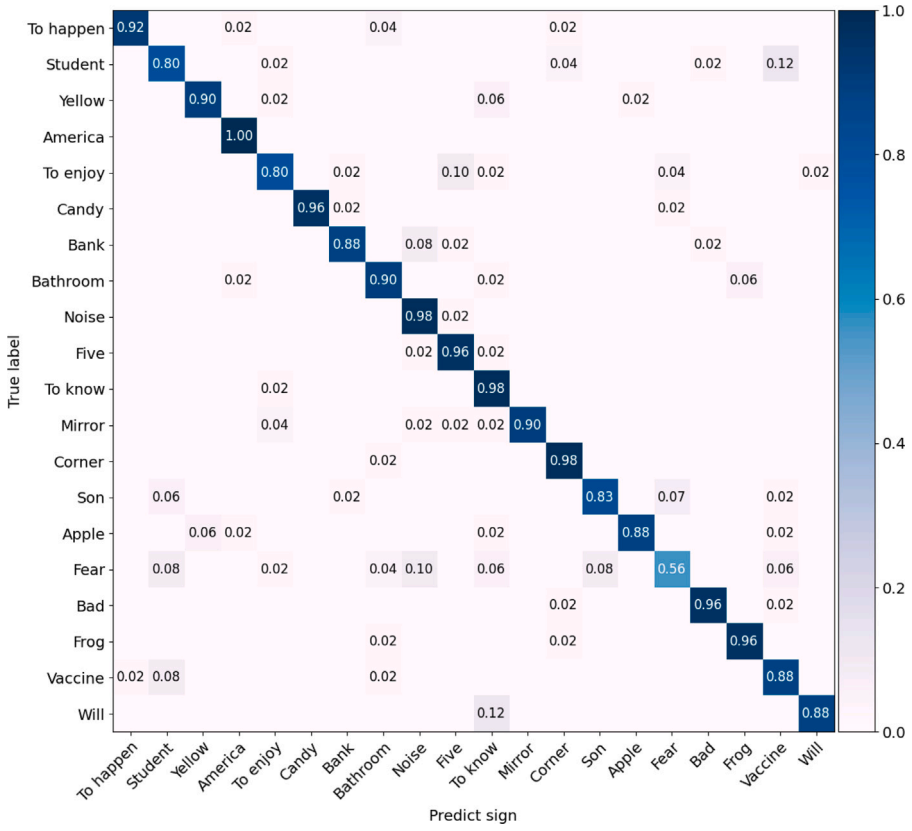
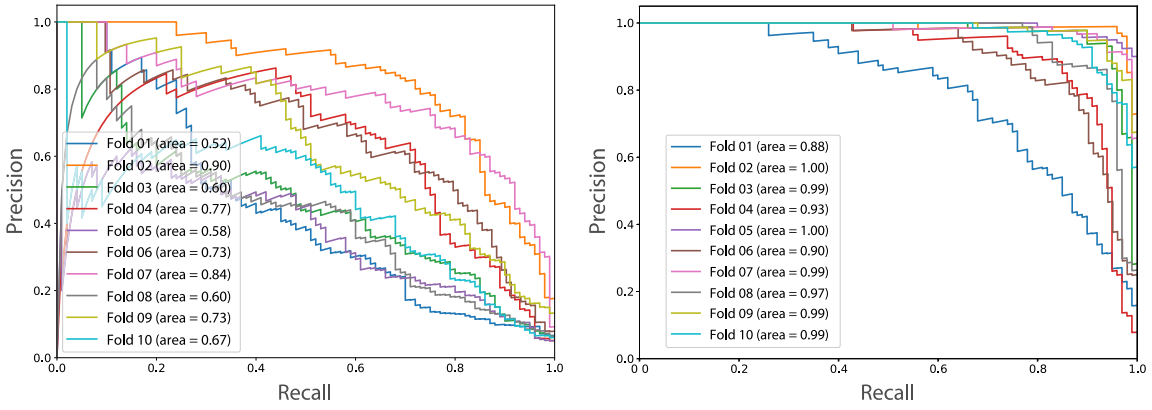


Fig. 10. Confusion matrix for the multi-stream network.



(a) Using only RGB information as input of a sequential model

(b) Using the multi–stream model with artificial depth maps

Fig. 11. AUC Precision–Recall. Only the RGB information, without segmentation, is not enough to encode the signs (Fig. 11(a)). On the other hand, by adding information that represents the phonological parameters of visual languages, the performance of the model is improved (Fig. 11(b)).

