STAG: Smart Tools and Applications in Graphics (2021)
P. Frosini, D. Giorgi, S. Melzi, and E. Rodolà (Editors)

STRONGER: Simple TRajectory-based ONline GEsture Recognizer

M. Emporio[1] A. Caputo[1] A. Giachetti[1]

[1]Univerity of Verona, Italy

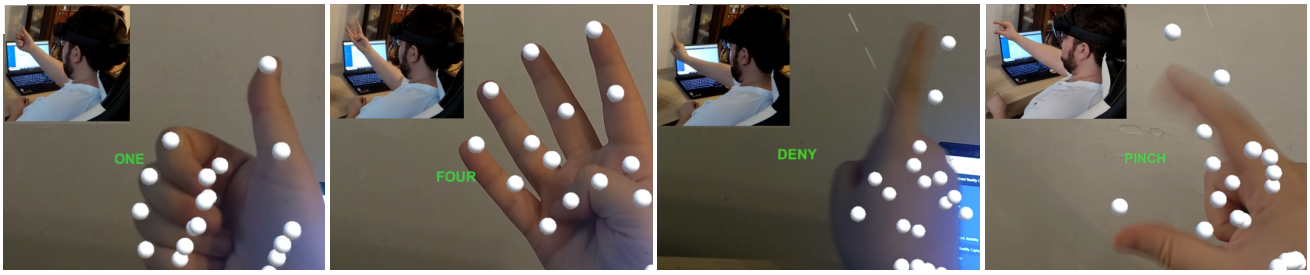

Figure 1: *Snapshots showing the online display viewed by the subject wearing a Hololens 2 running the online recognizer demo.*

Abstract

In this paper, we present STRONGER, a client-server solution for the online gesture recognition from captured hands' joints sequences. The system leverages a CNN-based recognizer improving current state-of-the-art solutions for segmented gestures classification, trained and tested for the online gesture recognition task on a recent benchmark including heterogeneous gestures. The recognizer provides good classification accuracy and a limited number of false positives on most of the gesture classes of the benchmark used and has been used to create a demo application in a Mixed Reality scenario using an Hololens 2 optical see through Head-Mounted Display with hand tracking capability.

CCS Concepts

• ***Computing methodologies*** → ***Neural networks;*** • ***Human-centered computing*** → ***Gestural input;***

1. Introduction

User interfaces based on 3D mid-air gestures are expected to become popular in the next years, as they are a viable solution for natural interaction in mixed reality not requiring specific devices and being suitable for being used in crowded or noisy environments [GLY21]. Currently, a relevant number of interactive systems in many domains already feature this kind of interface: virtual environments, smart surveillance systems, teleconferencing, home or car entertainment controls, virtual Personal Aerobics Trainers (PAT), and so on. Creating advanced touchless user interfaces is also fundamental to maintain a high level of hygiene for work environments or in public terminals, and this can solve several issues related to the pandemic emergence. These systems are promising but are typically limited to the recognition of a few simple gestures. A relevant amount of recent research work is therefore dedicated to the development of more advanced and flexible gesture recognizers able to deal with more complex dictionaries and gesture types still performing real-time recognition. Many of these methods are based on the processing of hand pose (skeleton) streams, enabled by popular low-cost devices like the Leap Motion or the availability of effective network-based solutions for hand pose tracking like Mediapipe [ZBV*20].

Skeleton-based gesture recognizers need to capture the fine differences among gestures and distinguish one gesture from another, ideally with a high degree of confidence but also being able to avoid false positives, e.g. detection of gestures corresponding to non-significant actions. This is a challenge of fundamental importance to create interfaces with a reasonable usability degree.

Several solutions for the skeleton-based recognition of heterogeneous gestures have been proposed and work well on offline classification benchmarks like SHREC'17 [DSWV*17].

However, the offline classification task does not test the ability to avoid false positives with hard time constraints in an online recognition scenario (i.e. sequential processing of the datastream and on-the-fly classification) and the methods are not demonstrated in practical applications or interactive mockups.

DOI: 10.2312/stag.20211481

In this work, we propose STRONGER, an online skeleton-based gesture recognizer that can be used to develop mid-air deviceless interfaces. We test it on a recent online classification benchmark [CGS*21], and demonstrate it within a prototype mixed reality interface.

The recognizer is based on a revised version of DDNet [YSWN19], modified adding novel features to handle a larger set of gestures and to perform online classification. The prototype application is based on Hololens 2 and a client-server architecture to process hand pose streams.

The results obtained in the online benchmark and our preliminary tests on the prototype show the feasibility of a gesture-based user interface for mixed reality able to recognize a large dictionary of heterogeneous gestures.

