applied sciences

MDPI

Article

A Dynamic Gesture Recognition Interface for Smart Home Control based on Croatian Sign Language

Luka Kraljević *,†, Mladen Russo †, Matija Pauković and Matko Šarić

Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture,
Laboratory for Smart Environment Technologies, University of Split, 21000 Split, Croatia;
mrusso@fesb.hr (M.R.); matija.paukovic.00@fesb.hr (M.P.); msaric@fesb.hr (M.Š.)
* Correspondence: lkraljev@fesb.hr
† These authors contributed equally to this work.

Received: 3 February 2020; Accepted: 23 March 2020; Published: 27 March 2020

Abstract: Deaf and hard-of-hearing people are facing many challenges in everyday life. Their communication is based on the use of a sign language, and the ability of the cultural/social environment to fully understand such a language defines whether or not it will be accessible for them. Technology is a key factor that has the potential to provide solutions to achieve a higher accessibility and therefore improve the quality of life of deaf and hard-of-hearing people. In this paper, we introduce a smart home automatization system specifically designed to provide real-time sign language recognition. The contribution of this paper implies several elements. Novel hierarchical architecture is presented, including resource-and-time-aware modules—a wake-up module and high-performance sign recognition module based on the Conv3D network. To achieve high-performance classification, multi-modal fusion of RGB and depth modality was used with the temporal alignment. Then, a small Croatian sign language database containing 25 different language signs for the use in smart home environment was created in collaboration with the deaf community. The system was deployed on a Nvidia Jetson TX2 embedded system with StereoLabs ZED M stereo camera for online testing. Obtained results demonstrate that the proposed practical solution is a viable approach for real-time smart home control.

Keywords: sign language; multimodal gesture recognition; home automatization; 3D convolution

1. Introduction

Most people at some point in life, especially in older age, probably experience either temporary or permanent disability, or are facing increasing difficulties in functioning [1,2]. Considering the type of impairment we focus on within this paper—in 2019, around 466 million people in the world had a disability based on deafness, of which 34 million were children. By 2050, indicators predict that 900 million people will face the consequences of the inability of equal communication daily [3]. Regardless of the concerning quantitative indicators, the unquestionable truth is that such persons must necessarily use specific communication procedures to integrate. The communication of deaf and speech-impaired people is based on the use of sign language, and the knowledge of it allows the integration but only to a certain extent. Disability, as an inability to integrate, is a condition and a direct result of the inaccessible and complex environment surrounding those with a health impairment [1]. The environment either disables people or supports their inclusion and participation in the general population [1].

Technology plays a crucial part in accessibility initiatives and solutions [1,2], where particular emphasis is placed on research and development focused on new methods in human–computer interaction (HCI) oriented toward natural user interfaces (NUI). Having the demand for familiar, practical, intuitive, and natural interfaces, the research community and the industry started to focus on the speech and vision-based interaction and interfaces [4,5]. Incorporating the circumstances of the deaf and speech-impaired people who use sign language to communicate, the vision-based systems are a reasonable direction that has a high potential to provide applicable and effective solutions. Sign languages are complex languages, with a specific structure, grammar, and lexis, and with those attributes, it falls into the category of natural languages. This primarily implies the non-existence of universality, and the exclusion of absolute mutual intelligibility between different sign languages depending on the region, family, speaking area, dialect, and so on [6,7]. Because of the nature of sign language, the research within sign language recognition implies the research in gesture recognition [8] as the most respective subproblem. In general, there are two main approaches to gesture recognition: vision-based and non-vision based [9]. For a long time, the performance of the non-vision based approach, i.e., use of different sensory gloves, has dominated in comparison with vision-based approaches, and this difference has been most pronounced in experimental environments that capture real-world conditions such as poor lighting and background complexity. Although the non-vision approach reports good results in terms of both accuracy and speed, this approach requires specific hardware with supporting equipment, which is, for most HCI cases, impractical. As is the case with the most computer vision tasks, such as object detection and facial recognition, the employment of deep learning methods in the scientific field of gesture recognition has delivered excellent results, surpassing state-of-the-art methods [8]. The improvement made a vision-based approach preferable as it offers a remote, contactless data acquisition. Recently, HCI systems based on convolutional neural networks have shown excellent performance. Owing to its ability to learn spatial features through the use of multiple layers automatically, a CNN was employed as the main building block in almost every recent solution of computer vision problems.

The field of dynamic gesture recognition primarily emphasizes research on large databases [10,11] where different methods of extending training sets based on spatio–temporal transformations are often used to achieve better performance [12]. Further, to validate and compare the results or to adapt the system to work with another set of gestures, it is common to use a transfer learning method, where the initial parameters of deep architectures obtained by training on a large dataset are modified by additional re-training with the base of interest [4]. The most crucial challenge in deep-based gesture recognition is how to deal with the temporal dimension. As shown in the related work, many different approaches are employed for modeling the time sequence of dynamic gestures; nevertheless, our motivation behind using a three-dimensional convolutional neural network (3DCNN) was primarily because it showed excellent results extracting relevant spatio–temporal features commonly used in the related fields of activity recognition and action localization from video. When presented with enough data to cover various situations such as differing complexities of the background and lighting and a variety of signers, a system based on a 3DCNN can demonstrate good performance in generalizing unseen data. Additionally, the use of a diverse number of input modalities such as RGB, depth, and infrared (IR) images reported different results depending on the experimental environment, where it was shown that a combination of multiple modalities could improve performance in terms of accuracy and better robustness to different factors like occlusion. Compared to the relevant scientific research, besides using modality fusion to improve performance, the primary motivation of this work is to investigate appropriate deep learning architecture suitable for real-time recognition. Many studies investigate only recognition performance, and some of the approaches are impossible to run in real-time as they consist of multiple deep architectures with several input modalities, thus forcing memory and computational limits [13]. The proposed system is designed to reduce unnecessary computational overflow by introducing real-time gesture spotting. Further, the overall effectiveness of the system indicates the possibility of an expansion towards the realization of continuous sign language, which at its core requires us to include other non-verbal communication factors, such as pose and motion estimation, and facial recognition, whose realization must be performed in parallel with gesture detection.

In this paper, we propose a system for real-time sign language dynamic gesture recognition with application in the context of a smart home environment. The contributions of our work are founded on several different elements. First, achieving real-time dynamic gesture recognition with online recognition being deployed and realized on NVIDIA Jetson TX2 embedded system combined with StereoLabs ZED M stereo camera. Second, the use of a hierarchical architecture approach to achieve more effective use of resources (memory and processing demand) using the wake-up module as an activation network and the fusion of two 3DCNN networks as a high-performance classifier with the multimodal inputs—RGB with depth inputs. Third, the formation of a specific custom sign language gesture control commands to interact with the smart home environment. Lastly, training and evaluation of the proposed system is based on Croatian Sign Language gestures/commands, forming dataset with a specific application in the smart home environment.

The rest of the paper is organized as follows. Section 2 explains the relevant research. In Section 3, the proposed sign language interface, and the corresponding component modules are described. The performance evaluation is made in Section 4, and Section 5 concludes the paper.

