Word separation in continuous sign language using isolated signs and post-processing

Razieh Rastgoo[a] (rrastgoo@semnan.ac.ir), Kourosh Kiani[a] (Kourosh.kiani@semnan.ac.ir), Sergio Escalera[b] (sergio@maia.ub.es)

[a] Electrical and Computer Engineering Department, Semnan University, Semnan, 3513119111, Iran
[b] Department of Mathematics and Informatics, Universitat de Barcelona and Computer Vision Center, 585, 08007 Barcelona, Spain

Corresponding Author:
Dr. Kourosh Kiani
Electrical and Computer Engineering Department, Semnan University, Semnan, 3513119111, Iran
Tel.: +989122361274, Fax: +98-23-333-21005
Email: Kourosh.kiani@semnan.ac.ir

Word separation in continuous sign language using isolated signs and post-processing

Razieh Rastgoo[a], Kourosh Kiani[a,*], Sergio Escalera[b]

[a]*Electrical and Computer Engineering Department, Semnan University, Semnan, 3513119111, Iran*
[b]*Department of Mathematics and Informatics, Universitat de Barcelona and Computer Vision Center, 585, 08007 Barcelona, Spain*

Abstract

Continuous Sign Language Recognition (CSLR) is a long challenging task in Computer Vision due to the difficulties in detecting the explicit boundaries between the words in a sign sentence. To deal with this challenge, we propose a two-stage model. In the first stage, the predictor model, which includes a combination of CNN, SVD, and LSTM, is trained with the isolated signs. In the second stage, we apply a post-processing algorithm to the Softmax outputs obtained from the first part of the model in order to separate the isolated signs in the continuous signs. Due to the lack of a large dataset, including both the sign sequences and the corresponding isolated signs, two public datasets in Isolated Sign Language Recognition (ISLR), RKS-PERSIANSIGN and ASLVID, are used for evaluation. Results of the continuous sign videos confirm the efficiency of the proposed model to deal with isolated sign boundaries detection.

Keywords: Continuous Sign Language Recognition (CSLR), Isolated Sign Language Recognition (ISLR), Word separation, Sign boundaries, Transfer learning

1. Introduction

Most of the hearing-impaired people employ a sign language for communication. Information in sign language can be in the form of manual hand gestures, hand movements, body postures, and facial expressions. However, the hearing majority and also a part of the hearing-impaired community, do not know sign language. Due to the considerable amount of population of the hearing-impaired people around the world, many researchers have shown interest in developing intelligent and automatic sign language translators to facilitate the communication between the deaf community and the hearing majority. Furthermore, such

[*] Corresponding author.

Email addresses: `rrastgoo@semnan.ac.ir` (Razieh Rastgoo), `Kourosh.kiani@semnan.ac.ir` (Kourosh Kiani), `sergio@maia.ub.es` (Sergio Escalera) translators can provide equal communication opportunities and improve public welfare for the hearing-impaired community. In line with this requirement, we propose a simple yet efficient model to facilitate the translation task of the continuous signs by accurately detecting the isolated signs in a continuous sign [25].

Generally, Sign Language Recognition (SLR) is categorized into two groups: Isolated Sign Language Recognition (ISLR) and Continuous Sign Language Recognition (CSLR). According to the Wadhawan and Kumar (2019) study, most of the available models were proposed for ISLR [27]. However, there are still some challenges in ISLR such as high occlusions of hands, fast hands movement, background complexity, inexistence of the large and diverse datasets, varying illumination conditions, different hand gestures, and complex interactions between hands and objects [24]. In addition to the challenges in ISLR, there are also some challenges in CSLR. The most important one is detecting the isolated sign boundaries in a sign sequence. Hence, it is challenging to perform temporal segmentation on a sign sequence since there are no such explicit boundaries between the isolated signs of a sign sequence in any dataset [12]. Another challenge is the varying length of the signing during different sentences that needs to be handled in the proposed models. In this paper, we propose a framework to solve these challenges.

Recently, Deep Learning approaches have obtained state-of-the-art performance in various tasks [24], especially SLR [7]. However, Deep Learning-based models require multiple instances of labeled sequence data to enable end-to-end sequence recognition. Since there are few public datasets for CSLR, it could be useful to employ the knowledge obtained by learning the model on isolated words to improve the training process for CSLR. In the case that the labeled sentence data is not readily available, a transfer learning mechanism can be used to facilitate the training process. Considering this, in this paper, we propose a simple yet efficient post-processing algorithm to transfer the knowledge of the trained model on the isolated signs into the CSRL problem.

The remaining paper is organized as follows. Related literature on deep models in CSLR, considering the transfer learning mechanism, is reviewed in section 2. Details of the proposed methodology are described in section 3. Results are presented and discussed in section 4. Finally, the work is discussed and concluded in section 5.

2. Related work

Here, we briefly review recent works in CSLR considering transfer learning. With the advent of deep learning in recent years, many recent approaches achieved state-of-the-art performance using the combination of different deep learning-based models, such as CNN and RNN [4, 5, 20, 25, 22, 23, 16, 17, 21, 1, 9, 2, 14, 15, 27, 6, 8, 11, 26, 10, 19, 18]. More specifically, while many advancements have been obtained in ISLR and CSLR with the capabilities of deep learning-based models, some challenges in both tasks still need to be discussed.

For instance, the challenge of detecting the isolated sign boundaries in a continuous sign is one of them. Generally, recognizing unseen continuous signs with different sequential patterns is hard for a trained network. Furthermore, training such models is generally non-trivial, as most of them require pre-training and incorporating an iterative training strategy [12], which greatly lengthens the training process. Transferring the advancements obtained in ISLR into CSLR can be a useful solution for this challenge. However, some models do not consider the transfer mechanism. In this way, we briefly review recent models in two categories:

- **Transfer learning-based models:** Different transfer methodologies have been defined and used in deep learning-based models and applications, such as hand gesture recognition [3], sign language recognition [11, 1, 2], and speech emotion recognition [28]. Using the multi-modal data for improving the recognition accuracy of a deep model is the main idea of the proposed model by Bird et al. The proposed model aimed to transfer the knowledge learned from the bigger dataset of British Sign Language (BSL) to the target model [2]. Halvardsson et al. [10] transferred the knowledge of the first 20 layers of the pre-trained InceptionV3 model into a new model for static Swedish Sign Language (SSL) recognition. Furthermore, they included some new layers on top of the frozen layers, obtaining a classification accuracy of 85%İn another work, Jiang et al. made various transfer learning configurations on the later layers of AlexNet using fingerspelling images from the Chinese sign language. Results showed that the highest obtained recognition accuracy was 89.48% [11]. Sharma et al. used six tri-axis accelerometers and gyroscopes, placed on both hands of the signer, to record some isolated and continuous signs. They proposed a CNN-BiLSTM-CTC network and trained on the isolated word sign dataset. They transferred the knowledge of the model to the continuous samples and analyzed the performance of the model on their own dataset [26]. In line with the transfer learning mechanism used in these models, in this paper, we propose a simple yet efficient post-processing model for the separation of the isolated signs into a continuous sign. We transfer the obtained knowledge from the trained model on the isolated signs into the continuous signs, relying on large datasets with a large number of samples in each class.