2. Related Work

Several works in the literature show the need to implement specific solutions for hand gesture and action recognition considering the complex nature of the hand movements, different from those applied in full-body skeleton-based action recognition or similar tasks.

While a number of relevant works rely on methods such as SVMs, Random Forests, dissimilarity-based classifiers, etc. [MDZ16] [CGG*20], the use of neural networks has become a dominant trend in this context and provides promising results. However, there are still open issues that are still considered a challenge.

A popular solution considered suitable for time-series like hand pose streams is the use of recurrent networks. In [ABC*18] a stack of LSTM units is trained by using as features the angles formed by the finger bones of human hands. The Deep Gesture Recognition Utility [MLJ18] is based on a set of stacked gated recurrent units (GRU) [CvMG*14] and a global attention model. The authors demonstrate that GRUs are fast to train and produce good results. A lightweight version of this system, combined with a smart data augmentation method provided the best results in an online gesture recognition contest [CBP*19]. Nguyen et al. [NBLB21] exploited high-order statistics of hand poses coupled with the Statistical Recurrent Units architecture.

Graph-based solutions have been proposed to exploit the relationships between hand joints. Li et al [LHY*19] construct graphs with three types of edges to finely describe the linkage action of joints. An end-to-end deep neural network is then used for the classification, where the convolution is conducted only on linked skeleton joints.

Guo et al. [GHZ*21] propose a novel edge-varying graph together with a normalized edge convolution operation, and a zig-zag sampling strategy. Based on these innovations, they create spatial-based Graph Convolution Networks called normalized edge convolutional networks for hand gesture recognition.

In [CZP*19] the authors build a fully connected graph from the hand skeleton and learn node features and edges via a self-attention mechanism that performs in both spatial and temporal domains.

Hou et al. [HWC*18] propose an end-to-end Spatial-Temporal Attention Residual Temporal Convolutional Network (STA-Res-TCN) which learns different levels of attention and assigns them to each spatial-temporal feature extracted by the convolution filters at each time step.

A relevant issue making the recognition of heterogeneous hand gestures (for example, compared with action recognition) is that the gesture classes may be differentiated by different features, which may be in some cases subtle. Heterogeneous gesture recognition benchmarks like Shrec'17 [DSWV*17] or SFINGE3D [CGG*20] features "coarse" gestures characterized by long whole-hand trajectories (lasting 1 second and more) and "fine" gestures characterized by fast changes of finger articulations (lasting 100 milliseconds or less).

To solve this issue, Li et al. [LLG*21] propose a two-stream neural network with one stream being an adaptive self-attention based graph convolutional network (SAGCN) extracting the short-term temporal information and hierarchical spatial information, and the other being a residual-connection enhanced bidirectional Independently Recurrent Neural Network (RBi-IndRNN) to extract long-term temporal information. The method has been tested with promising results on the DHG gesture dataset [DSWV16] and the FPHA hand action dataset [GHYBK18].

Yang et al. [YSWN19] noted that a very simple network architecture based on 1D convolutions, fed with simple features derived from the hand joints sequence and using a motion summarization module to reduce noise from non-relevant frames, can provide state-of-the-art results with reduced computational complexity.

The biggest problem with these gesture classifiers is that they cannot be directly used to create a gesture-based interface. An interface of this kind needs to detect gestures in a continuous stream of hand poses and correctly classifying them, avoiding missing relevant gestures and false detections.

Benchmarks only testing the accuracy in the classification don't test the performances of online detection, requiring continuous input/output and the addition of a "non-gesture" label for the characterization of non-meaningful sequences. A recurrent network, but also a generic classifier with a coupled detection module or a sliding window approach can provide the continuous input/output, but the training of the methods using labeled sequences is not trivial. It is hard to collect a large amount of labeled data and the non-gesture class is typically quite different from the others due to the variability of the elements and the larger number of examples available.

A benchmark specifically designed for online classification of heterogeneous hand gestures, including coarse, fine, and also static gestures has been proposed in [CGS*21]. This dataset includes training and test sequences including gestures and non-gesture frames and can be used to validate online classification approaches that can directly be used to build gestural interfaces. We exploited this dataset to train our system and used it also to assess the online performances.

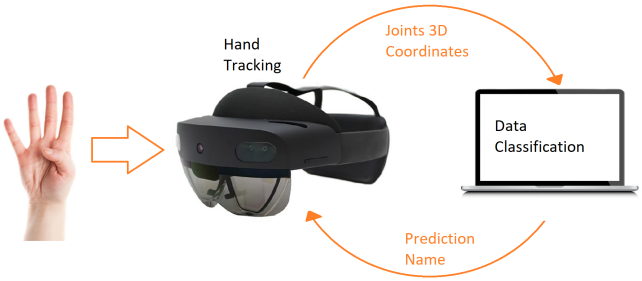

Figure 2: *The client-server application: the Hololens2 app sends the joints' stream to the PC and receives the classification results to be displayed.*

3. Proposed approach

3.1. System architecture

In our work, we not only design and implemented a network solution for gesture recognition, but we also developed a prototype gesture-based interface for mixed reality running on a Hololens 2 head-mounted display.