2. Related Work

Hand gesture recognition is one of the most prominent fields of human–computer interaction. There are many related studies for hand gesture recognition using wearable [9] and non-wearable sensors [8]. Considering the different sensors, the research involves the use of specialized hardware such as Microsoft Kinect [14–16], stereo camera [17], sensor gloves [18–20], and non-specialized hardware like mobile phone cameras [21,22], web cams [23], etc. The use of specialized hardware for hand gesture acquisition primarily bridges certain steps in the process that would otherwise have to be taken into account, such as hand segmentation, hand detection, and hand orientation, finger isolation, etc. Traditional approaches in the problem of gesture classification were based on hidden Markov models (HMMs) [24], support vector machines (SVMs) [25], conditional random fields (CRFs) [26], and multi-layer perceptron (MLP) [27]. In recent years, research interests have been shifted from a sensor-based approach to a vision-based approach, thanks to rapid advancement in the field of deep learning-based computer vision. The most crucial challenge in deep learning-based gesture recognition is how to deal with the temporal dimension. More recent work implies the use of models based on approaches utilizing deep convolutional neural network (CNN), long-short term memory (LSTM), and derivative architectures. Earlier, 2DCNNs were shown to provide high accuracy results on images, so those were applied to videos combined with different approaches. Video frames were used as multiple inputs for a 2DCNN in [28,29]. A combination of a 2DCNN and an LSTM was proposed in [30] where features were extracted with a 2DCNN network and then applied to an LSTM network to cover the temporal component. In [30], spatial features were firstly extracted from frames with long-term recurrent convolutional network (LRCN) and then temporal features were extracted with a recurrent neural network (RNN). A two-stream convolutional network (TSCN) was used in [29] to extract spatial and temporal features. In [31], a convolutional LSTM—VideoLSTM was used to learn spatio–temporal features from previously extracted spatial features. In [32] the proposed model is a combination of a three-dimensional convolutional neural network (3DCNN) and long short-term memory (LSTM) and used to extract the spatio–temporal features from the dataset containing RGB and depth images. In [33], spatiotemporal features were extracted in parallel utilizing a 3D convolutional neural network (3DCNN). In [34], 3DCNNs were used for spatio–temporal feature extraction with 3D convolution and pooling. The extraction and the quality of spatial features in the recognition process is highly influenced by the factors such as background complexity, hand position, hand-to-scene size, hand/fingers overlapping, etc, [8,35]. In such circumstances, spatial features can easily be overwhelmed by those factors and become undiscriminating in the process. Therefore, temporal information provided by the sequence of scenes/frames becomes the key factor [36]. This information, especially for real-time gesture recognition process, is of high importance considering the stream of video frames and learning the spatio–temporal features simultaneously is more likely to provide quality results rather than when either separate or in sequence [4,36].

Considering the modality, in [37] the RGBD data of a hand with the upper-body was combined and used for sign language recognition. To detect the hand gestures, in [38], YCbCr and SkinMask segmented images were the CNN's two-channel inputs. In [39], a method for fingertip detection and real-time hand gesture recognition based on RGBD modality and the use of 3DCNN network was proposed. In [40], the best performance was reported using RGBD data and a histogram of gradient (HOG) with an SVM as a classifier. Further, in [41], a dynamic time wrapping (DTW) method was used on the HOG and histogram of the optical flow (HOF) to recognize gestures.

Advancement in the field of automatic sign recognition is profoundly dependent on the availability of the relevant sign language databases which are specified for the language area [42] or have limited vocabulary for the area of application [43]. Given the practical applications of automatic sign language recognition, most studies are focused on the methods oriented toward the translation of sign language gestures into textual information. One of the interesting solutions for upgrading the social interaction of sign language users is proposed in [38], where authors introduced a sign language translation model using 20 common sign words. In [42], a deep learning translation system is proposed for 105 sentences that can be used in emergency situations. Considering that hand gestures are the most natural and thus the most commonly used modality, among others in HCI communication, it is vital to consider sign language as the primary medium for NUI. An example of a practical system for the automatic recognition of the American sign language finger-spelling alphabet to assist people living with speech or hearing impairments is presented in [44]. The system was based on the use of Leap Motion and Intel RealSense hardware with the SVM classification. In [45], a wearable wrist-worn camera (WwwCam) was proposed for real-time hand gesture recognition to enable services such as controlling mopping robots, mobile manipulators, or appliances in a smart home scenario.

3. Proposed Method—Sign Language Command Interface

In this section, we describe a proposed system for human–computer interaction based on Croatian Sign Language. The design of the proposed method was envisaged as a dynamic gesture-based control module for interaction with the smart environment. Our goal was to implement a touchless control interface customized for sign language users. By selecting a limited vocabulary tailored in the form of smart home automatization commands, the proposed solution was used to manage household appliances such as lights, thermostats, door locks, and domestic robots. The infrastructure of the proposed sign language control module was designed to meet certain requirements for real-time online applications such as reasonable classification performance and hardware efficiency with swift reaction time. The proposed infrastructure consists of three main parts: wake-up module (see Section 3.1), sign recognition module (see Section 3.2), and sign command interface (see Section 3.3). Figure 1 illustrates the pipeline of the proposed sign language command interface. In the defined workflow, the proposed system continuously receives RGB-D data, placing it into two separate input queues. The wake-up module is subscribed to the data queue containing a sequence of depth images. In each operating cycle, a wake-up module performs gesture detection based on N consecutive frames. If the start of a gesture was successfully detected, the command interface is placed in an attention state, and the sign language recognition module becomes active. It performs hierarchical gesture classification on two modality sequences of maximum size M starting from the beginning of the input queues. The result of sign classification was passed as a one-hot encoded vector to command parser for mapping of recognized gesture sign to a predefined vocabulary word, further used for building the automatization command in JSON format.

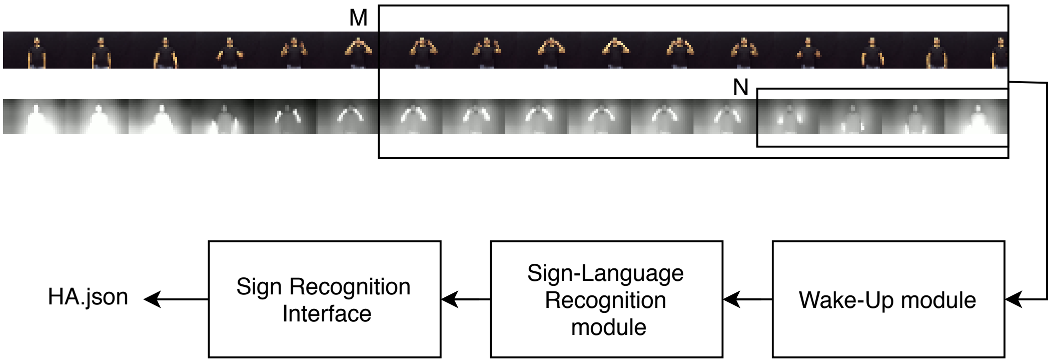


Figure 1. Proposed sign language command interface.