- **Other models:** The models in this category focus on different feature extractor models, especially deep learning-based models. Cheng et al. proposed a fully convolutional network (FCN) for CSLR. To this end, a Gloss Feature Enhancement (GFE) module was proposed for sequence alignment learning. Results on two datasets, Chinese Sign Language (CSL) and RWTH-PHOENIX-Weather- 2014 (RWTH), show that the model outperforms state-of-the-art models in the field [6]. Camgoz et al. proposed a model to simultaneously consider the alignment and recognition tasks in CSLR. To this end, they defined some expert systems, namely SubUNets, to use the spatio- temporal relationships between these SubUNets for modeling the tasks. Results on the RWTH- PHOENIX-Weather-2014 dataset show an accuracy improvement compared to state-of-the-art models in CSLR [4]. Papastratis et al. introduce a generative-based model for CSLR using a Generative Adversarial Network (GAN) architecture, namely Sign Language Recognition Generative Adversarial Network (SLRGAN). The spatio-temporal features extracted from video sequences are fed into the Bidirectional Long Short-Term Memory (Bi-LSTM) module of the generator to improve the recognition accuracy of the model. Results on three datasets, RWTH-Phoenix-Weather-2014, Chinese Sign Language (CSL), and Greek Sign Language (GSL) Signer Independent (SI), show relative performance improvements of 0.5 % 0.3 % and 1.26 % respectively [15].

3. Proposed model

In this section, we present the proposed model, which is illustrated in Figure 1.

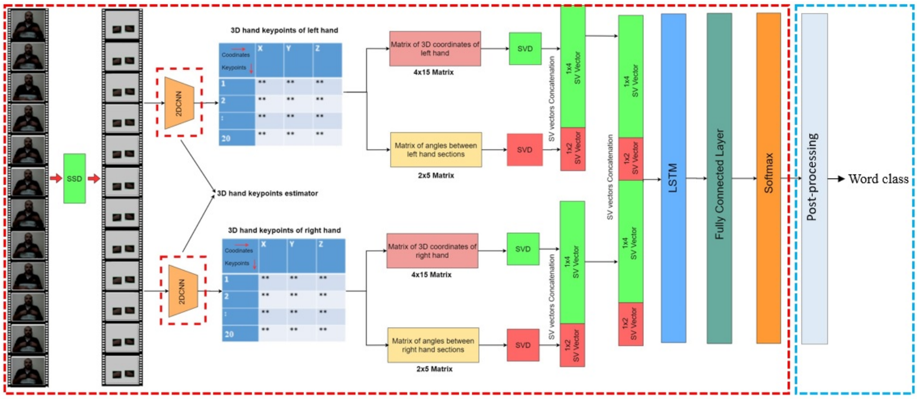

SSD
2DCNN
3D hand keypoints of left hand
Coodinates
Keypoints
X
Y
Z
1
2
:
20
**
3D hand keypoints estimator
3D hand keypoints of right hand
Matrix of 3D coordinates of left hand
4x15 Matrix
Matrix of angles between left hand sections
2x5 Matrix
Matrix of 3D coordinates of right hand
4x15 Matrix
Matrix of angles between right hand sections
2x5 Matrix
SVD
SV vectors Concatenation
1x4 SV Vector
1x2 SV Vector
LSTM
Fully Connected Layer
Softmax
Post-processing
Word class

Figure 1: An overview of the proposed model

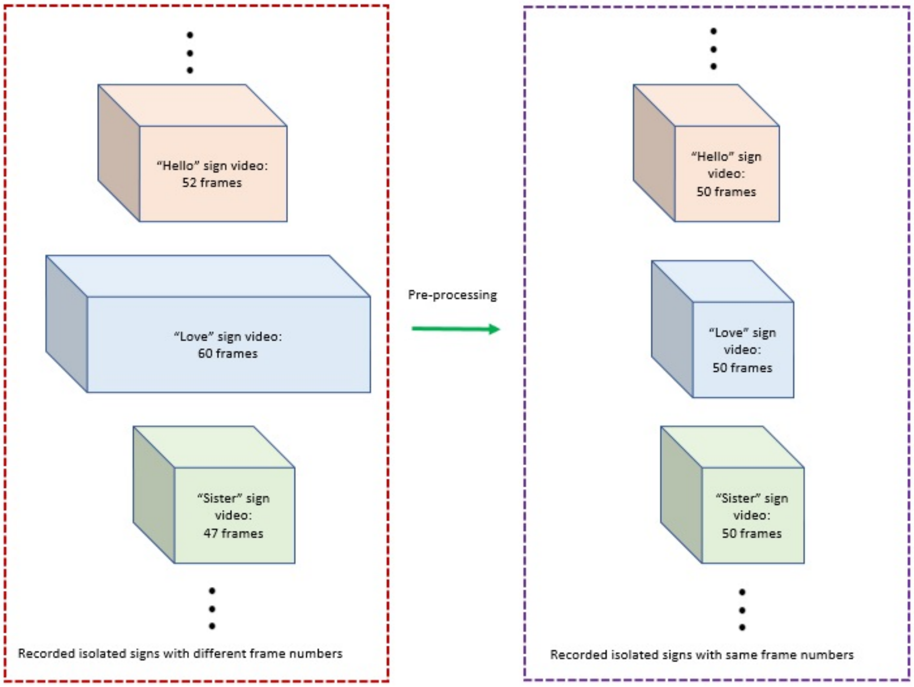
3.1. Problem definition

Let $\{V_{train} = \{(x^1_{train}, y^1_{train}), (x^2_{train}, y^2_{train}), ..., (x^N_{train}, y^N_{train})\}$ denote a set of N pairs of isolated sign video x and the corresponding class label y of the seen data during training, with the subscript standing for training data. In a similar way, let $\{V_{test} = \{(x^1_{test}, y^1_{test}), (x^2_{test}, y^2_{test}), ..., (x^M_{test}, y^M_{test})\}$ denote a set of M pairs of isolated sign video x and the corresponding class label y of the unseen data during testing, with the subscript standing for test data. After the training and testing of the model, we feed a continuous sign video to the model in order to recognize and separate the isolated signs in the input sign sequence.

3.2. Model

Here, we describe the details of the proposed model. As stated in the previous section, the main challenge in CSLR is the boundary detection of the sign words in a continuous sign video. To solve this challenge, we propose to use transfer learning to employ the knowledge of the trained model on the isolated sign data to the continuous sign videos. More concretely, the proposed model includes the following parts:

- **The predictor model:** While the extracted features from the predictor model can be obtained using any hand-crafted or end-to-end models, we borrowed this part from our previous work in ISLR [25]. In this model, we use hand-crafted SVD features to feed to a LSTM Network. In order to prepare datasets for training and test the predictor model, we apply a pre-processing step to the recorded isolated sign videos with different frame numbers to have equal frame numbers in all isolated sign videos. Figure 2 shows this pre-processing step.



"Hello" sign video: 52 frames
"Love" sign video: 60 frames
"Sister" sign video: 47 frames
Recorded isolated signs with different frame numbers
Pre-processing
"Hello" sign video: 50 frames
"Love" sign video: 50 frames
"Sister" sign video: 50 frames
Recorded isolated signs with same frame numbers

Figure 2: Equalizing the frame numbers in all isolated sign videos

To apply the predictor model to a sign sequence, we need the sign sequences, including the isolated signs. Due to the lack of such dataset, including both the sign sequences and the corresponding isolated signs, we make these sequences by concatenating the isolated sign videos without any pre-processing. Figure 3 shows this step schematically.