Hololens 2 is provided with a hand tracking module able to capture hand skeletons (20 joints) is similar to the one captured with the Leap Motion and this allows us to use an online classifier trained on existing gesture datasets captured with the latter device for online gesture recognition on the Hololens 2. We plan, however, to capture further training data with the Hololens 2 finger tracking system in the near future.

The interactive application works with a client-server architecture, sending raw skeletons streams over TCP to a remote server where a Python application performs the recognition and sends back the results to the app running on the Hololens.

Figure 2 shows the operation scheme of the client-server architecture.

3.2. Gesture classification network

As it is based on a simpler 1D convolutional neural network and it provides the state-of-the-art performances on the SHREC 2017 benchmark [DSWV*17], we chose DDNet [YSWN19] as the base method to build our online recognition system. However, we noted that this method does not perform well on the offline classification of the gestures of the SHREC 2021 benchmark. This is not due to limitations of the network architecture, but it depends on the data used to feed it, that it is not the complete stream, but a set of hand-crafted features derived from it and does not include information that is important to disambiguate gesture classes. In detail, the two network branches of the original network process joints' velocities and between-joints distances, meaning that the evolution of the hand's orientation is lost. This makes it not possible, for example, to distinguish the classical menu gesture with the hand open with the palm in front of the user view from another static gesture keeping the hand open parallel to the floor.

For this reason, we added two branches to the 1D convolutional network, one processing the sequence of palm orientations and the second processing a set of unit vectors representing the directions defined by couples of joints.

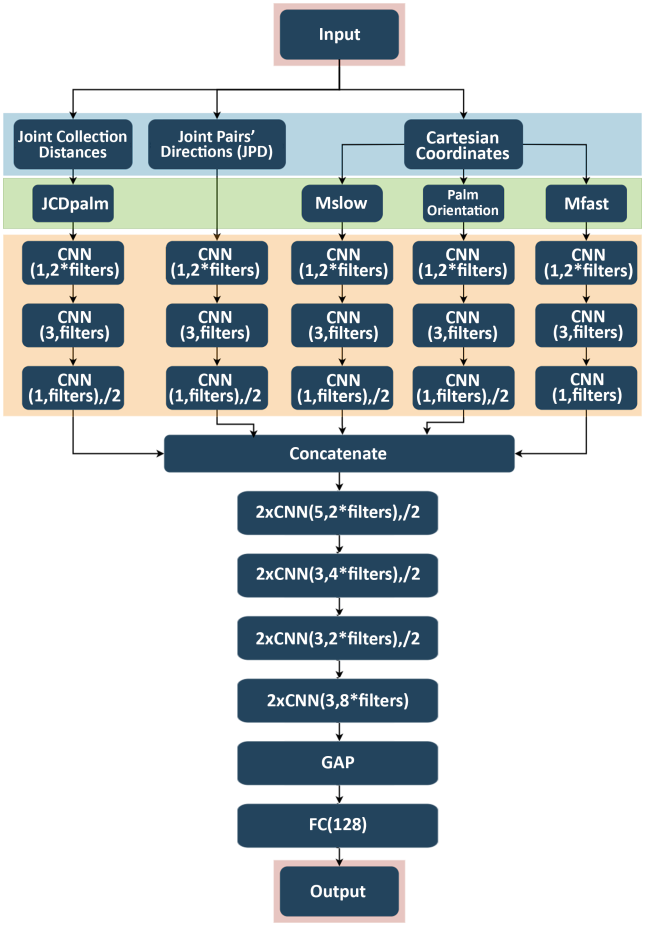

Figure 3: *Our classifier adds to the DDNet architecture two branches with 1D convolutions processing palm orientation data and joint pairs' directions.*

The complete architecture of the proposed network is represented in Figure 3.