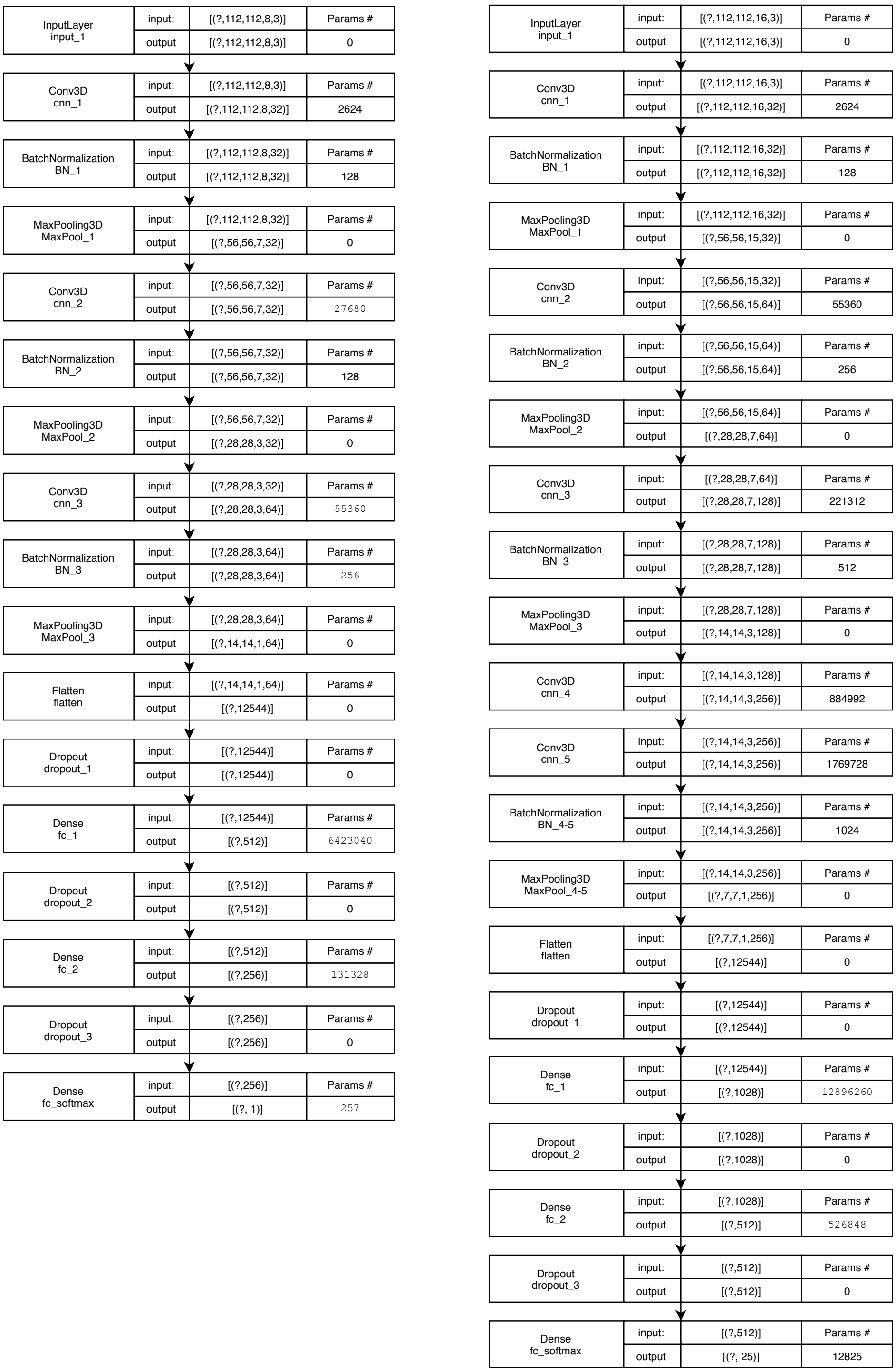
3.1. Wake-Up Gesture Detector

The primary purpose of the wake-up module was to lower power consumption and to deliver better hardware performance and memory resource efficiency in the implementation of the proposed sign language command interface. Given the case where the control interface continuously receives an incoming video stream of RGB-D data, where for the most time no gesture is present, the main task of the proposed wake-up module was to act as an initiator for sign language recognition module. This component performs a binary classification (gesture, no gesture) on the first N frames of the Depth data queue; thus, it reduces the number of undesired activations of the proposed command module. To adequately address the real-life scenario where different signs of Croatian sign language have different duration; in this work, the proposed detector was employed for managing the queue length M of the sign recognition module. In case when sign recognition was active, the detector module keeps operating as a sliding window with corresponding stride of 1. When the number of following no gesture prediction reaches the threshold, the wake-up module performs a temporal deformation by resampling input sequences of size M to fixed size m on witch sign language classification operates. In this work, resampling was done by selecting m input frames linearly distributed between 1 and M. Since the wake-up module runs continuously, it was implemented as a lightweight 3DCNN architecture as shown in Figure 2, left.

The network consists of four convolutional blocks, followed by two fully-connected layers. Each convolution layer was interspersed with a batch normalization layer, a rectified linear unit (ReLu), and max-pooling layers. The three convolutional layers had 32, 32, and 64 filters, respectively. The kernel size of each Conv3D layer was 3 × 3 × 3, together with a stride of 1 in each direction. A drop-out was utilized during the training after the last convolutional layer, in addition to L2-regularization on all weight layers. To decrease the probability of false-positive predictions, the model was trained on BinaryCrossentropy loss using Adam optimization with a mini-batch size of 16. The proposed gesture wake-up architecture presents a fast and robust method suitable for real-time applications.

3.2. Sign Recognition Module

This module represents a core functionality of the proposed smart home system. It has the task of recognizing users' gestures and decoding them to sign language. Given the practical application of our system, it was necessary to consider the deep learning architecture suitable for available memory capacity and power budget. Also, it needs to provide a good trade-off between classification performance, fast reaction time, and good robustness to environmental conditions such as varying lighting conditions and complex background. Motivated by the recent success of adopting multi-modal data for robust dynamic gesture recognition, in this work, we propose using a multi-input network classifier. The proposed solution has consisted of two identical parallel sub-networks, where each operates on different data modality. Both stream of modalities, respectively RGB and Depth data, were spatially and temporally aligned, which facilitates our subnetworks to have the same understanding of an input sign gesture. By opting for two different modalities, our architecture ensures that each subnetwork learns relevant spatiotemporal features of a given input modality. In our approach, Conv3D was employed to extract spatiotemporal information from input video data. The network architecture of the proposed sign language classifiers is illustrated in Figure 3.



InputLayer input_1	input:	[(?,112,112,8,3)]	Params #
	output	[(?,112,112,8,3)]	0
Conv3D cnn_1	input:	[(?,112,112,8,3)]	Params #
	output	[(?,112,112,8,32)]	2624
BatchNormalization BN_1	input:	[(?,112,112,8,32)]	Params #
	output	[(?,112,112,8,32)]	128
MaxPooling3D MaxPool_1	input:	[(?,112,112,8,32)]	Params #
	output	[(?,56,56,7,32)]	0
Conv3D cnn_2	input:	[(?,56,56,7,32)]	Params #
	output	[(?,56,56,7,32)]	27680
BatchNormalization BN_2	input:	[(?,56,56,7,32)]	Params #
	output	[(?,56,56,7,32)]	128
MaxPooling3D MaxPool_2	input:	[(?,56,56,7,32)]	Params #
	output	[(?,28,28,3,32)]	0
Conv3D cnn_3	input:	[(?,28,28,3,32)]	Params #
	output	[(?,28,28,3,64)]	55360
BatchNormalization BN_3	input:	[(?,28,28,3,64)]	Params #
	output	[(?,28,28,3,64)]	256
MaxPooling3D MaxPool_3	input:	[(?,28,28,3,64)]	Params #
	output	[(?,14,14,1,64)]	0
Flatten flatten	input:	[(?,14,14,1,64)]	Params #
	output	[(?,12544)]	0
Dropout dropout_1	input:	[(?,12544)]	Params #
	output	[(?,12544)]	0
Dense fc_1	input:	[(?,12544)]	Params #
	output	[(?,512)]	6423040
Dropout dropout_2	input:	[(?,512)]	Params #
	output	[(?,512)]	0
Dense fc_2	input:	[(?,512)]	Params #
	output	[(?,256)]	131328
Dropout dropout_3	input:	[(?,256)]	Params #
	output	[(?,256)]	0
Dense fc_softmax	input:	[(?,256)]	Params #
	output	[(?, 1)]	257