Table 6
Time spent in each step of the process. Each video is variable in length and is summarized in 10 frames to eliminate redundancy. For the time calculation, we used 100 signs videos.

Background removal	Distances/Speeds	Depth maps generation	Classification
HSV: (8.23 ± 1.26) ms U–Net: (522.15 ± 27.16) ms	(0.15 ± 0.03) ms	(7.06 ± 0.48) ms	(15.20 ± 0.38) ms

Table 7
Description of the data division in each experiment.

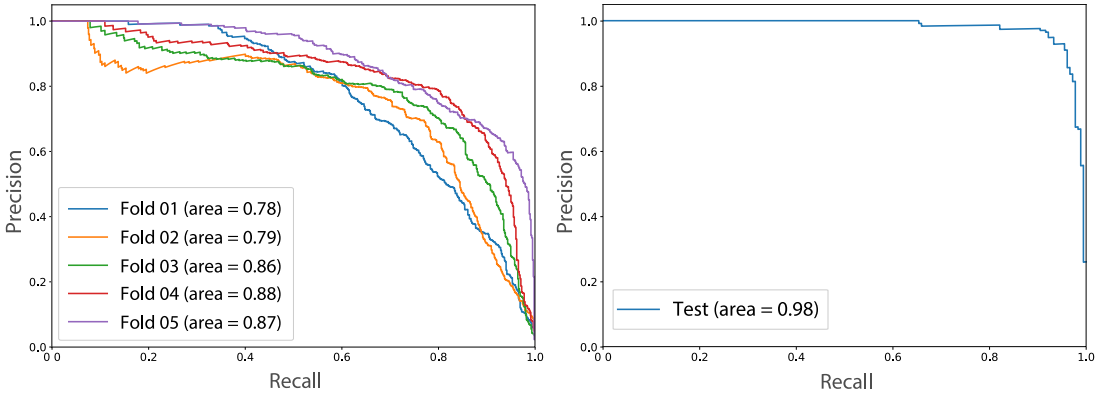
Dataset	Description
MINDS–Libras (Rezende et al., 2021)	We split the data into 10 folds using the Leave One Person Out approach. The data of one signaler is part of the test set at each iteration, and 9 are part of the training set. The final result is the average test performance of the experiments.
LIBRAS–UFOP (Cerna et al., 2021)	We split the data into 5 folds using the Leave One Person Out approach. We use the same division of the folds proposed by Cerna et al. (2021). We evaluate the result by category in the test set and compute the average performance of the experiments by category. The final result is the average of all the signalers considering the five iterations.
INCLUDE–50 (Sridhar et al., 2020)	Sridhar et al. (2020) previously defined the videos that make up the training and test sets. We use the same division proposed in INCLUDE-50 (Sridhar et al., 2020).
KSL Dataset (Yang et al., 2020)	We split the data into 5 folds. The data of four signalers are part of the test set at each iteration, and 16 are part of the training set. The final result is the average test performance of the experiments. Yang et al. (2020) does not explicitly inform how they split the data.

Table 8
Evaluation of our proposed method in public benchmarks. In Cerna et al. (2021), we compare methods that do not depend on depth sensors (Cerna et al., 2021 RGB) and methods that use depth sensors (Cerna et al., 2021 Kinect).

Method	Dataset	Accuracy	F1-score
RGB DM HHDS (Ours)	MINDS-Libras	0.91 ± 0.07	0.90 ± 0.08
Passos et al. (2021)	MINDS-Libras	0.85 ± 0.02	–
RGB DM HHDS (Ours)	LIBRAS-UFOP	0.74 ± 0.04	0.71 ± 0.05
Cerna et al. (2021) RGB	LIBRAS-UFOP	0.61 ± 0.03	–
Passos et al. (2021)	LIBRAS-UFOP	0.65 ± 0.04	–
Cerna et al. (2021) Kinect	LIBRAS-UFOP	0.74 ± 0.03	–
RGB DM HHDS (Ours)	INCLUDE–50	0.94	0.93
Sridhar et al. (2020)	INCLUDE–50	0.95	–
RGB DM HHDS (Ours)	KSL Dataset	0.50 ± 0.05	0.48 ± 0.04
LRCN + Transfer learning (Yang et al., 2020)	KSL Dataset	0.80	–