Five 1D-CNN branches are used to process different 1D features derived from the input hand pose sequence, that is resampled to a fixed number of time steps. The first feature is the Joint Collection of Distances (JCD) that is the matrix of the Euclidean distances between hand joints across time flattened to become a one-dimensional vector. In particular, the matrix JCD^k (shown in equation 1 as reported in [YSWN19]) is an N-1-by-N-1 with N representing the index of the joint.

$$JCD^k = \begin{bmatrix} \left\| \overrightarrow{J_2^k J_1^k} \right\|_2 & & \\ \vdots & \ddots & \\ \vdots & \cdots & \ddots \\ \left\| \overrightarrow{J_N^k J_1^k} \right\|_2 & \cdots & \cdots & \left\| \overrightarrow{J_N^k J_{N-1}^k} \right\|_2 \end{bmatrix} \quad (1)$$

that is then flattened to be a one dimensional vector to be used as network input.

$$\left\| \overrightarrow{J_i^k J_j^k} \right\|_2, \; i \neq j \; and \; i \in [2,N], \; j \in [1,N-1]$$

indicates the Euclidean distance between vectors $J_i^k = (x,y,z)^k$ containing the coordinates of the joint at the frame k. The other features used also in the original DDNet are Mslow and MFast, that are joints speeds (differences between positions at consecutive times) computed at two different scales (reported again in [YSWN19]).

The two 1D inputs added are the palm orientation (PO) represented by the normal vector of the hand's palm across time ($PO^k = (x_{po}, y_{po}, z_{po})^k$), and a subset of the directions of joint pairs (Joint Pairs' Directions, JPD), obtained as the difference of selected joints' vectors

$$JPD^k = \begin{bmatrix} \overrightarrow{J_2^k J_1^k} & & \\ \vdots & \ddots & \\ \overrightarrow{J_N^k J_1^k} & \cdots & \overrightarrow{J_N^k J_{N-1}^k} \end{bmatrix} \quad (2)$$

as shown in Figure 4. The vector components of the matrix are flattened as well to be used in the 1D convolutional network.

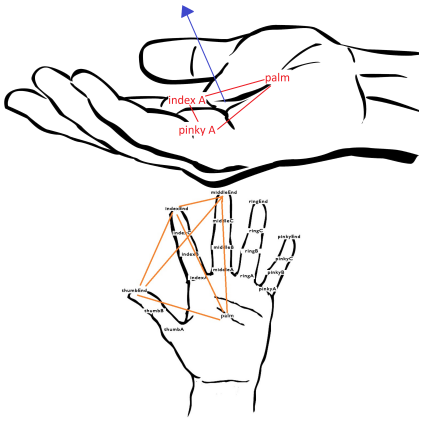

Figure 4: *The novel feature are the time evolution of the palm orientation (left) and of the directions defined by selected pairs of joints (JPD, right)*

All the inputs are processed by similar CNN branches to derive intermediate representations that are then concatenated and passed through 3 more convolutional layers, a Global Average Pooling (GAP) and a Fully Connected layer (FC) providing the class probabilities.

The training code takes hand pose time sequences corresponding to labelled gestures, re-samples the trajectories with a fixed number of elements, pre-processes them creating the gesture descriptors feeding the network branches and train the classifier by minimizing the cross-entropy loss.

To handle gesture classes together with non-gestures, we trained the network with the gesture windows corresponding to the annotations with the corresponding classes and a set of non-gestures example cropped randomly outside the gesture time frames and assigned to a 19th class.

3.3. Online recognition and the SHREC'21 dataset

In the offline benchmark, it is sufficient to train the network with segmented gestures using the procedure described above and test the automatic labelling provided by the trained classifier on the segmented gestures to assess the recognition accuracy.

The online recognition task is intrinsically different, as the gestures should be detected and segmented, or they must be recognized in a continuous stream. As the proposed network is quite efficient, this can be avoided by using a sliding window approach, testing windows of multiple lengths. However, this means that we need to consider the non-gesture class assignment, and there is a strong class imbalance in the data, as most of the windows cropped from the sequences are labelled as non-gesture.

A specific method to train the recognizer needs therefore to be designed for the task and different evaluation methods should be applied to assess the quality of the results.

The only benchmark available to test online detection of complex gestures from hand skeletons' sequences has been proposed in the SHREC'21 contest on Skeleton-based Hand Gesture Recognition in the Wild [CGS*21]. We decided to use the dataset of this contest both to train the classifier and to implement the mixed reality recognizer application, as it features a sufficiently large and heterogeneous gesture dictionary and an online evaluation protocol.

This dataset features 20-nodes hand skeletons' sequences including several examples of gestures interleaved with non-meaningful gesticulation and it is divided into a training set with 108 sequences including 24 examples for each gesture class and a test set of 72 sequences including 16 examples for each gesture class. Hand pose sequences in this benchmark have been acquired with a Leap Motion device mounted as in a Head-Mounted display configuration and with a frequency of 50 skeletons per second. This makes the data quite similar to those produced by the Hololens 2 capturing gesture streams at 45 fps from a similar point of view [UBG*20], and this makes possible to train the method on the contest data and to test recognition both on Leap Motion and Hololens 2 data streams.