InputLayer input_1	input:	[(?,112,112,16,3)]	Params #
	output	[(?,112,112,16,3)]	0
Conv3D cnn_1	input:	[(?,112,112,16,3)]	Params #
	output	[(?,112,112,16,32)]	2624
BatchNormalization BN_1	input:	[(?,112,112,16,32)]	Params #
	output	[(?,112,112,16,32)]	128
MaxPooling3D MaxPool_1	input:	[(?,112,112,16,32)]	Params #
	output	[(?,56,56,15,32)]	0
Conv3D cnn_2	input:	[(?,56,56,15,32)]	Params #
	output	[(?,56,56,15,64)]	55360
BatchNormalization BN_2	input:	[(?,56,56,15,64)]	Params #
	output	[(?,56,56,15,64)]	256
MaxPooling3D MaxPool_2	input:	[(?,56,56,15,64)]	Params #
	output	[(?,28,28,7,64)]	0
Conv3D cnn_3	input:	[(?,28,28,7,64)]	Params #
	output	[(?,28,28,7,128)]	221312
BatchNormalization BN_3	input:	[(?,28,28,7,128)]	Params #
	output	[(?,28,28,7,128)]	512
MaxPooling3D MaxPool_3	input:	[(?,28,28,7,128)]	Params #
	output	[(?,14,14,3,128)]	0
Conv3D cnn_4	input:	[(?,14,14,3,128)]	Params #
	output	[(?,14,14,3,256)]	884992
Conv3D cnn_5	input:	[(?,14,14,3,256)]	Params #
	output	[(?,14,14,3,256)]	1769728
BatchNormalization BN_4-5	input:	[(?,14,14,3,256)]	Params #
	output	[(?,14,14,3,256)]	1024
MaxPooling3D MaxPool_4-5	input:	[(?,14,14,3,256)]	Params #
	output	[(?,7,7,1,256)]	0
Flatten flatten	input:	[(?,7,7,1,256)]	Params #
	output	[(?,12544)]	0
Dropout dropout_1	input:	[(?,12544)]	Params #
	output	[(?,12544)]	0
Dense fc_1	input:	[(?,12544)]	Params #
	output	[(?,1028)]	12896260
Dropout dropout_2	input:	[(?,1028)]	Params #
	output	[(?,1028)]	0
Dense fc_2	input:	[(?,1028)]	Params #
	output	[(?,512)]	526848
Dropout dropout_3	input:	[(?,512)]	Params #
	output	[(?,512)]	0
Dense fc_softmax	input:	[(?,512)]	Params #
	output	[(?, 25)]	12825

Figure 2. Proposed network architectures of the wake-up module (**left**) and unimodal sign recognition module (**right**).

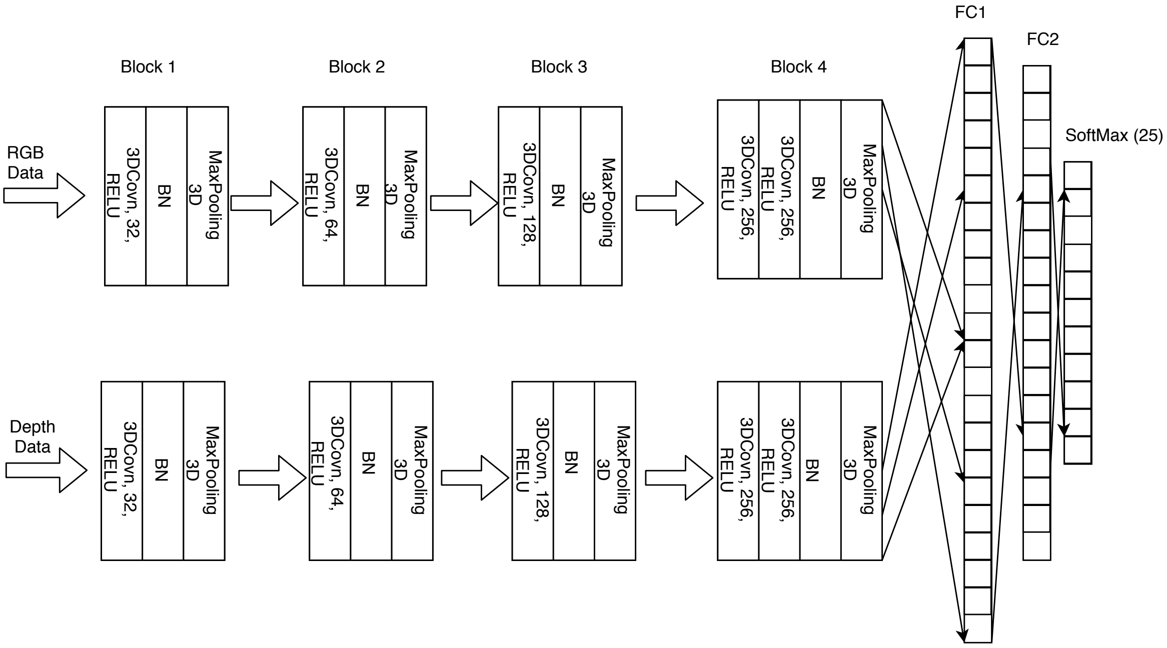


Figure 3. Sign language classificator—network architecture.

In our approach, each network was trained separately with the appropriate data type, each maintaining its auxiliary loss on the last fully connected layer. For modality fusion, both subnetworks where concatenated with a new dense layer. Training was performed by optimizing weights of dense layers while freezing the weights of the ConvNet block. Each unimodal architecture consists of blocks numbered 1 to 4, followed by two fully connected layers and the output Softmax layer. In each block, a Conv3D was succeeded by Batch normalization and MaxPool3D layer except for block 4, which contains two Conv3D layers. The number of features maps per Conv3D are 32, 64, 128, 256, 256, respectively, for each layer. All Conv3D layers were defined by a filter size of 3 × 3 × 3, while padding and stride dimensions are set to 1 × 1 × 1. Further, in the first block, only spatial down-sampling was performed, while for the rest of the blocks, pool size and stride of the MaxPool3D layer was set to 2 × 2 × 2. A drop-out of 0.2 was employed during the training before each fully connected layer. The training was performed using Adam optimizer with a mini-batch size of 16. A detailed structure of the unimodal RGB model is shown in Figure 2, right.

To properly train 3DCNN, much more training data were required than with 2D counterparts since a number of learnable parameters were much higher. To prevent overfitting during training in this work, we performed an online spatiotemporal data augmentation. Spatial augmentation included translation, rotation, and scaling, where transformation parameters were fixed for a particular data batch. Besides affine transformations, we also applied Gaussian blur on RGB data. Since we trained our network from scratch, we adopted lower transformation values for these steps, which helps the model to converge faster. For temporal augmentation, we randomly selected successive frames in a range of defined input size. In the case of fusion training, parameters of data augmentation were fixed to maintain spatial and temporal alignment between two input modalities. All of these operations were applied randomly inside mini-batches that were fed into the model.