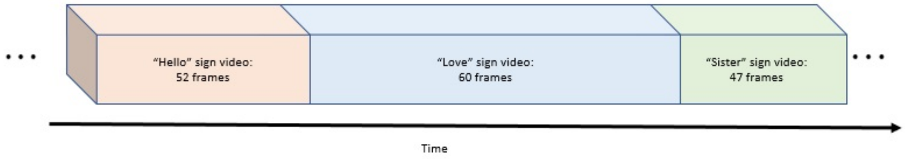


Figure 3: Building a sign sequence by concatenating the isolated signs without any pre-processing

- **Post-processing:** In this part, we apply a post-processing algorithm to the Softmax outputs obtained from the predictor model in order to separate the isolated signs in a sign sequence.

3.3. Details of the post-processing method

We apply a sliding window with a window size of 50 frames and feed them one by one to the hand pose estimator and also the SVD feature extractor. Then, LSTM Network (many-to-one) maps the SVD features sequence corresponding to 50 frames into a single vector. After that, this vector is passed to a Fully Connected (FC) layer. Finally, a Softmax layer is applied to the FC outputs. If the recognition accuracy of the Softmax layer for an isolated sign is higher than a predefined threshold, we accept it as a recognized isolated sign. We assign a value of 0.51 for this threshold in our experiments. The reason behind this assumption is that we generally have only one class above this threshold in each sliding window. The sliding window with stride one runs through incoming video frames.

Once an isolated sign ("Hello") is recognized (Figure 4), the immediate recognition of the same sign ("Hello") is assigned to the "Blank" class, as shown in Figure 5.

If all outputs of the Softmax layer are less than the predefined threshold (0.51), the post-processing algorithm assigns a "Blank" label to the current sliding window (Figure 6).

Once the recognized sign in the current window is the same as the already recognized sign class followed by only one or some "Blank" classes, we assign this recognized class to the "Blank" (Figure 7). It is worth to mentioning that in order to achieve a higher accuracy, we ignore the repeated sing words.

We apply a sliding window across the video frames. In this way, the second sign class is recognized, as shown in Figure 8.

Once an isolated sign ("Love") is recognized, the immediate recognition of the same sign ("Love") is assigned to the "Blank" class, as shown in Figure 9.

The sliding processing goes on and the post-processing algorithm accepts the new sign class ("Sister"), as shown in Figure 10.

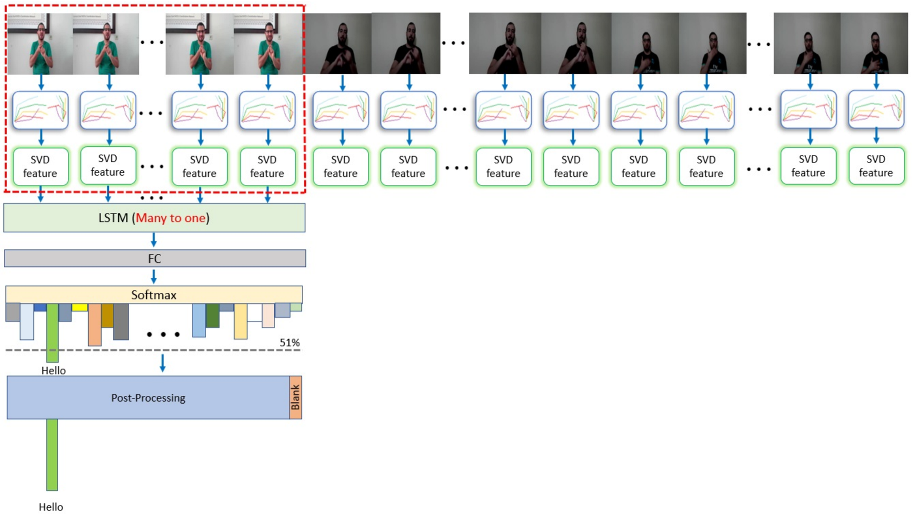


Figure 4: The first recognition of the "Hello" sign, which post-processing algorithm accepts it.

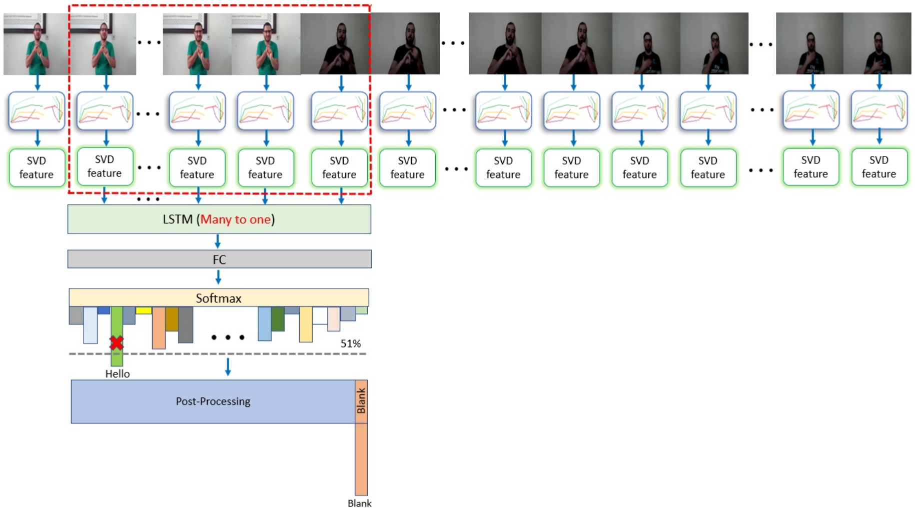


Figure 5: The immediate recognition of the "Hello" sign, which is assigned to the "Blank" label using the post-processing algorithm.

The "Blank" is assigned to the repeated classes, as we mentioned above for other classes (Figure 11).

In the final step, we only reserve one "Blank" between the recognized sign

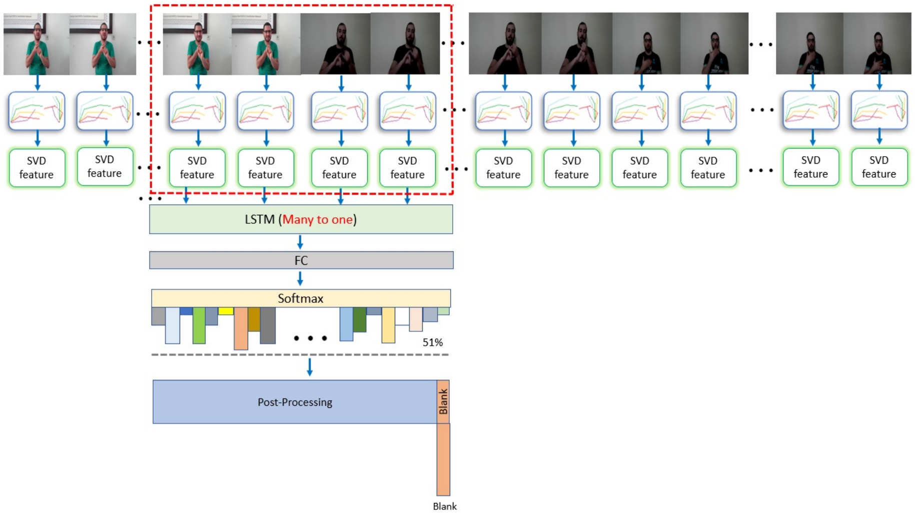


Figure 6: The post-processing algorithm assigns a "Blank" label if all outputs of the Softmax layer are less than 0.51.