The gesture dictionary includes 18 classes divided in 3 types:

Static 7 classes characterized by a hand pose kept fixed for at least one second (One, Two, Three, Four, OK, Menu, Pointing)

Dynamic coarse 5 classes characterized by a single global trajectory of the hand (Left, Right, Circle, V, Cross)

Dynamic fine 6 classes characterized by variations in the fingers' articulation (Grab, Pinch, Tap, Deny, Knob, Expand)

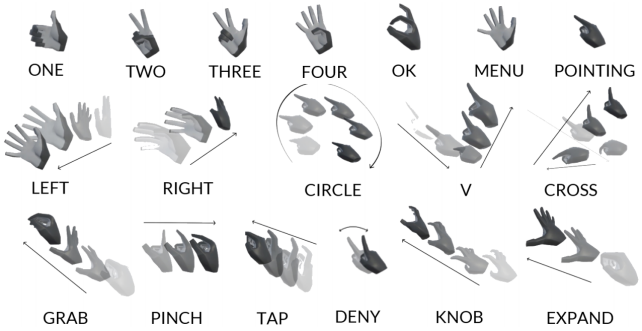

Figure 5: *Gesture dictionary of SHREC'21 (from [CGS*21])*

These gestures, shown in Figure 5 are interleaved in the sequences with non-significant hand motions of various types. In our experiments on SHREC 2021, we trained the modified DDNet using cropped sequences representing both segmented gestures of the different classes defined in the dictionary and non-gesture sequences obtained by extracting sequences of random length from $0.2s$. to $1.2s$.

The trained network can be fed with windows of skeleton's streams and outputs an array with the probabilities that it belongs to each class.

For the online classification, we don't simply assign to the corresponding frames the label corresponding to the maximal probability, but we introduce a threshold for each gesture class, corresponding to the minimum probability making the recognition acceptable. All the gestures detected in the sliding window procedure with a probability lower than the corresponding threshold are then discarded and treated as non-gesture. This should reduce the false positive detections due to the limited representativity of the non-gesture examples.

Thresholds are learned from training data as follows: during the offline training, the average probability estimated by the classifiers for the gestures belonging to that class is set as initial thresholds estimates.

These thresholds are then refined with a specific procedure optimizing them on the training sequences (including the labeled gestures interleaved with non-gesture movements). For each class, if the number of false positives is larger than a target value (FP-ratio > 0.5), the probability threshold is increased until the target is reached or the accuracy falls below 60%

The thresholds learned with this procedure are stored in a specific array called Probability Threshold Array (PTA) and used in online detection.

Another class-specific parameter is learned from training data and used to improve the quality of online detection. We call it Window Thresholds Array (WTA) and defines the minimum and maximum acceptable duration for each class. Each gesture to be executed needs a different number of frames, so if a short gesture is recognized in a very large window (or vice versa), and should be discarded. The time length of the gestures featured in the SHREC'21 benchmark is quite variable ranging from 0.18s on average for gestures like EXPAND or TAP to several seconds for the static ones (see Table 1).

Gesture	type	avg #frames	avg. time (s)
ONE	static	149	2,98
TWO	static	169	3,38
THREE	static	166	3,32
FOUR	static	163	3,26
OK	static	148	2,96
MENU	static	151	3,02
POINTING	static	97	1,94
LEFT	dynamic coarse	23	0,46
RIGHT	dynamic coarse	17	0,34
CIRCLE	dynamic coarse	54	1,08
V	dynamic coarse	27	0,54
CROSS	dynamic coarse	43	0,86
GRAB	dynamic fine	14	0,28
EXPAND	dynamic fine	9	0,18
PINCH	dynamic fine	22	0,44
TAP	dynamic fine	12	0,24
DENY	dynamic fine	78	1,56
KNOB	dynamic fine	45	0,9

Table 1: *Duration of the 18 gestures of the SHREC'21 dataset. There is a huge variability among the gesture classes.*

We set as acceptable duration limits for each class in the WTA the minimal length in the training set diminished by 0.2 s and the maximal length in the training set increased by 0.2 s. All the gestures recognized in the sliding window procedure are discarded if the window size of the corresponding class is outside the limits defined in the WTA.

We set also a global limit to the maximal length of the gesture tested which is also useful to reduce the delay in the online procedure. This limit is set to 1 s.

In the sliding window procedure, time samples of different sizes (from 5 frames to 50 frames with a step of 5 frames) are slid along the signal with a fixed shift of 5 frames (0.1s).

If there is a gesture prediction corresponding to a window, a vote for the corresponding class is assigned to the superimposed frames, and the frame label is assigned as the one with maximal votes. A gesture is finally detected as a set of consecutive frames with the same assigned label.

3.4. Mixed reality interactive mockup

The client application running on the Hololens 2 device has been created using the Unity game engine and the Mixed Reality Toolkit (MRTK) that has been used in this project. This toolkit simplifies the development of XR headsets, providing a cross-platform input system, a basic set of components and features and common building blocks for spatial interactions in Unity.