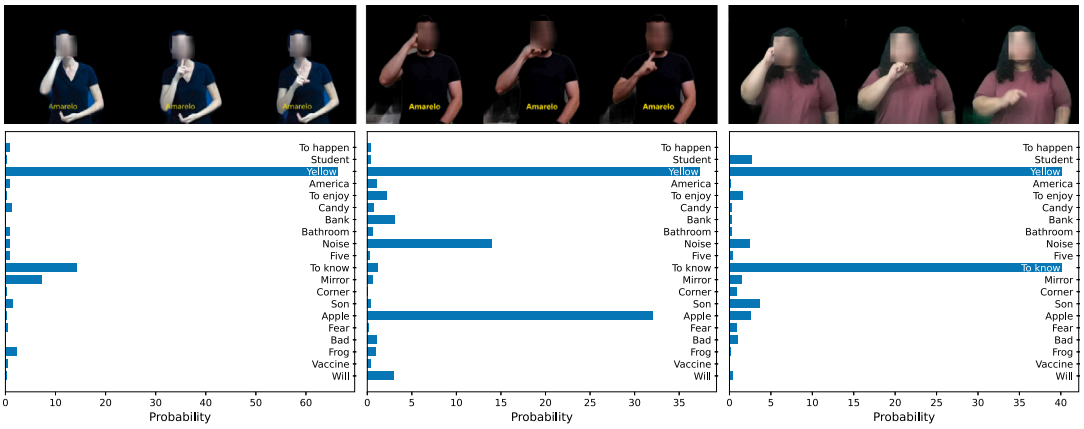
Table 9
Results for each category in the LIBRAS-UFOP dataset.

Method	Categ. 1	Categ. 2	Categ. 3	Categ. 4	All
Cerna et al. RGB	0.62 ± 0.03	0.60 ± 0.02	0.76 ± 0.01	0.60 ± 0.02	0.61 ± 0.03
Passos et al.	0.58 ± 0.03	0.72 ± 0.01	0.93 ± 0.06	0.82 ± 0.07	0.65 ± 0.04
Ours	0.60 ± 0.03	0.72 ± 0.09	0.76 ± 0.07	0.83 ± 0.07	0.74 ± 0.04
Cerna et al. Kinect	0.79 ± 0.04	0.72 ± 0.03	0.92 ± 0.02	0.72 ± 0.02	0.74 ± 0.03

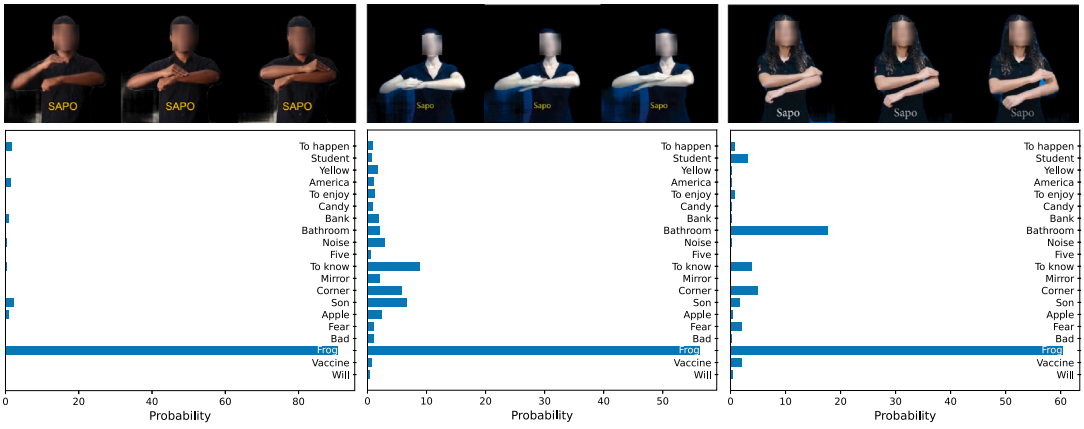


(a) Precision-Recall LIBRAS-UFOP by signaller (b) Precision-Recall INCLUDE–50

Fig. 12. AUC Precision–Recall.



(a) Output, in probability, of the “yellow” sign



(b) Output, in probability, of the “frog” sign

Fig. 13. Sample of videos collected from YouTube and the corresponding predictions for each video, where the highest probability gives the final prediction. In Fig. 13(a), all signs were correctly classified as yellow, and in Fig. 13(b), all signs were correctly classified as frog. We blurred the faces for the subjects’ privacy.

since when using real-world data, there would be no separation stage between static and moving signs, and even static signs are described within a moving sequence.