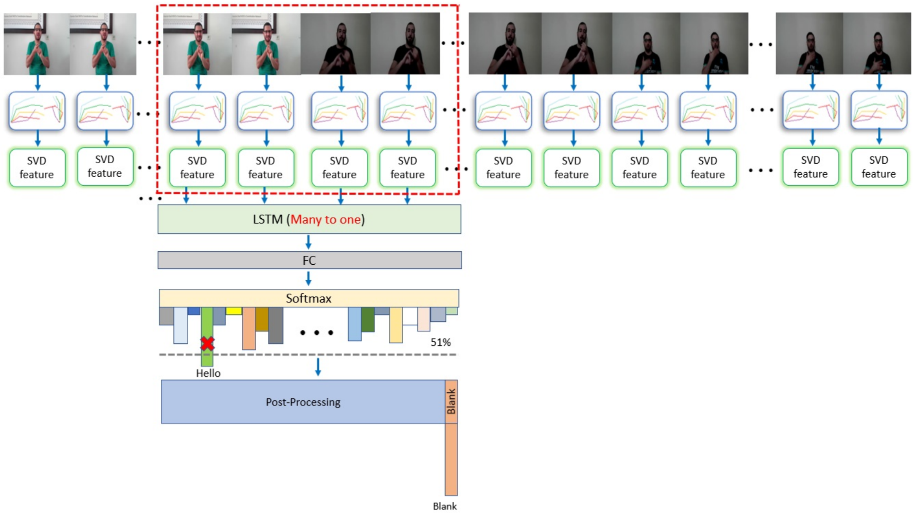


Figure 7: Assigning a "Blank" to the repeated sign class

classes and remove the repeated "Blank" classes. Here, the final result is ready to convert to the voice. Figure 12 shows an overview of the post-processing algorithm.

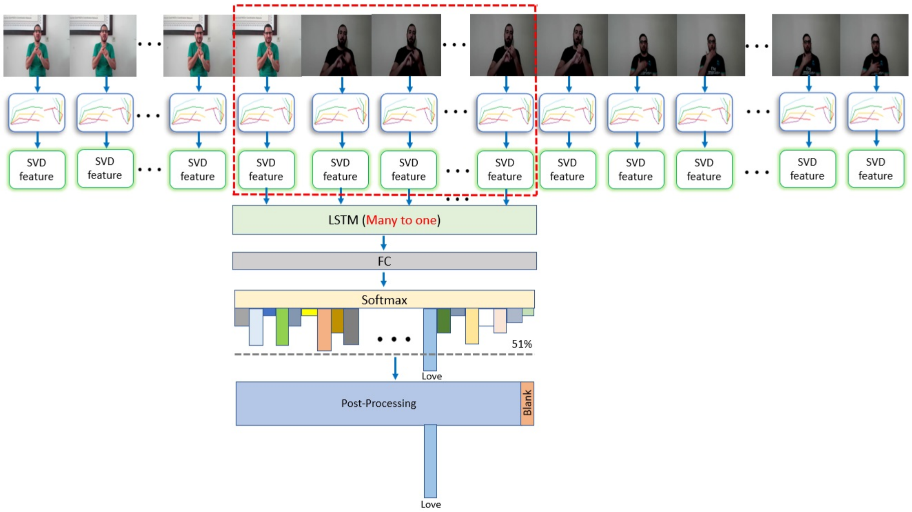


Figure 8: Recognition of the "Love" class

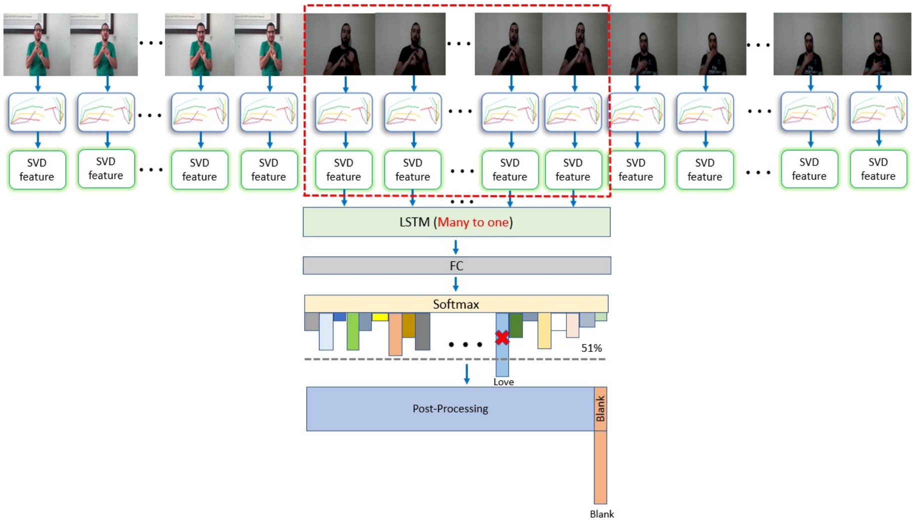


Figure 9: The immediate recognition of the "Love" sign assigned to the "Blank" label.

4. Experimental results

In this section, we present details of the datasets and results.

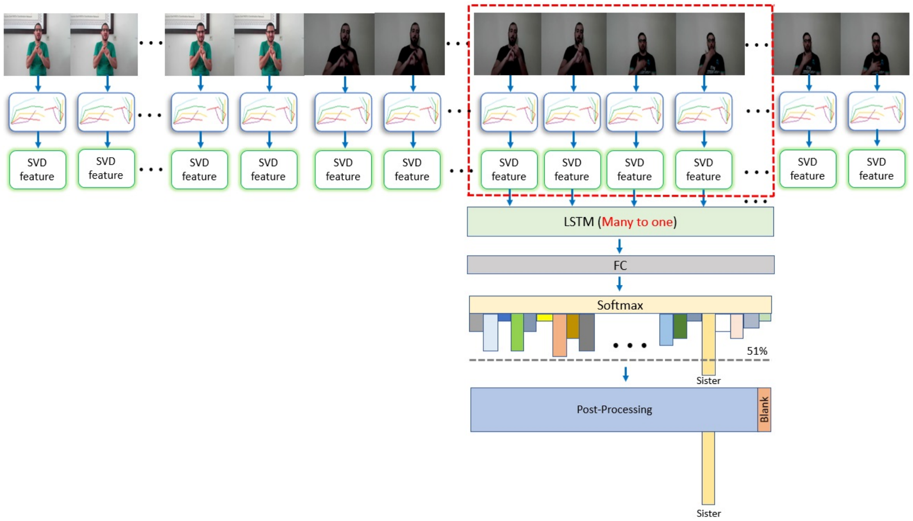


Figure 10: The new sign class (”Sister”) is accepted by the post-processing algorithm.