The application developed is a very simple mock-up superimposing to the user view 3D primitives representing the captured hand joints and a text string with the currently recognized gesture.

The application is developed with Unity and features C # scripts parsing and sending to the server the joint positions stream provided by the MRTK API and getting back the current classification labels, that are displayed on the semitransparent screen (See Figure 1).

4. Results

4.1. Implementation details

The network code has been developed using PyTorch and CUDA and it has been trained and tested on a Lenovo Legion 5 PC with a RAM of 16 Gb a Nvidia RTX 2060 (6Gb) graphics card. The network architecture features a set of parameters that have been tuned on the SHREC'21 training set, namely filters' size, number of training epochs, batch size, and, most important, the number of sample for input gesture resize. This value was set to 34. Different Python scripts have been developed for offline classification testing of segmented gestures, for the hand pose sequences processing following the SHREC'21 protocol using the sliding window approach, and for the online recognition receiving the Hololens 2 stream and providing it with the online classification results.

4.2. Offline evaluation of the gesture classifier

Before evaluating the online performances of our method, we performed a couple of tests to evaluate the offline classification performances of our implementation. Table 2 shows that the addition of the novel features results in a 10% increase in the accuracy of the classification of segmented test set gestures of SHREC'21. The reason for the difference is the large amount of information lost in the original encoding (not including, for example, any hint on the spatial orientation of the hand).

Figure 6, comparing the confusion matrices of the two classifiers, clearly shows that the modified version solve relevant issues of the original methods in recognizing gestures 3,5,7 (FOUR, MENU, LEFT) where the hand orientation information can disambiguate the gesture from some non-gesture movements. The classification of KNOB gestures (17) is also relevantly improved.

As the network is trained with segmented gestures, we could also evaluate the offline classification performances of our modified network on the popular SHREC '17 [DSWV*17] benchmark. Here the improvement with respect to the original DDNet is not similarly high, but, in any case, our modified network performs slightly better than the original and better than the other method proposed in the literature (see Table 3).

Method	Accuracy
DD-Net [YSWN19]	87.8%
Our model	97.5%

Table 2: *The additional features added to DDNet strongly improve the classification accuracy on SHREC 2021 gestures*

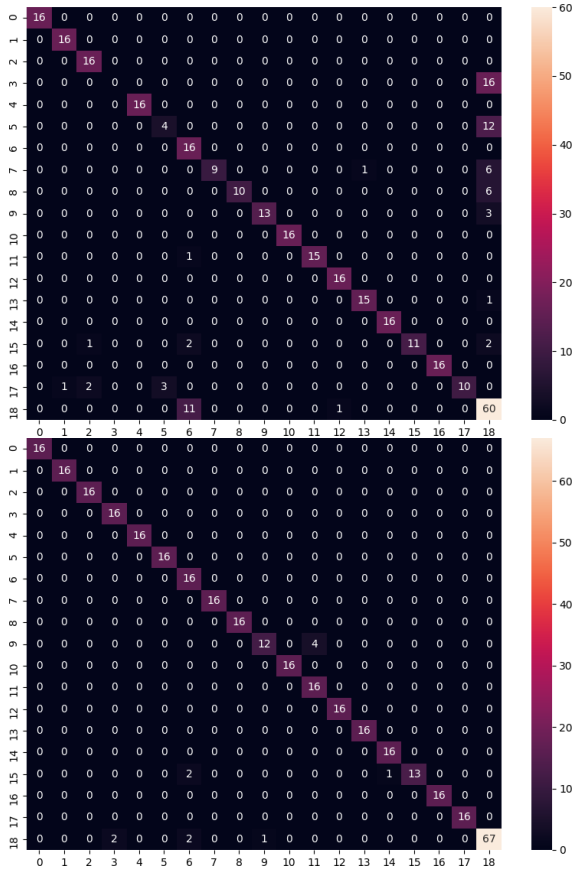

Figure 6: *Top: confusion matrix of DD-net on SHREC 2021 classes. Bottom: confusion matrix of the modified network with the additional features PO and JPD.*

Methods	Accuracy
3 Cent [ZLX17]	77.9%
Key-frame CNN [DSWV*17]	82.9%
Dynamic hand [MZW*20]	88.2%
CNN+LSTM [NCP*18]	89.8%
MFA-Net [XWG*19]	91.3%
Parallel CNN [DXMY18]	91.3%
STA-Res-TCN [HWC*19]	93.6%
DD-Net [YSWN19]	94.6%
Our model	95.0%