5. Prediction on YouTube videos

In this section, we present the results, in probability, of the predictions of “frog” and “yellow” signs collected from YouTube videos.

All signs were correctly classified when making predictions from these videos, as shown by the highest probability corresponding to the actual sign. However, the top-2 probability varies between videos. For example, in the second video of Fig. 13(a), the subject moves his lips to pronounce the yellow word. This lip movement explains the second-highest probability of this sequence, the apple sign, which is performed using facial expressions. The variation between the videos shows the challenge of recognizing signs, which may present regional differences and variations in the execution side of the sign, for example. There is no better or worse form of sign.

6. Applications and future works

Several studies explore the use of depth sensors for the sign recognition task, suggesting that this type of data can improve the generalization of models. However, as the model expands to encompass a more extensive vocabulary, depth sensors become impractical, as much data is present only in RGB. Using depth data in model training implies that sign recognition will depend on the use of these sensors. In this context, a solution that uses only computational factors to estimate joint position and depth is more suitable. With this idea, we propose a depth sensor independent method for the sign recognition task. Considering that the largest source of sign data comes from the internet, our method has the potential to be expanded on a large scale. However, there are new stages towards the translation task. Identifying signs within a sentence is even more challenging since pausing does not define the transition between them. Thus, in future work, we intend to identify candidate temporal regions that contain a sign sequence on a time scale, extending the 2D applications for segmenting gestures in videos. These candidate volumes could be classified using the methods presented in this work. Once detected a set of signs temporally and spatially in a sequence of frames, the next step consists of describing sentences from one domain to another. Besides that, refining the depth map generation model is also presented as future work. One important implication of this research is applying this method in new ways of human–computer interaction using visual languages, for instance, virtual personal assistants.

7. Conclusion

In this work, we propose a method that receives only the RGB data as input for the sign recognition task, indirectly getting the pose and depth information. Most of the works described make use of depth sensors for hand tracking, as a way to facilitate the segmentation task or to provide additional information to the classifiers. Our proposal for an independent method of depth sensors and instrumented gloves

opens the possibility of expanding the task, using a set of vocabulary that allows encoding a language. The multi-stream architecture showed a greater capacity for generalization for different signalers, and the addition of artificial depth maps seems to be promising.

CRediT authorship contribution statement

Giulia Zanon de Castro: Conceptualization, Methodology, Software, Writing – original draft. **Rúbia Reis Guerra:** Methodology, Writing – review & editing, Validation. **Frederico Gadelha Guimarães:** Conceptualization, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We used public datasets in our research.

Acknowledgments

This work has been supported by the Coordination for the Improvement of Higher Education Personnel (CAPES, *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*), the Foundation for Research of the State of Minas Gerais (FAPEMIG, *Fundação de Amparo à Pesquisa do Estado de Minas Gerais*) and by the National Council for Scientific and Technological Development (CNPq), Brazil, Grants no. 306850/2016-8 and 312991/2020-7. We thank Dr. Sílvia Grasiella from the Federal Institute of Minas Gerais (IFMG), https://www.ifmg.edu.br/ouropreto, in the city of Ouro Preto, Brazil, and the MINDS (*Machine Intelligence and Data Science*) Laboratory at UFMG, https://minds.eng.ufmg.br/, for the database acquisition and curating.

The authors would like to thank the anonymous reviewers for their earnest efforts and thorough review of our manuscript.

References

Almeida, S. G. M., Guimarães, F. G., & Ramírez, J. A. (2014). Feature extraction in Brazilian Sign Language Recognition based on phonological structure and using RGB-D sensors. *Expert Systems with Applications*, *41*(16), 7259–7271.

Almeida, S. G. M., Rezende, T. M., Almeida, G. T. B., Toffolo, A. C. R., & Guimarães, F. G. (2019). MINDS-libras dataset. URL: https://doi.org/10.5281/zenodo.2667329.

Amrutha, K., & Prabu, P. (2021). ML based sign language recognition system. In *2021 international conference on innovative trends in information technology* ICITIIT, (pp. 1–6). IEEE.