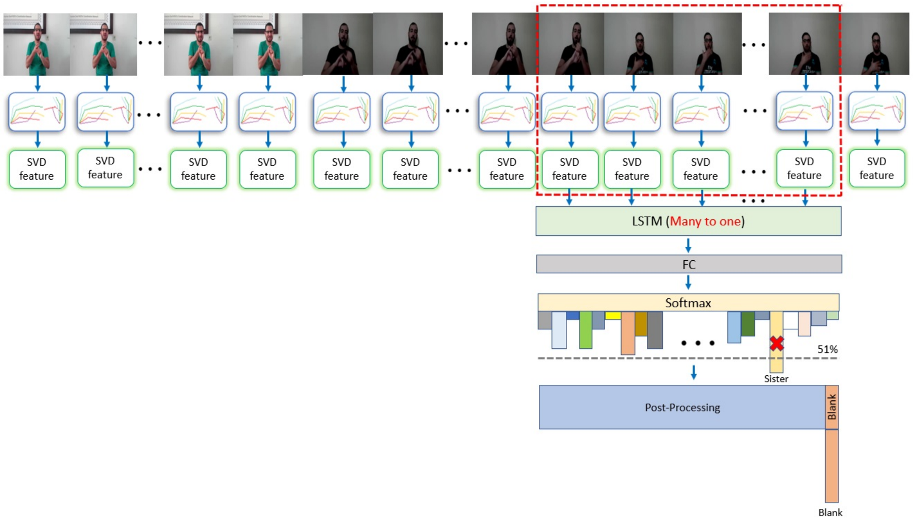


Figure 11: Assigning the ”Blank” to the repeated sign classes.

4.1. Datasets

The available datasets in CSLR do not include both the sign streams and the corresponding isolated signs for each stream. So, to tackle this challenge, two datasets in ISLR are used for evaluation, RKS-PERSIANSIGN [21] and ASLVID

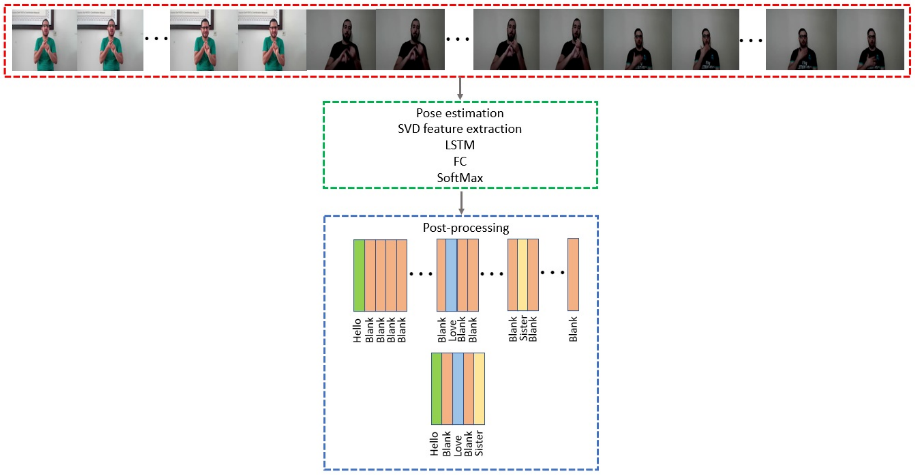


Figure 12: Assigning the "Blank" to the repeated sign classes.

[13]. The first dataset includes 10'000 RGB videos of 100 Persian signs using 10 contributors, 5 women and 5 men, in a simple background with a maximum distance of 1.5 meter between the contributor and camera. There are 100 video samples for each Persian sign word. The second dataset, ASLVID, contains some American sign words with their corresponding annotations. There are different sample numbers in each class label. We selected the 100 sign categories which contain at least seven video samples. We pre-process these datasets to have equally frame numbers for all video samples during the training phase. However, we do not perform any pre-processing on the test data and use them with different frame numbers. Table 1 shows details of these datasets.

	RKS-PERSIANSIGN	**ASLVID**
Isolated sign video numbers per class	100	7
Total isolated sign video numbers	10'000	700
Total continuous sign videos	100	7
Subjects per class	10	-
Video sign language	Persian	American

Table 1: Details of datasets used in evaluation

4.2. Implementation details

Our evaluation has been done on an Intel(R) Xeon(R) CPU E5-2699 (2 processors) with 90GB RAM with Microsoft Windows 10 operating system and Python software with NVIDIA Tesla K80 GPU. We implemented our model in the Keras library. We use Adam optimizer with a mini-batch size of 50.

The learning rate starts from 0.005 and is divided by 10 every 10 epochs. The proposed model is trained for a total of 200 epochs with early stopping. In addition, we use a weight decay of 1e-4 and a momentum of 0.92. We used two datasets for the evaluation: RKS-PERSIANSIGN and ASLVID, which are divided randomly into training (80 %) and testing (20 %) sets. Table 2 shows the details of the experimentally set model parameters.

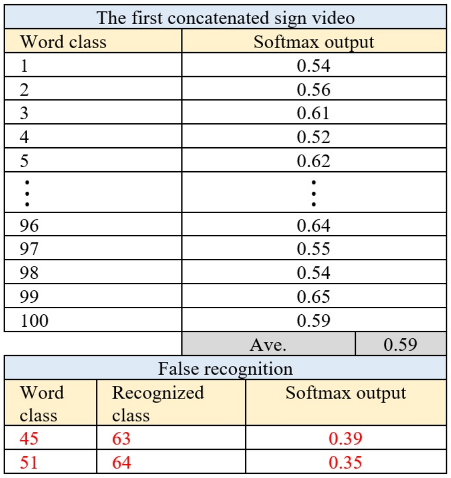
Parameter	Value	Parameter	Value
Weight decay	1e-4	Epoch numbers	200
Learning rate	0.005	Number of frames per video sample	50
Batch size	50	Processing way	GPU
Keypoint dimension	21x3	Number of singular values	12

Table 2: Details of the parameters used in the proposed model

4.3. Ablation analysis

To analyze the impact of the post-processing methodology, we present our ablation analysis on two datasets used for evaluation (Table 3-12). As these tables show, there are some false recognition in the two datasets due to the similarities between the signs in the datasets. The average of recognized Softmax outputs on RKS-PERSIANSIGN and ASLVID are 0.98 and 0.59, respectively.

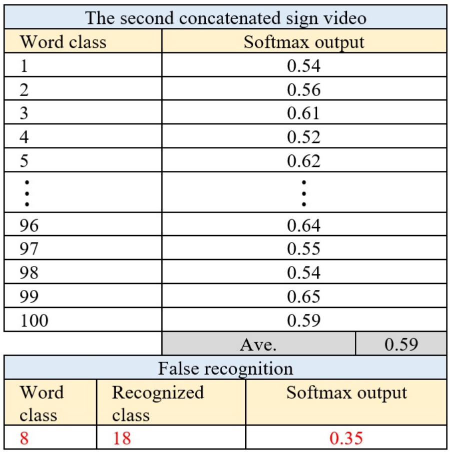
Table 3: The first concatenated sign video of the ASLVID dataset.



The first concatenated sign video		
Word class	Softmax output	
1	0.54	
2	0.56	
3	0.61	
4	0.52	
5	0.62	
⋮	⋮	
96	0.64	
97	0.55	
98	0.54	
99	0.65	
100	0.59	
	Ave.	0.59

False recognition		
Word class	Recognized class	Softmax output
45	63	0.39
51	64	0.35

Table 4: The second concatenated sign video of the ASLVID dataset.



The second concatenated sign video		
Word class	Softmax output	
1	0.54	
2	0.56	
3	0.61	
4	0.52	
5	0.62	
⋮	⋮	
96	0.64	
97	0.55	
98	0.54	
99	0.65	
100	0.59	
	Ave.	0.59

False recognition		
Word class	Recognized class	Softmax output
8	18	0.35

Table 5: The third concatenated sign video of the ASLVID dataset.