3.3. Sign Command Interface

After the wake-up module concludes that there are no more gestures presented in the input data stream i.e., the number of consecutive "no gesture" predictions reaches the defined threshold, it signals the sign language recognition module to perform sign classification on the current RGB-D queues. The result of sign classification is given through the Softmax layer as class-membership probabilities. These probabilities are then passed to the Sign command interface in form one-hot encoded vector.

The main task of the Sign command interface is to create structured input from the sequence of recognized language signs. This sequence can be later used by smart home automatization to realize user commands. In this work, we established a sign language vocabulary suitable for a specific range of commands within the home automatization context. We also defined a grammar file with all possible command patterns. For a given smart home automatization scenario, we distinguished between two types of devices: an actuator where the user can set the state, and the sensor, which allows for the acquisition of information to get the state of the sensor/environment parameter.

Further, we designed commands patterns/to control or read the state of the individual device or a group of devices. This command pattern was formatted as follows: action part, device name/device group part, and an optional part of a command, which was the location descriptor (e.g., room, kitchen). An example of user command is shown in Figure 4.

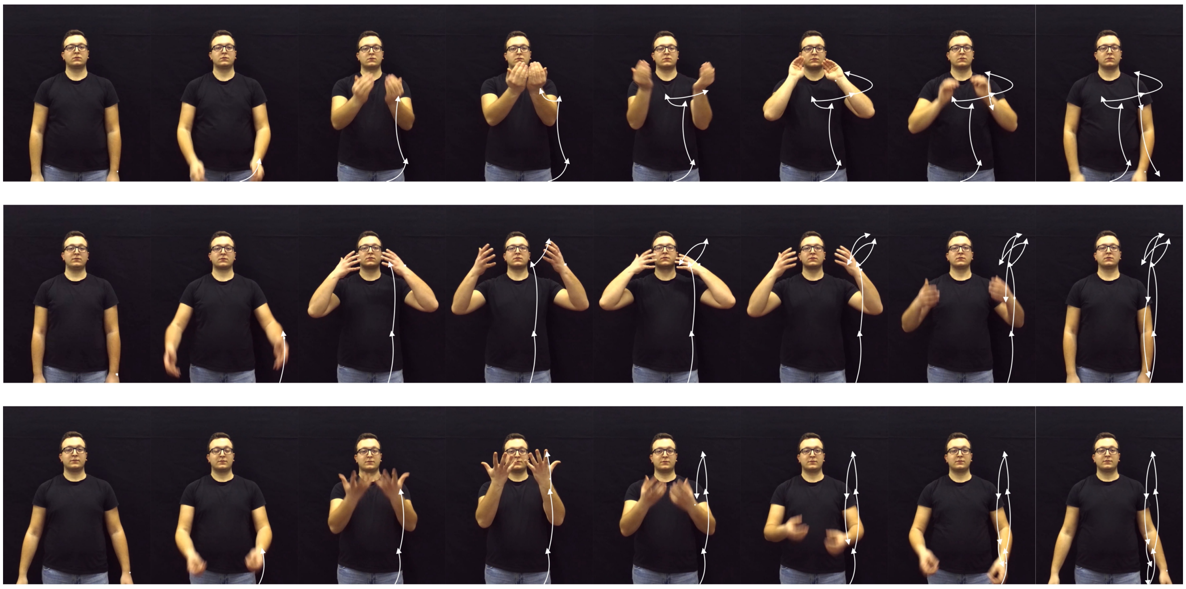


Figure 4. An example of sign language command sequence—"Turn ON Air conditioner in the bathroom". White trajectory illustrates the movement of hand during the gesture.

Another functional contribution of the command interface was to improve recognition of users sign language command. Relying on the established command pattern/format, which is seen as the sequence structure, the proposed sign interface additionally performed classification refinement. If the prediction at the current sequence step does not fit in expected language sign subclass, it selects the next prediction with the highest probability. This refinement can be recursively repeated for next two steps. For example, if the system recognizes a sign related to the household device while the system expects a command action, it can reject prediction and look for the next sign command with the biggest probability score, which could also be the one expected.

3.4. Croatian Sign Language Dataset

Sign language is a visual language of communication used by deaf and hard-of-hearing people. According to [46], almost all EU nations have some form of recognition of its sign language. The task of automatic sign language recognition primarily includes the same constraints relevant to dynamic gesture classification. In the context of the task, it is common to use a collection of labeled video clips in which users are performing predefined hand gestures. Although there is a considerable number of available dynamic hand gesture datasets, such as 20BN-jester [11], EgoHands [47], Nvidia Dynamic Hand Gesture [48], a relatively small number of the available sign language corpora exist [49]. Since sign language is unique for a particular region and, since there are no available Croatian datasets, in this work, we created a new small dataset based on Croatian sign language—SHSL. Considering that the sign languages come with a broad range of factors, we restricted ourselves to the topic of smart home automatization command. First, through collaboration with the deaf community, the vocabulary for home automatization was defined. The proposed sign language corpus contains 25 language signs needed to construct a smart home command simultaneously. The previously mentioned command pattern groups SHSL signs in three categories: actions, household items, and house location, where each sign category contains 13, 8, and 4 signs, respectively. For the production of the SHSL dataset, 40 volunteers were selected to perform each of the 25 sign gestures twice, which resulted in a total collection of 2000 sign videos. The average duration of the gesture length is 3.5 s. In our work, data acquisition was performed with a ZED M camera, which facilitates the collection for RGB-D data. The video was recorded in 1920 × 1080 resolution with 30 fps. The camera stand was placed 1.3 m from the signer, and each signer was instructed to wear a dark shirt.

4. Experiments

In this section, an experimental evaluation of the proposed system was made by analyzing each component of the proposed system separately. Given that the network architecture of the lightweight wake-up module and high-performance sign recognition module was based on Conv3D, which generally has a considerable amount of learnable parameter, as illustrated in Figure 2, a substantial amount of data were required to minimize the potential of overfitting. In this work, apart from applying data augmentation methods, we also pre-trained our models with the Nvidia hand gesture dataset to obtain proper model initialization. The Nvidia hand gesture dataset contains 1532 clips distributed between 25 dynamic hand gesture classes. In total, 20 subjects were recorded with multiple sensors, installed on several locations inside the car simulator. To initialize models, we only used the provided RGB and depth data modality randomly split with 5:1 ratio, into the training and validation set. Each component of the system was trained on two Nvidia's RTX 2080 Ti's using the TensorFlow mirrored distribution strategy. In the testing phase, modules were integrated and exported to a power-efficient embedded AI computing device, Nvidia Jetson T2X, to calculate forward prediction. Jetson TX2 is a CUDA compatible device, which was required for our ZED M camera to compute the depth modality in real-time.

4.1. Performance Evaluation of the Wake-Up Module

In this work, the wake-up module had the role of a real-time hand gesture spotter, which in addition to distinguishing between "gesture" and "non-gesture", it also manages the length of input sequence M used by sign recognition module. Considering that the efficiency of the Sign language command interfaces highly depends on the performance of the Wake-up module, we needed to analyze the accuracy regarding the size of the input sequence N on which the gesture detector operates. Using longer input sequence N directly affects the time it takes to recognize a gesture and thus to execute the users' command. Performance evaluation of the proposed wake-up module was given as a binary classification accuracy concerning three input sizes 4, 8, and 16. For the sake of real-time execution, a lightweight architecture was trained using only depth modality data, and the corresponding results are shown in Table 1.

Table 1. Wake-up binary classification accuracy.