Table 3: *Results on SHREC 2017 (Using 3D skeletons only)*

4.3. Online evaluation

For the online evaluation we followed the protocol of SHREC '21: using the procedure described in Section 3.3, we obtained a per frame labeling of the sequences and the timestamps of gesture beginning and end. We could therefore estimate the "detection rate" in the test data, e.g. the percentage of predicted gestures (of each class) corresponding to ground truth ones correctly detected (where corresponding means at least 50% of overlap), the false-positive ra-

Method	Det. Rate	FP Rate	JI	Time(s)
Group 1-Run 3 [CGS*21]	0.729	0.257	0.603	1.36
Group 2-Run 1 [CGS*21]	0.486	0.927	0.277	0.41
Group 3-Run 2 [CGS*21]	0.757	0.340	0.619	**0.3e-5**
Group 4-Run 3 [CGS*21]	0.899	**0.066**	**0.853**	0.16
Our method	**0.906**	0.347	0.740	0.10

Table 4: *Average scores for the SHREC'21 benchmark compared with contest participants. It should be noted that the method of Group 4 includes a gesture segmentation module to reduce false positives.*

Method	Det. Rate	FP Rate	JI
Orig.DDNet	0.858	2.052	0.353
DDNet+PO+JPD	0.944	1.896	0.431
DDNet+PO+JPD+WTA	0.906	0.347	0.740

Table 5: *Ablation study showing that the novel 1D features strongly increase the detection rate while the learning of the Wondow Threshold Array is effective in reducing the false positives rate.*

Gesture	Det.rate	FP ratio	JI	Thr.
ONE	1.000	0.000	1.00	9
TWO	1.000	0.062	0.94	8
THREE	1.000	0.062	0.94	8.2
FOUR	1.000	0.312	0.76	8.7
OK	1.000	0.062	0.94	9.2
MENU	1.000	0.062	0.94	7.6
POINTING	0.937	0.062	0.88	7.5
LEFT	0.937	0.000	0.93	7
RIGHT	0.937	0.250	0.75	7.2
CIRCLE	0.750	1.500	0.30	7
V	0.937	0.187	0.78	6
CROSS	0.687	1.187	0.31	6.5
GRAB	1.000	0.375	0.72	8
PINCH	1.000	0.125	0.88	8
TAP	0.375	0.875	0.20	5
DENY	0.812	0.187	0.68	6
KNOB	0.937	0.250	0.75	8
EXPAND	1.000	0.687	0.59	7.2
Total	0.906	0.347	0.74	-

Table 6: *Per-class evaluation of the scores of the SHREC2021 contest obtained with the current STRONGER recognizer architecture.*

tio, i.e. the ratio between the number of predictions not corresponding to ground truth ones divided by the total number of gestures of the corresponding class in the sequences, and the Jaccard Index (JI) [WZZ*16,ZCCL18] measuring the average relative overlap between the ground truth and the predicted label sequences:

$$JI_{s,i} = \frac{GT_{s,i} \cap P_{s,i}}{GT_{s,i} \cup P_{s,i}} \quad (3)$$

Table 4 shows the scores obtained by our algorithm on the SHREC'21 benchmark, compared with the best runs of the participants. Our method performs well, but the false positives are still a bit high compared to the results of the contest winners. However, it must be noted that this group used a pre-segmentation module based on gesture energy to reduce false positives and this should be the reason for this difference. We plan to add a similar module in the future versions of STRONGER.

It is interesting to see how the addition of the JPD and palm orientation features as well as the training of the Window Threshold Array improves the results with respect to a simple application of a sliding-window DDNet.

Table 5 shows the results of an ablation study demonstrating the effectiveness of the added orientation-related features to improve the detection rate of the gestures and the effectiveness of the threshold training to reduce the False Positives rate.

Table 6 shows the SHREC'21 scores obtained with STRONGER for each gesture class together with the classifier thresholds learned from the training set, i.e. the values in the WTA. It is possible to see that results are very good for many classes, while just a few gesture classes (Circle, Cross, Tap) are responsible for the decrease of the detection rate and the increase of the False Positive ratio. This means that just excluding these few classes, our system would be quite effective.

Bar charts in Figure 7 demonstrate that the addition of the two novel features to the network input results in an increase of the detection rate and of the Jaccard Index and a reduction of false positives, with the only exception of the tap gesture, that is not well handled by our system.

4.4. System prototype

Our recognizer can directly process gesture stream and has been used as the server-side application connected to the Hololens app described in Section 3.4.

In the demo setting the multiple windows, classification is performed every 5 frames of the continuous stream (45 fps).

The prediction results are sent back to the Hololens app and are then printed on the screen on the user's display. The classification time is 0.1 seconds.

An example of the online recognition system can be seen in the video at the link https://streamable.com/5q6pl1.