The third concatenated sign video	
Word class	Softmax output
1	0.54
2	0.56
3	0.61
4	0.52
5	0.62
⋮	⋮
96	0.64
97	0.55
98	0.54
99	0.65
100	0.59
Ave.	0.59

False recognition		
Word class	Recognized class	Softmax output
45	63	0.40
50	80	0.45

Table 6: The fourth concatenated sign video of the ASLVID dataset.

The fourth concatenated sign video	
Word class	Softmax output
1	0.54
2	0.56
3	0.61
4	0.52
5	0.62
⋮	⋮
96	0.64
97	0.55
98	0.54
99	0.65
100	0.59
Ave.	0.59

False recognition		
Word class	Recognized class	Softmax output
45	63	0.38

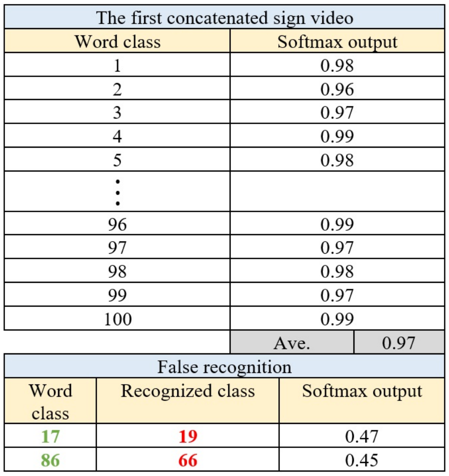
Table 7: Details of the recognition accuracy of the proposed post-processing algorithm on the ASLVID dataset.

Concatenated sign video	Avg of Softmax output of Recognized class	Ground truth Word class	Softmax output of Ground truth Word class	Recognized class	Softmax output of Recognized class
1	**0.59**	**45**	**0.37**	**63**	**0.39**
		51	**0.33**	**64**	**0.35**
2	**0.59**	**8**	**0.36**	**18**	**0.38**
3	**0.59**	**45**	**0.38**	**63**	**0.40**
		50	**0.44**	**80**	**0.45**
4	**0.59**	**45**	**0.37**	**63**	**0.39**
5	**0.59**	**45**	**0.37**	**63**	**0.39**
		64	**0.33**	**51**	**0.35**
6	**0.59**	**18**	**0.34**	**8**	**0.36**
		63	**0.35**	**45**	**0.37**
7	**0.59**	**45**	**0.33**	**63**	**0.35**
		50	**0.46**	**80**	**0.48**

5. Discussion and conclusion

In this work, we proposed a simple yet efficient post-processing methodology for the separation of the isolated signs in a continuous sign video, as a long challenging task in Computer Vision. Due to the lack of a continuous sign dataset including both the continuous signs and the corresponding isolated signs, we use the datasets in isolated sign language and concatenate them to make the continuous sign videos. Furthermore, since there is no similar work to ours, we

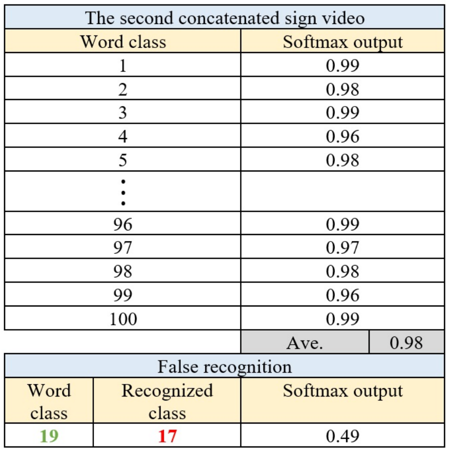
Table 8: The first concatenated sign video of the RKS-PERSIANSIGN dataset.



The first concatenated sign video	
Word class	Softmax output
1	0.98
2	0.96
3	0.97
4	0.99
5	0.98
⋮	
96	0.99
97	0.97
98	0.98
99	0.97
100	0.99
	Ave. 0.97

False recognition		
Word class	Recognized class	Softmax output
17	19	0.47
86	66	0.45

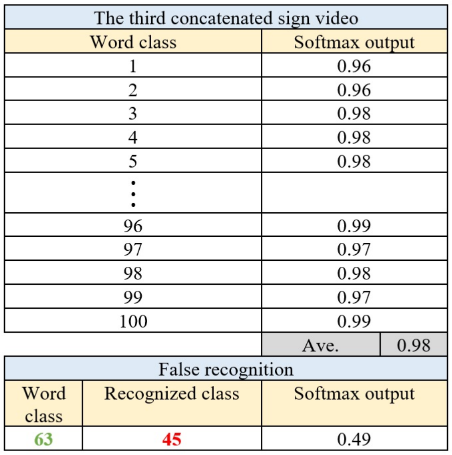
Table 9: The second concatenated sign video of the RKS-PERSIANSIGN dataset.



The second concatenated sign video	
Word class	Softmax output
1	0.99
2	0.98
3	0.99
4	0.96
5	0.98
⋮	
96	0.99
97	0.97
98	0.98
99	0.96
100	0.99
	Ave. 0.98

False recognition		
Word class	Recognized class	Softmax output
19	17	0.49

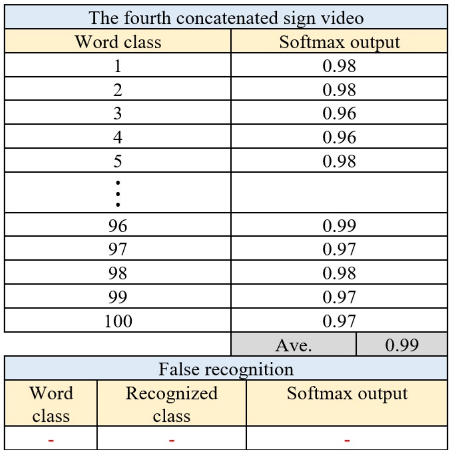
Table 10; The third concatenated sign video of the RKS-PERSIANSIGN dataset.



The third concatenated sign video	
Word class	Softmax output
1	0.96
2	0.96
3	0.98
4	0.98
5	0.98
⋮	
96	0.99
97	0.97
98	0.98
99	0.97
100	0.99
	Ave. 0.98

False recognition		
Word class	Recognized class	Softmax output
63	45	0.49

Table 11: The fourth concatenated sign video of the RKS-PERSIANSIGN dataset.