	4-Frames	**8-Frames**	**16 Frames**
Nvidia Hand Gesture	91.3	94.5	94.7
SHSL	89.2	95.1	95.7

From the results, it is visible that the best accuracy results were obtained using 16 frames. Concerning practical real-time requirements explained in Section 3, the proposed Sign language command interface integrates the 8-frames Wake-up module. This decision was made based on parameter reduction with a negligible decrease in performance, and a smaller frame size also provides better wake-up resolution.

Figure 5 shows the accuracy of the classification of the proposed model on the train and validation sets using 8-frames, from which it is visible that the model has a good fit on the problem. Training was performed with the class ratio of 5:3 for "no-gesture" and "gesture" as our experiments showed that this proportion was sufficient to obtain model relevance of 93.8% and 93.1% in terms of precision and recall.

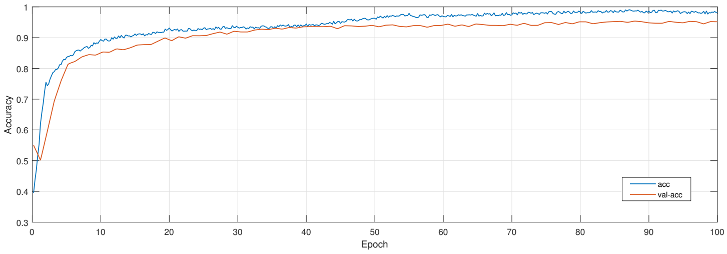


Figure 5. The accuracy of Wake-up model during training.

4.2. Performance Evaluation of the Sign Recognition Module

In this work, a sign recognition module is realized as a hierarchical architecture of two identical parallel subnetworks, each operating on different input modality. For the training process, we firstly initialize each of two subnetworks with Nvidia hand gesture dataset using respective RGB and Depth data. For model initialization, a total of five training-validation runs are performed. The pre-trained subnetworks are selected based on max reported validation accuracy of 62.3% and 65.6%, respectively, for RGB and depth modality. After obtaining the validation accuracy, the model was updated with the rest of the 306 validation samples to finish the initialization process. The following fine-tuning process to the SHSL dataset started with a learning rate of 0.005, and it was controlled by scheduler that adjusted in order to reduce the learning rate by factor 5 if the cost function did not improve by 10%. For this experiment, we compared the performance of each single-modality subnetwork, and modality-fusion model concerning the number of input frames M, to determine the influence of a particular input modality on the recognition result. The sign recognition module was evaluated using a leave-one-subject-out (LOSO) cross-validation scheme on the SHSL dataset. As explained in Section 3.4, the SHLS dataset contains video data of 40 users where every user was recorded performing each of 25 sign language gestures twice. In the proposed validation scheme, analyzed networks are trained with video data collected from 39 subjects, and tested on a single user. The average classification results of 40 users for LOSO are shown in Table 2. From the results, it is visible that network accuracy depends on the length of input sequence M, were using longer input achieves a higher performance. Also, we can inspect that employing Depth modality network performs better in comparison with RGB data, while a significant performance improvement was reported when modality fusion was introduced.

Table 2. Sign language recognition—leave-one-subject-out (LOSO).

	8-Frames	**16-Frames**	**32 Frames**
SHSL–Depth	60.2	66.8	67.4
SHSL–RGB	59.8	62.2	63.4
SHSL–Fusion	60.9	69.8	70.1

To improve the recognition for personalized use, we implemented an additional fine-tuning process in which we retrain dense layers with the first version of users' trials containing each of 25 gestures. Similarly, as in the LOSO scheme, performance evaluation was made by training each model with data of 39 users (1950 videos) together with the first 25 corresponding videos of the current signer. Results in Table 3 show an average accuracy of 40 users from which it is evident that the user adaptation technique can additionally increase performance.

Table 3. Sign language recognition—user adaptation.

	8-frames	16-frames	32 frames
SHSL- Depth	62.2	68.8	69.4
SHSL- RGB	60.3	63.5	63.9
SHSL- Fusion	63.1	71.8	72.2

Performance per user is given in Figure 6. where it is visible that signer #5 achieved the lowest accuracy of 58.8% while signer #2 achieved the best accuracy of 80.9%. Also, Figure 6 enables us to inspect accuracy improvement per user where it is visible that most notable improvement of 7.8% is achieved for signer #21, while there are few cases where this procedure decreases the results as with singer #12 (−3.3%); in the average subject, adaptation delivers an improvement of 2.1%.

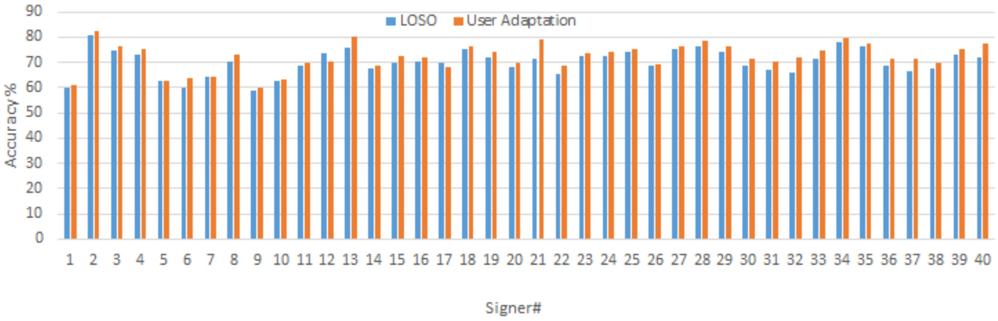


Figure 6. Accuracy per user for LOSO and user adaptation.

4.3. Performance Evaluation of the Sign Command Interface

The purpose of the proposed sign command interface is to interpret a sequence of recognized language signs as command format consisting of three parts: action part (A), device part (D), and location part (L). Given the application domain, the proposed sign language corpus contains 25 language signs grouped according to the command format as 13, 8, and 4 signs, respectively, as action, device, and location part. The performance of each command group is given in the following Tables 4–6. The confusion matrix in Table 4 presents the misclassification between action gestures (1–13) in terms of precision and recall per gesture, with the average classification accuracy of 73.3%. Likewise, the confusion matrix in Table 5 refers to device gestures (14–21) with an average classification accuracy of 70%, and Table 6 refers to location vocabulary (22–25) that accomplishes 71.9% accuracy. From the confusion matrices, we show that knowing the type of gesture in advance can minimize the errors, that is by the rejection of all gestures that do not belong to the observed set precision can reach 90.2%, 81.6%, and 73.1% respectively for A, D and I.

To analyze the possible usability of the proposed system in real-life applications, we tested our solution regarding sentence error rate (SER), and we analyzed the execution time of the proposed wake-up and sign recognition module. In this work, SER was calculated as the percentage of language sign sequences that do not have the exact match with those of reference. Based on directions by deaf people, for this experiment, we defined 15 different combinations of sign language sequences (commands), so that every sign is performed at least once.

Reported results for SER follow the conclusion reached from Figure 6, where the worst performance (practically unusable) was obtained for signer #5, and the best score was reported for user #2, who achieved 40% for LOSO and 33% for user adaptation. Additionally, a refinement procedure was introduced based on following the established command format. This method achieves performance improvement, thus reporting SER of 20% for LOSO and subject adaptation.