Bai, Y., & Bruno, D. (2020). Addressing communication barriers among deaf populations who use American sign language in hearing-centric social work settings. *Columbia Social Work Review*, *18*(1), http://dx.doi.org/10.7916/CSWR.V18I1.5928, URL: https://journals.library.columbia.edu/index.php/cswr/article/view/5928.

Barnett, S., Klein, J. D., Pollard, R. Q., Samar, V., Schlehofer, D., Starr, M., et al. (2011). Community participatory research with deaf sign language users to identify health inequities. *American Journal of Public Health*, *101*(12), 2235–2238. http://dx.doi.org/10.2105/ajph.2011.300247.

Bilge, Y. C., Cinbis, R. G., & Ikizler-Cinbis, N. (2022). Towards zero-shot sign language recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., et al. (2019). Sign language recognition, generation, and translation: An interdisciplinary perspective. In *The 21st international ACM SIGACCESS conference on computers and accessibility* (pp. 16–31). ACM, http://dx.doi.org/10.1145/3308561.3353774, URL: https://dl.acm.org/doi/10.1145/3308561.3353774.

Brito, L. F. (2010). *Por uma gramática de línguas de sinais*. TB-Edições Tempo Brasileiro.

Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., & Sheikh, Y. A. (2019). OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Cerna, L. R., Cardenas, E. E., Miranda, D. G., Menotti, D., & Camara-Chavez, G. (2021). A multimodal LIBRAS-UFOP Brazilian sign language dataset of minimal pairs using a microsoft Kinect sensor. *Expert Systems with Applications*, *167*, Article 114179.

Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, *10*(1), 131–153. http://dx.doi.org/10.1007/s13042-017-0705-5.

Chong, T.-W., & Lee, B.-G. (2018). American sign language recognition using leap motion controller with machine learning approach. *Sensors*, *18*(10), 3554.

Cui, R., Liu, H., & Zhang, C. (2017). Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7361–7369).

Dalal, N., Triggs, B., & Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *European conference on computer vision* (pp. 428–441). Springer.

Das, A., Gawde, S., Suratwala, K., & Kalbande, D. (2018). Sign language recognition using deep learning on custom processed static gesture images. In *2018 international conference on smart city and emerging technology* ICSCET, (pp. 1–6). IEEE.

Dhanjal, A. S., & Singh, W. (2022). An automatic machine translation system for multilingual speech to Indian sign language. *Multimedia Tools and Applications*, *81*(3), 4283–4321.

Du, Y., Jin, W., Wei, W., Hu, Y., & Geng, W. (2017). Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation. *Sensors*, *17*(3), 458.

Escalera, S., Guyon, I., & Athitsos, V. (2017). *Gesture recognition*. Springer.

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on image analysis* (pp. 363–370). Springer.

Forshay, L., Winter, K., & Bender, E. M. (2016). SignAloud open letter.

Fuhl, W., Geisler, D., Rosenstiel, W., & Kasneci, E. (2019). The applicability of Cycle GANs for pupil and eyelid segmentation, data generation and image refinement. In *Proceedings of the IEEE international conference on computer vision workshops*.

Geng, W., Du, Y., Jin, W., Wei, W., Hu, Y., & Li, J. (2016). Gesture recognition by instantaneous surface EMG images. *Scientific Reports*, *6*, 36571.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *27*.

Guzsvinecz, T., Szucs, V., & Sik-Lanyi, C. (2019). Suitability of the kinect sensor and leap motion controller—A literature review. *Sensors*, *19*(55), 1072. http://dx.doi.org/10.3390/s19051072.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034).

Hoy, M. B. (2018). Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical Reference Services Quarterly*, *37*(1), 81–88. http://dx.doi.org/10.1080/02763869.2018.1404391.

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

Huang, J., Zhou, W., Zhang, Q., Li, H., & Li, W. (2018). Video-based sign language recognition without temporal segmentation. In *Thirty-second AAAI conference on artificial intelligence*.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125–1134).

Jadon, S., & Jasim, M. (2020). *Video summarization using keyframe extraction and video skimming: Technical Report*, EasyChair.

Katılmış, Z., & Karakuzu, C. (2021). ELM based two-handed dynamic turkish sign language (TSL) word recognition. *Expert Systems with Applications*, Article 115213.

Khari, M., Garg, A. K., Crespo, R. G., & Verdú, E. (2019). Gesture recognition of RGB and RGB-D static images using convolutional neural networks. *International Journal of Interactive Multimedia & Artificial Intelligence*, *5*(7).