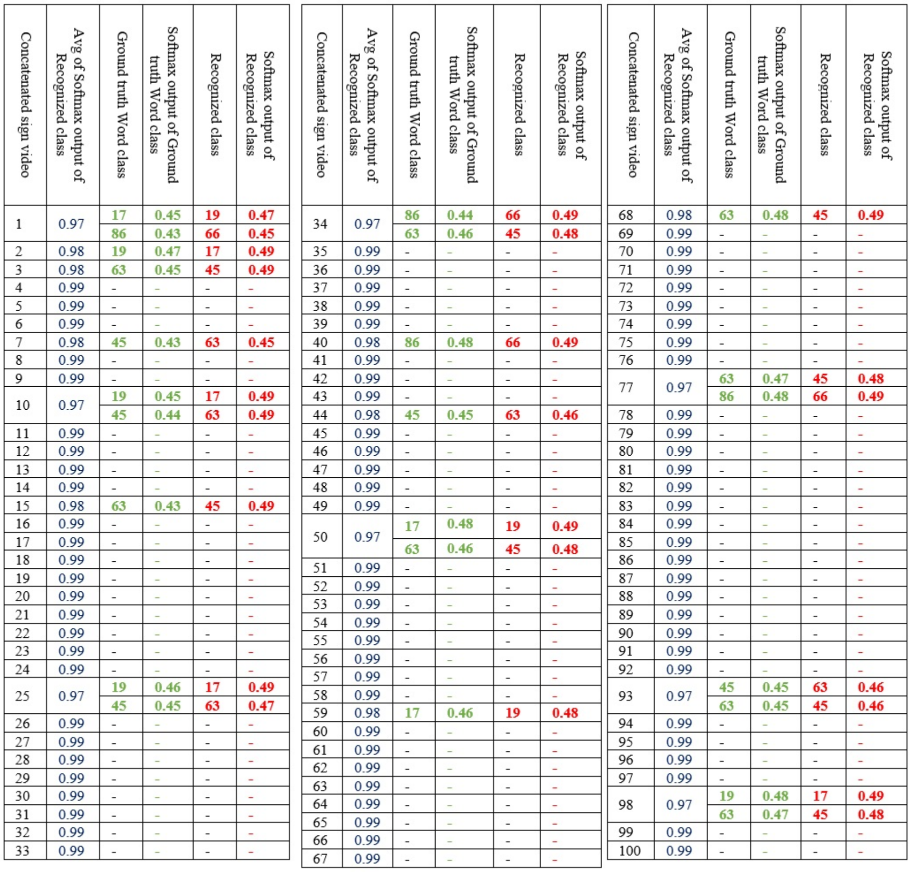


The fourth concatenated sign video	
Word class	Softmax output
1	0.98
2	0.98
3	0.96
4	0.96
5	0.98
⋮	
96	0.99
97	0.97
98	0.98
99	0.97
100	0.97
	Ave. 0.99

False recognition		
Word class	Recognized class	Softmax output
-	-	-

cannot compare the proposed model with other models. The proposed methodology used a predefined threshold, 0.51, to accept or reject a recognized class in the current sliding window. The intuition behind this predefined value is that we generally can have only one class above this threshold in the current sliding window. This comes from the property of the Softmax layer that gives the output probabilities of all classes sum to 1. We aim to suppress the false recognized classes and assign a “Blank” label to them. Considering this, the

Table 12: Details of the recognition accuracy of the proposed postprocessing algorithm on the RKS-PERSIANSIGN dataset.



Concatenated sign video	Avg of Softmax output of Recognized class	Ground truth Word class	Softmax output of Ground truth Word class	Recognized class	Softmax output of Recognized class
1	0.97	17 86	0.45 0.43	19 66	0.47 0.45
2	0.98	19	0.47	17	0.49
3	0.98	63	0.45	45	0.49
4	0.99	-	-	-	-
5	0.99	-	-	-	-
6	0.99	-	-	-	-
7	0.98	45	0.43	63	0.45
8	0.99	-	-	-	-
9	0.99	-	-	-	-
10	0.97	19 45	0.45 0.44	17 63	0.49 0.49
11	0.99	-	-	-	-
12	0.99	-	-	-	-
13	0.99	-	-	-	-
14	0.99	-	-	-	-
15	0.98	63	0.43	45	0.49
16	0.99	-	-	-	-
17	0.99	-	-	-	-
18	0.99	-	-	-	-
19	0.99	-	-	-	-
20	0.99	-	-	-	-
21	0.99	-	-	-	-
22	0.99	-	-	-	-
23	0.99	-	-	-	-
24	0.99	-	-	-	-
25	0.97	19 45	0.46 0.45	17 63	0.49 0.47
26	0.99	-	-	-	-
27	0.99	-	-	-	-
28	0.99	-	-	-	-
29	0.99	-	-	-	-
30	0.99	-	-	-	-
31	0.99	-	-	-	-
32	0.99	-	-	-	-
33	0.99	-	-	-	-
34	0.97	86 63	0.44 0.46	66 45	0.49 0.48
35	0.99	-	-	-	-
36	0.99	-	-	-	-
37	0.99	-	-	-	-
38	0.99	-	-	-	-
39	0.99	-	-	-	-
40	0.98	86	0.48	66	0.49
41	0.99	-	-	-	-
42	0.99	-	-	-	-
43	0.99	-	-	-	-
44	0.98	45	0.45	63	0.46
45	0.99	-	-	-	-
46	0.99	-	-	-	-
47	0.99	-	-	-	-
48	0.99	-	-	-	-
49	0.99	-	-	-	-
50	0.97	17 63	0.48 0.46	19 45	0.49 0.48
51	0.99	-	-	-	-
52	0.99	-	-	-	-
53	0.99	-	-	-	-
54	0.99	-	-	-	-
55	0.99	-	-	-	-
56	0.99	-	-	-	-
57	0.99	-	-	-	-
58	0.99	-	-	-	-
59	0.98	17	0.46	19	0.48
60	0.99	-	-	-	-
61	0.99	-	-	-	-
62	0.99	-	-	-	-
63	0.99	-	-	-	-
64	0.99	-	-	-	-
65	0.99	-	-	-	-
66	0.99	-	-	-	-
67	0.99	-	-	-	-
68	0.98	63	0.48	45	0.49
69	0.99	-	-	-	-
70	0.99	-	-	-	-
71	0.99	-	-	-	-
72	0.99	-	-	-	-
73	0.99	-	-	-	-
74	0.99	-	-	-	-
75	0.99	-	-	-	-
76	0.99	-	-	-	-
77	0.97	63 86	0.47 0.48	45 66	0.48 0.49
78	0.99	-	-	-	-
79	0.99	-	-	-	-
80	0.99	-	-	-	-
81	0.99	-	-	-	-
82	0.99	-	-	-	-
83	0.99	-	-	-	-
84	0.99	-	-	-	-
85	0.99	-	-	-	-
86	0.99	-	-	-	-
87	0.99	-	-	-	-
88	0.99	-	-	-	-
89	0.99	-	-	-	-
90	0.99	-	-	-	-
91	0.99	-	-	-	-
92	0.99	-	-	-	-
93	0.97	45 63	0.45 0.45	63 45	0.46 0.46
94	0.99	-	-	-	-
95	0.99	-	-	-	-
96	0.99	-	-	-	-
97	0.99	-	-	-	-
98	0.97	19 63	0.48 0.47	17 45	0.49 0.48
99	0.99	-	-	-	-
100	0.99	-	-	-	-

proposed model prefers to have a “Blank” label in the output instead of a false recognized label. Results on two datasets, as shown in Table 3-12, confirmed that in a case of false recognition of the model, the recognized Softmax outputs for all classes are lower than the predefined threshold. As shown in Table 3-12, the proposed model obtains an average of recognized Softmax outputs equal to 0.98 and 0.59 on the RKS-PERSIANSIGN and ASLVID datasets, respectively. We have a higher recognition accuracy on the RKS-PERSIANSIGN dataset than the ASLVID one due to a higher video sample instances in each class. More concretely, we have 100 and 7 video samples for each class in the RKS-PERSIANSIGN and ASLVID datasets, respectively. This comes from the fact that deep learning-based models generally have a better performance if they train with a large amount of data. Furthermore, there are some challenges in the similar signs, such as ’Congratulation’, ’Excuse’, ’Upset’, ’Blame’, ’Fight’, ’Competition’. For example, ’Excuse’ and ’Congratulation’, ’Upset’ and

'Blame', 'Fight' and 'Competition' signs include many similar frames. Thus, adding additional samples to these categories could help to learn more powerful features to better represent sign categories and reduce miss-classifications due to low inter-class variabilities. As future work, we aim to collect a dataset, including more realistic continuous sign videos and the corresponding isolated sign videos. Relying on this dataset, we can check the performance of the proposed model to use in a realistic scenario.