Table 4. Confusion matrix regarding actions signs.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**D**	**L**	**Precision**	**Recall**
1	28	0	0	3	0	0	0	8	0	0	0	0	0	1	0	0.7	0.73
2	0	29	0	0	0	0	0	4	0	0	0	0	0	7	0	0.73	0.69
3	1	0	33	0	0	0	0	0	0	1	0	0	0	5	0	0.83	0.80
4	0	1	0	32	0	0	3	0	0	0	0	1	0	0	3	0.8	0.82
5	0	0	1	0	29	0	2	0	0	0	4	0	0	4	0	0.73	0.88
6	0	5	0	0	0	29	0	0	0	2	3	0	0	0	1	0.73	0.81
7	0	0	0	0	0	0	27	0	5	0	0	0	0	4	4	0.68	0.79
8	0	4	0	0	0	0	0	26	0	0	0	7	0	3	0	0.65	0.49
9	7	0	0	0	0	0	0	0	29	0	0	0	0	2	2	0.73	0.62
10	0	0	0	0	0	0	0	0	0	27	0	0	9	2	2	0.68	0.57
11	0	2	0	0	0	1	0	0	0	0	33	1	0	0	3	0.83	0.73
12	0	0	0	0	0	3	0	0	0	0	4	27	0	6	0	0.675	0.66
13	0	0	0	0	0	0	0	6	0	0	0	0	32	2	0	0.8	0.56
D	0	1	7	0	4	1	2	8	4	12	1	0	10	261	9	0.82	0.85
L	2	0	0	4	0	2	0	1	9	5	0	5	6	9	117	0.73	0.83

Table 5. Confusion matrix regarding household devices signs.

	14	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**A**	**L**	**Precision**	**Recall**
14	26	0	6	0	0	0	0	3	5	0	0.65	0.72
15	0	29	0	0	0	0	3	0	8	0	0.73	0.88
16	0	0	28	0	0	4	3	0	5	0	0.7	0.67
17	1	0	0	30	0	0	0	0	7	2	0.75	0.81
18	0	0	0	0	28	0	0	5	7	0	0.7	0.82
19	0	0	2	0	0	27	0	0	10	1	0.68	0.75
20	0	0	0	0	0	0	28	0	6	6	0.7	0.78
21	0	0	0	4	0	4	0	30	2	0	0.75	0.58
A	9	4	6	2	6	0	2	7	469	15	0.90	0.85
L	0	0	0	1	0	1	0	7	34	117	0.73	0.83

Table 6. Confusion matrix regarding house locations signs.

	22	**23**	**24**	**25**	**A**	**D**	**Precision**	**Recall**
22	30	0	0	0	10	0	0.75	0.88
23	1	29	1	0	9	0	0.725	0.86
24	0	0	28	0	4	8	0.7	0.72
25	0	0	0	28	11	1	0.7	0.82
A	2	5	4	4	469	36	0.90	0.85
D	1	0	6	2	50	261	0.82	0.85

To meet the design requirements concerning real-time application, we based our practical solution on the Nvidia TX2 platform employing a TensorRT framework to achieve low latency runtime and high-throughput for deep learning applications. TensorRT-based solutions deliver up to 40× better performance than CPU counterparts [50], which enables our system to perform efficient subject adaptation procedures in real-time. To analyze the effectiveness of the proposed solution, we measured the execution time of each component in the prediction phase. Using a data batch size of 10, the recorded execution time of 180 ms, 290 ms, and 510 ms, respectively, and the wake up module, unimodal subnetwork, and modality fusion demonstrated that the proposed solution could maintain real-time processing criterion using a power-efficient embedded AI computing device.

5. Conclusions

Sign language is a primary form of communication for deaf and hard-of-hearing people. This visual language is established with its own vocabulary and syntax, which poses a serious challenge for the deaf community integrating with social and work environments. To assist the interaction of deaf and hard-of-hearing people, we introduced an efficient smart home automatization system. The proposed system integrates a touchless interface tailored for sign language users. In collaboration with deaf people, a small Croatian sign language database was created, containing 25 different language signs within the vocabulary specifically intended for a smart home automatization. For developing a real-time application, we presented a novel hierarchical architecture that consists of a lightweight wake-up module and a high-performance sign recognition module. The proposed models were based on employing Conv3D to extract spatiotemporal features. To obtain high-performance classification, in this work, we performed a multi-modal fusion with the temporal alignment between two input modalities—RGB and depth modalities. Moreover, effective spatiotemporal data augmentation was applied to obtain better accuracy and to prevent overfitting of the model. The evaluation results demonstrate that the proposed sign classifier can reach reasonable accuracy recognizing individual language sings in a LOSO scheme. We also demonstrated the improvement of the results when subject adaption was performed. The performance of the whole system was given in terms of the sentence error rate. From the results presented in Section 3, we can conclude that the proposed Sign command interface can be efficiently used within the smart home environment. The online phase of the proposed real-time system was, as a practical realization, implemented and tested on the Nvidia Jetson TX2 embedded system with Stereolabs ZED M stereo camera.

In future work, research efforts will be made towards the continuous recognition of sign language in a smart home environment. We plan to include the expansion of the vocabulary of our Croatian sign language database, followed by developing a more complex grammar and language model. Improvements in deep learning framework will also be made, which will include a higher degree of user modalities such as facial expression, head, and body postures.

Author Contributions: Conceptualization and methodology, L.K. and M.R.; investigation, validation, and software L.K.; funding acquisition, M.R.; visualization, M.P.; data curation, M.P. and M.Š.; supervision, M.R. and M.Š.; writing—original draft, L.K. and M.P.; writing—review and editing M.R. and M.Š. All authors have read and agreed to the published version of the manuscript.

Funding: The work of doctoral student Luka Kraljević has been fully supported by the "Young researchers' career development project—training of doctoral students" of the Croatian Science Foundation.

Acknowledgments: We gratefully acknowledge the support of Dina Lijić and Toni Ljubić for their contribution in developing SHSL database.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. World Health Organization. *World Report on Disability 2011*; WHO: Geneva, Switzerland, 2011.
2. Shahrestani, S. Internet of Things and Smart Environments. In *Assistive Technologies for Disability, Dementia, and Aging*; Springer International Publishing: Cham, Switzerland, 2017.
3. World Health Organization. *Millions of People in the World Have Hearing Loss that Can Be Treated or Prevented*; WHO: Geneva, Switzerland, 2013; pp. 1–17.
4. Köpüklü, O.; Gunduz, A.; Kose, N.; Rigoll, G. Real-time hand gesture detection and classification using convolutional neural networks. In Proceedings of the 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 14–18 May 2019; pp. 1–8.
5. Gaglio, S.; Re, G.L.; Morana, M.; Ortolani, M. Gesture recognition for improved user experience in a smart environment. In Proceedings of the Congress of the Italian Association for Artificial Intelligence, Turin, Italy, 4–6 December 2013; pp. 493–504.
6. Sandler, W.; Lillo-Martin, D. *Sign Language and Linguistic Universals*; Cambridge University Press: Cambridge, UK, 2006.