Kumar, P., Gauba, H., Roy, P. P., & Dogra, D. P. (2017). Coupled HMM-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters*, *86*, 1–8.

Kushalnagar, P., Moreland, C. J., Simons, A., & Holcomb, T. (2018). Communication barrier in family linked to increased risks for food insecurity among deaf people who use American Sign Language. *Public Health Nutrition*, *21*(5), 912–916. http://dx.doi.org/10.1017/S1368980017002865.

Lampert, C. H., Nickisch, H., & Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 951–958). IEEE.

Lee, C. K., Ng, K. K., Chen, C.-H., Lau, H. C., Chung, S., & Tsoi, T. (2021). American sign language recognition and training method with recurrent neural network. *Expert Systems with Applications*, *167*, Article 114403.

Li, H., Zhang, L., Zhang, X., Zhang, M., Zhu, G., Shen, P., et al. (2020). Color vision deficiency datasets & recoloring evaluation using GANs. *Multimedia Tools and Applications*, *79*(37), 27583–27614.

Liang, Z.-j., Liao, S.-b., & Hu, B.-z. (2018). 3D convolutional neural networks for dynamic sign language recognition. *The Computer Journal*, *61*(11), 1724–1736.

Lupinetti, K., Ranieri, A., Giannini, F., & Monti, M. (2020). 3D dynamic hand gestures recognition using the leap motion sensor and convolutional neural networks. In *International conference on augmented reality, virtual reality and computer graphics* (pp. 420–439). Springer.

Ma, Y., Zhou, G., Wang, S., Zhao, H., & Jung, W. (2018). Signfi: Sign language recognition using wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *2*(1), 1–21.

Marin, G., Dominio, F., & Zanuttigh, P. (2014). Hand gesture recognition with leap motion and kinect devices. In *2014 IEEE international conference on image processing* ICIP, (pp. 1565–1569). http://dx.doi.org/10.1109/ICIP.2014.7025313.

Masood, S., Srivastava, A., Thuwal, H. C., & Ahmad, M. (2018). Real-time sign language gesture (word) recognition from video sequences using CNN and RNN. In *Intelligent engineering informatics* (pp. 623–632). Springer.

Meulder, M. (2019). *The legal recognition of sign languages : Advocacy and outcomes around the world*. Bristol Blue Ridge Summit: Multilingual Matters.

Passos, W. L., Araujo, G. M., Gois, J. N., & de Lima, A. A. (2021). A gait energy image-based system for Brazilian sign language recognition. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *68*(11), 4761–4771.

Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., & Jagersand, M. (2020). U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition*, *106*, Article 107404.

Raghuveera, T., Deepthi, R., Mangalashri, R., & Akshaya, R. (2020). A depth-based Indian sign language recognition using microsoft kinect. *Sādhanā*, *45*(1), 34.

Rastgoo, R., Kiani, K., & Escalera, S. (2021a). Hand pose aware multimodal isolated sign language recognition. *Multimedia Tools and Applications*, *80*(1), 127–163.

Rastgoo, R., Kiani, K., & Escalera, S. (2021b). Sign language recognition: A deep survey. *Expert Systems with Applications*, *164*, Article 113794.

Rastgoo, R., Kiani, K., Escalera, S., & Sabokrou, M. (2021). Sign language production: a review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3451–3461).

Rezende, T. M., Almeida, S. G. M., & Guimarães, F. G. (2021). Development and validation of a brazilian sign language database for human gesture recognition. *Neural Computing and Applications*, 1–19.

Rezende, T. M., Castro, C., & Almeida, S. (2016). An approach for Brazilian Sign Language (BSL) recognition based on facial expression and k-NN classifier. In *29th SIBGRAPI, workshop on face processing applications, on proceedings* (pp. 1–2).

Santhalingam, P. S., Pathak, P., Košecká, J., Rangwala, H., et al. (2019). Sign language recognition analysis using multimodal data. In *2019 IEEE international conference on data science and advanced analytics* DSAA, (pp. 203–210). IEEE.

Santos, A. S., & Portes, A. J. F. (2019). Perceptions of deaf subjects about communication in primary health care. *Revista Latino-Americana de Enfermagem*, *27*, http://dx.doi.org/10.1590/1518-8345.2612.3127.