References

[1] Bhagat, N. K., Vishnusai, Y. and Rathna, G. N. (2019) 'Indian Sign Language Gesture Recognition using Image Processing and Deep Learning', 2019 Digital Image Computing: Techniques and Applications, DICTA 2019. IEEE, pp. 1–8. doi: 10.1109/DICTA47822.2019.8945850.

[2] Bird, J. J., Ekárt, A. and Faria, D. R. (2020) 'British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language', Sensors (Switzerland), 20(18), pp. 1–19. doi: 10.3390/s20185151.

[3] Bu, Q. et al. (2020) 'Deep transfer learning for gesture recognition with WiFi signals', Personal and Ubiquitous Computing. Personal and Ubiquitous Computing. doi: 10.1007/s00779-019-01360-8.

[4] Camgoz, N. C. et al. (2017) 'SubUNets: End-to-End Hand Shape and Continuous Sign Language Recognition', Proceedings of the IEEE International Conference on Computer Vision, 2017-October, pp. 3075–3084. doi: 10.1109/ICCV.2017.332.

[5] Camgoz, N. C. et al. (2018) 'Neural Sign Language Translation', Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 7784–7793. doi: 10.1109/CVPR.2018.00812.

[6] Cheng, K. L. et al. (2020) 'Fully Convolutional Networks for Continuous Sign Language Recognition', Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12369 LNCS, pp. 697–714. doi: 10.1007/978-3-030-58586-0-41.

[7] Cui, R., Liu, H. and Zhang, C. (2019) 'A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training', IEEE Transactions on Multimedia. IEEE, 21(7), pp. 1880–1891. doi: 10.1109/TMM.2018.2889563.

[8] Escobedo Cardenas, E. J. and Chavez, G. C. (2020) 'Multimodal hand gesture recognition combining temporal and pose information based on CNN descriptors and histogram of cumulative magnitudes', Journal of Visual Communication and Image Representation. Elsevier Inc., 71, p. 102772. doi: 10.1016/j.jvcir.2020.102772.

[9] Gomez-Donoso, F., Orts-Escolano, S. and Cazorla, M. (2019) 'Accurate and efficient 3D hand pose regression for robot hand teleoperation using a monocular RGB camera', Expert Systems with Applications. Elsevier Ltd, 136, pp. 327–337. doi: 10.1016/j.eswa.2019.06.055.

[10] Halvardsson, G. et al. (2021) 'Interpretation of Swedish Sign Language Using Convolutional Neural Networks and Transfer Learning', SN Computer Science, 2(3), pp. 0–3. doi: 10.1007/s42979-021-00612-w.

[11] Jiang, X. et al. (2020) 'Fingerspelling Identification for Chinese Sign Language via AlexNet-Based Transfer Learning and Adam Optimizer', Scientific Programming, 2020. doi: 10.1155/2020/3291426.

[12] Nada B. Ibrahim, H. H. Z. and M. M. S. (2020) 'Advances, Challenges and Opportunities in Continuous Sign Language Recognition', Journal of Engineering and Applied Sciences, 15(5), pp. 1205–1227.

[13] Neidle, C., Thangali, A. and Sclaroff, S. (2012) 'Challenges in Development of the American Sign Language Lexicon Video Dataset (ASLLVD) Corpus', Proc. of 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC 2012, pp. 1–8.

[14] Papastratis, I. et al. (2020) 'Continuous Sign Language Recognition through Cross-Modal Alignment of Video and Text Embeddings in a Joint-Latent Space', IEEE Access, 8, pp. 91170–91180. doi: 10.1109/ACCESS.2020.2993650.

[15] Papastratis, I., Dimitropoulos, K. and Daras, P. (2021) 'Continuous sign language recognition through a context-aware generative adversarial network', Sensors, 21(7). doi: 10.3390/s21072437.

[16] Rastgoo, Razieh, Kiani, Kourosh, Escalera, Sergio, Sabokrou, M. (2021) 'Multi-Modal Zero-Shot Sign Language Recognition', arXiv:2109.00796.

[17] Rastgoo, Razieh, Kiani, Kourosh, Escalera, S. (2021) 'ZS-SLR: Zero-Shot Sign Language Recognition from RGB-D Videos', arXiv:2108.10059.

[18] Rastgoo, R. et al. (2021) 'Sign language production: A review', IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 3446–3456. doi: 10.1109/CVPRW53098.2021.00384.

[19] Rastgoo, R., Kiani, K. and Escalera, Sergio, Athitsos, Vassilis, Sabokrou, M. (2022) 'All You Need In Sign Language Production', arXiv:2201.01609v2.

[20] Rastgoo, R., Kiani, K. and Escalera, S. (2018) 'Multi-modal deep hand sign language recognition in still images using Restricted Boltzmann Machine', Entropy, 20(11), pp. 1–15. doi: 10.3390/e20110809.

[21] Rastgoo, R., Kiani, K. and Escalera, S. (2020a) 'Hand sign language recognition using multi-view hand skeleton', Expert Systems with Applications, 150. doi: 10.1016/j.eswa.2020.113336.

[22] Rastgoo, R., Kiani, K. and Escalera, S. (2020b) 'Video-based isolated hand sign language recognition using a deep cascaded model', Multimedia Tools and Applications. doi: 10.1007/s11042-020-09048-5.

[23] Rastgoo, R., Kiani, K. and Escalera, S. (2021a) 'Hand pose aware multimodal isolated sign language recognition', Multimedia Tools and Applications. Multimedia Tools and Applications, 80(1), pp. 127–163. doi: 10.1007/s11042-020-09700-0.

[24] Rastgoo, R., Kiani, K. and Escalera, S. (2021b) 'Sign Language Recognition: A Deep Survey', Expert Systems with Applications. Elsevier Ltd, 164(July 2020), p. 113794. doi: 10.1016/j.eswa.2020.113794.

[25] Rastgoo, R., Kiani, K. and Escalera, S. (2022) 'Real-time isolated hand sign language recognition using deep networks and SVD', Journal of Ambient Intelligence and Humanized Computing. Springer Berlin Heidelberg, 13(1), pp. 591–611. doi: 10.1007/s12652-021-02920-8.

[26] Sharma, S., Gupta, R. and Kumar, A. (2021) 'Continuous sign language recognition using isolated signs data and deep transfer learning', Journal of Ambient Intelligence and Humanized Computing. Springer Berlin Heidelberg, (2020). doi: 10.1007/s12652-021-03418-z.

[27] Wadhawan, A. and Kumar, P. (2020) 'Deep learning-based sign language recognition system for static signs', Neural Computing and Applications, pp. 1–12. Available at: https://doi.org/10.1007/s00521-019-04691-y.

[28] Zhang, S. et al. (2020) 'Learning Noise Invariant Features Through Transfer Learning for Robust End-to-End Speech Recognition', ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. IEEE, 2020-May, pp. 7024–7028. doi: 10.1109/ICASSP40776.2020.9053169.