7. Lewis, M.P.; Simons, G.F.; Fennig, C.D. *Ethnologue: Languages of the World*; SIL International: Dallas, TX, USA, 2009; Volume 12, p. 2010.
8. Neiva, D.H.; Zanchettin, C. Gesture recognition: A review focusing on sign language in a mobile context. *Expert Syst. Appl.* **2018**, *103*, 159–183. [CrossRef]
9. Ahmed, M.A.; Zaidan, B.B.; Zaidan, A.A.; Salih, M.M.; Lakulu, M.M.b. A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. *Sensors* **2018**, *18*, 2208. [CrossRef] [PubMed]
10. Zhang, Y.; Cao, C.; Cheng, J.; Lu, H. Egogesture: A new dataset and benchmark for egocentric hand gesture recognition. *IEEE Trans. Multimedia* **2018**, *20*, 1038–1050. [CrossRef]
11. Materzynska, J.; Berger, G.; Bax, I.; Memisevic, R. The Jester Dataset: A Large-Scale Video Dataset of Human Gestures. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
12. Wong, S.C.; Gatt, A.; Stamatescu, V.; McDonnell, M.D. Understanding data augmentation for classification: When to warp? In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, Australia, 30 November–2 December 2016; pp. 1–6.
13. Narayana, P.; Beveridge, R.; Draper, B.A. Gesture recognition: Focus on the hands. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5235–5244.
14. Li, S.Z.; Yu, B.; Wu, W.; Su, S.Z.; Ji, R.R. Feature learning based on SAE–PCA network for human gesture recognition in RGBD images. *Neurocomputing* **2015**, *151*, 565–573. [CrossRef]
15. Liu, T.; Zhou, W.; Li, H. Sign language recognition with long short-term memory. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 2871–2875.
16. Huang, J.; Zhou, W.; Li, H.; Li, W. Sign language recognition using 3d convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Multimedia and Expo (ICME), Torino, Italy, 29 June–3 July 2015; pp. 1–6.
17. Park, C.B.; Lee, S.W. Real-time 3D pointing gesture recognition for mobile robots with cascade HMM and particle filter. *Image Vis. Comput.* **2011**, *29*, 51–63. [CrossRef]
18. Bajpai, D.; Porov, U.; Srivastav, G.; Sachan, N. Two way wireless data communication and american sign language translator glove for images text and speech display on mobile phone. In Proceedings of the 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 4–6 April 2015; pp. 578–585.
19. Seymour, M.; Tšoeu, M. A mobile application for South African Sign Language (SASL) recognition. In Proceedings of the IEEE AFRICON 2015, Addis Ababa, Ethiopia, 14–17 September 2015; pp. 1–5.
20. Devi, S.; Deb, S. Low cost tangible glove for translating sign gestures to speech and text in Hindi language. In Proceedings of the 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 9–10 February 2017; pp. 1–5.
21. Jin, C.M.; Omar, Z.; Jaward, M.H. A mobile application of American sign language translation via image processing algorithms. In Proceedings of the 2016 IEEE Region 10 Symposium (TENSYMP), Bali, Indonesia, 9–11 May 2016; pp. 104–109.
22. Rao, G.A.; Kishore, P. Selfie video based continuous Indian sign language recognition system. *Ain Shams Eng. J.* **2018**, *9*, 1929–1939. [CrossRef]
23. Luo, R.C.; Wu, Y.; Lin, P. Multimodal information fusion for human-robot interaction. In Proceedings of the 2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, 21–23 May 2015; pp. 535–540.
24. Starner, T.; Weaver, J.; Pentland, A. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1371–1375. [CrossRef]
25. Dardas, N.H.; Georganas, N.D. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *IEEE Trans. Instrum. Measur.* **2011**, *60*, 3592–3607. [CrossRef]
26. Wang, S.B.; Quattoni, A.; Morency, L.P.; Demirdjian, D.; Darrell, T. Hidden conditional random fields for gesture recognition. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1521–1527.

27. Kopinski, T.; Magand, S.; Gepperth, A.; Handmann, U. A light-weight real-time applicable hand gesture recognition system for automotive applications. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 336–342.
28. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Li, F.-F. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 1725–1732.
29. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *Adv. Neural Inf. Process. Syst.* **2014**, *1*, 568–576.
30. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
31. Li, Z.; Gavrilyuk, K.; Gavves, E.; Jain, M.; Snoek, C.G. Videolstm convolves, attends and flows for action recognition. *Comput. Vis. Image Understand.* **2018**, *166*, 41–50. [CrossRef]
32. Hakim, N.L.; Shih, T.K.; Arachchi, K.; Priyanwada, S.; Aditya, W.; Chen, Y.C.; Lin, C.Y. Dynamic Hand Gesture Recognition Using 3DCNN and LSTM with FSM Context-Aware Model. *Sensors* **2019**, *19*, 5429. [CrossRef] [PubMed]
33. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
34. Hara, K.; Kataoka, H.; Satoh, Y. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6546–6555.
35. Ariesta, M.C.; Wiryana, F.; Kusuma, G.P. A Survey of Hand Gesture Recognition Methods in Sign Language Recognition. *Pertan. J. Sci. Technol.* **2018**, *26*, 1659–1675.
36. Zhu, G.; Zhang, L.; Shen, P.; Song, J. Multimodal gesture recognition using 3-D convolution and convolutional LSTM. *IEEE Access* **2017**, *5*, 4517–4524. [CrossRef]
37. Neverova, N.; Wolf, C.; Taylor, G.W.; Nebout, F. Multi-scale deep learning for gesture detection and localization. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 474–490.
38. Rahim, M.A.; Islam, M.R.; Shin, J. Non-Touch Sign Word Recognition Based on Dynamic Hand Gesture Using Hybrid Segmentation and CNN Feature Fusion. *Appl. Sci.* **2019**, *9*, 3790. [CrossRef]
39. Tran, D.S.; Ho, N.H.; Yang, H.J.; Baek, E.T.; Kim, S.H.; Lee, G. Real-Time Hand Gesture Spotting and Recognition Using RGB-D Camera and 3D Convolutional Neural Network. *Appl. Sci.* **2020**, *10*, 722. [CrossRef]
40. Ohn-Bar, E.; Trivedi, M.M. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2368–2377. [CrossRef]
41. Konečný, J.; Hagara, M. One-shot-learning gesture recognition using hog-hof features. *J. Mach. Learn. Res.* **2014**, *15*, 2513–2532.
42. Ko, S.K.; Kim, C.J.; Jung, H.; Cho, C. Neural sign language translation based on human keypoint estimation. *Appl. Sci.* **2019**, *9*, 2683. [CrossRef]
43. Forster, J.; Schmidt, C.; Koller, O.; Bellgardt, M.; Ney, H. *Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather*; LREC: Baton Rouge, LA, USA, 2014; pp. 1911–1916.
44. Quesada, L.; López, G.; Guerrero, L. Automatic recognition of the American sign language fingerspelling alphabet to assist people living with speech or hearing impairments. *J. Ambient Intell. Humaniz. Comput.* **2017**, *8*, 625–635. [CrossRef]
45. Chen, F.; Deng, J.; Pang, Z.; Baghaei Nejad, M.; Yang, H.; Yang, G. Finger angle-based hand gesture recognition for smart infrastructure using wearable wrist-worn camera. *Appl. Sci.* **2018**, *8*, 369. [CrossRef]
46. Pabsch, A.; Wheatley, M. *Sign Language Legislation in the European Union–Edition II*; EUD: Brussels, Belgium, 2012.
47. Bambach, S.; Lee, S.; Crandall, D.J.; Yu, C. Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Araucano Park, Las Condes, Chile, 11–18 December 2015.

48. Molchanov, P.; Yang, X.; Gupta, S.; Kim, K.; Tyree, S.; Kautz, J. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4207–4215.
49. Joze, H.R.V.; Koller, O. Ms-asl: A large-scale dataset and benchmark for understanding american sign language. *arXiv* **2018**, arXiv:1812.01053.
50. NVIDIA TensorRT. Available online: https://developer.nvidia.com/tensorrt (accessed on 7 February 2020).