Seredin, O., Kopylov, A., Huang, S.-C., & Rodionov, D. (2019). A skeleton features-based fall detection using microsoft kinect v2 with one class-classifier outlier removal. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.

Sharma, S., & Singh, S. (2021). Vision-based hand gesture recognition using deep learning for the interpretation of sign language. *Expert Systems with Applications*, *182*, Article 115657.

Sridhar, A., Ganesan, R. G., Kumar, P., & Khapra, M. (2020). Include: A large scale dataset for indian sign language recognition. In *Proceedings of the 28th ACM international conference on multimedia* (pp. 1366–1375).

Stergiou, A., Kapidis, G., Kalliatakis, G., Chrysoulas, C., Veltkamp, R., & Poppe, R. (2019). Saliency tubes: Visual explanations for spatio-temporal convolutions. In *2019 IEEE international conference on image processing* ICIP, (pp. 1830–1834). IEEE.

Tahir, Y., Chakraborty, D., Maszczyk, T., Dauwels, S., Dauwels, J., Thalmann, N., et al. (2015). Real-time sociometrics from audio-visual features for two-person dialogs. In *2015 IEEE international conference on digital signal processing* DSP, (pp. 823–827). IEEE.

Tyrone, M. E., & Mauk, C. E. (2010). Sign lowering and phonetic reduction in American Sign Language. *Journal of Phonetics*, *38*(2), 317–328.

Vahdani, E., Jing, L., Huenerfauth, M., & Tian, Y. (2022). Recognizing American sign language manual signs from Rgb-D videos. Available at SSRN 4019317.

Venugopalan, A., & Reghunadhan, R. (2021). Applying deep neural networks for the automatic recognition of sign language words: A communication aid to deaf agriculturists. *Expert Systems with Applications*, *185*, Article 115601.

Vogler, C., & Metaxas, D. N. (2003). *American sign language recognition: reducing the complexity of the task with phoneme-based modeling and parallel hidden markov models* (Ph.D. thesis), Citeseer.

Wadhawan, A., & Kumar, P. (2021). Sign language recognition systems: A decade systematic literature review. *Archives of Computational Methods in Engineering*, *28*(3), 785–813.

Wan, J., Zhao, Y., Zhou, S., Guyon, I., Escalera, S., & Li, S. Z. (2016). Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 56–64).

Wang, H., Leu, M. C., & Oz, C. (2006). American sign language recognition using multi-dimensional hidden Markov models. *Journal of Information Science and Engineering*, *22*(5), 1109–1123.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8798–8807).

Wei, S.-E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4724–4732).

Yan, J., Zhou, M., Pan, J., Yin, M., & Fang, B. (2022). Recent advances in 3D human pose estimation: From optimization to implementation and beyond. *International Journal of Pattern Recognition and Artificial Intelligence*, Article 2255003.

Yang, S., Jung, S., Kang, H., & Kim, C. (2020). The Korean sign language dataset for action recognition. In *International conference on multimedia modeling* (pp. 532–542). Springer.

Zeiler, M. D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint arXiv:1301.3557.

Zhang, L., Zhang, Y., & Zheng, X. (2020). WiSign: Ubiquitous American sign language recognition using commercial wi-fi devices. *ACM Transactions on Intelligent Systems and Technology*, *11*(3), 1–24.

Frederico G. Guimarães received the B.Eng., M.Sc., and Ph.D. degrees in Electrical Engineering from the Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2003, 2004, and 2008, respectively. In 2010, he joined the Department of Electrical Engineering, UFMG, where he became an Associate Professor in 2018. Since 2014, he has been responsible for the Machine Intelligence and Data Science Laboratory (MINDS) for computational intelligence research. He also had a Postdoctoral Fellowship with the Laboratoire Images, Signaux et Systèmes Intelligents (LiSSi), Université Paris-Est Créteil (UPEC), Paris, France, from 2017 to 2018. He has published more than 200 papers in journals, congresses, and chapters of national and international books. He has experience in electrical engineering and computer engineering, with an emphasis on optimization, computational intelligence, machine learning, time series forecasting, genetic algorithms, and evolutionary computation. Dr. Guimarães is a member of the IEEE Computational Intelligence Society (CIS), IEEE Systems Council and the IEEE Systems, Man, and Cybernetics Society (SMCS). He is also an Associate Editor of the journals IEEE Access, Engineering Applications of Artificial Intelligence (Elsevier) and Neurocomputing (Elsevier).