Considering that no post-processing is applied the results are sufficiently good, eve for some classes where the SHREC'21 testing was not optimal, such as "TAP", "PINCH", "EXPAND" and "DENY". Some gestures of the SHREC 2021 dictionary are not, however, well handles, especially dynamic coarse.

We believe that the system is almost ready for practical applications, as it is possible to restrict the useful dictionary only to well-recognized gestures. Furthermore, we now plan to acquire specific training sets with the Hololens not relying on external benchmarks acquired with different devices, and greatly increasing the number of training examples.

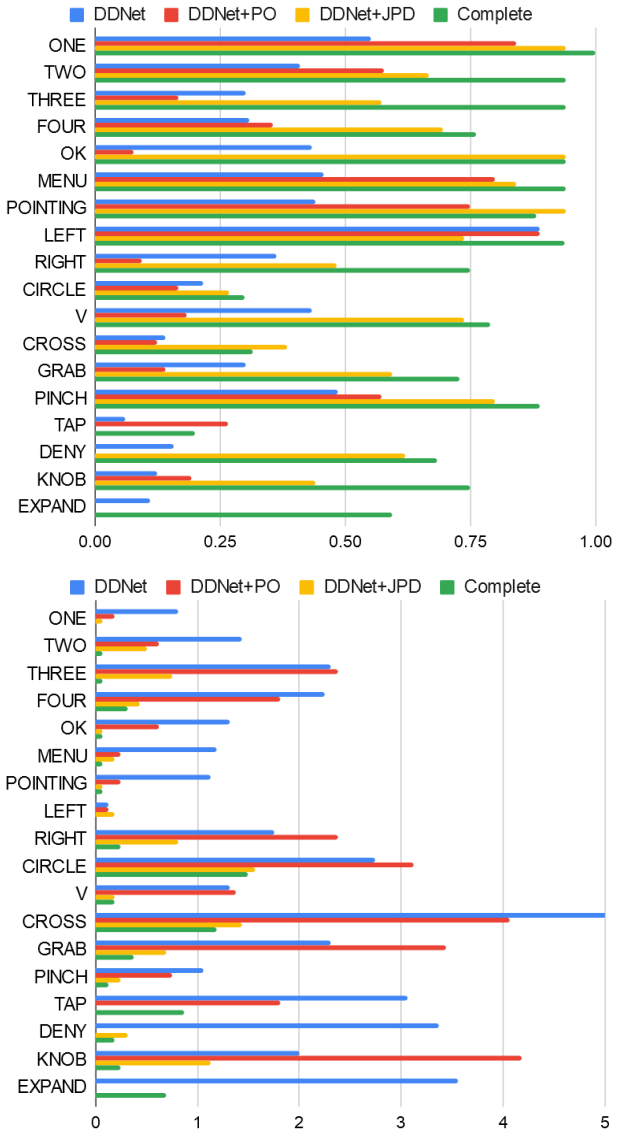

Figure 7: *Both the features added to the original DDNet contribute to improve the SHREC '21 benchmark scores. Top: Jaccard Index per class for the original DDNet, DDNet+PO, DDNet+JPD and both PO+JPD. Middle row: Bottom: Per class false positive ratio for the original DDNet, DDNet+PO, DDNet+JPD and both PO+JPD.*

5. Discussion

We presented STRONGER, an online gesture recognition system for mixed reality interaction, implemented with a client-server architecture on an Hololens 2 device and based on a modified DDNet architecture used with a sliding window approach and specialized training to reduce false positives based on the selection of optimal classifiers' thresholds. Results demonstrate that the proposed modification to the 1-D convolutional network approach is able to improve the classification performances and that the development of mid-air interfaces based on dictionaries of heterogeneous gestures more complex than those currently used in mixed reality apps are feasible.

We plan to improve STRONGER by adding heuristics for pre-segmentation of gestures as proposed by the group obtaining the best results in the SHREC'21 contest [CGS*21], to capture novel training sets directly with the Hololens 2 setup and optimize the code to improve the usability of the system.

The idea of adding specific classifiers for selected gesture classes could be also extended by actually training separate networks for single classes or subsets of gestures. Different networks could be trained for different windows sizes or to recognize different gesture types (e.g. static, dynamic coarse, and fine). The output of the different classifiers should then be combined in the final algorithm.

Acknowledgments This work was partially supported by the MIUR Excellence Departments project 2018-2022.

References

[ABC*18] AVOLA D., BERNARDI M., CINQUE L., FORESTI G. L., MASSARONI C.: Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. *IEEE Transactions on Multimedia 21*, 1 (2018), 234–245. 2

[CBP*19] CAPUTO F. M., BURATO S., PAVAN G., VOILLEMIN T., WANNOUS H., VANDEBORRE J.-P., MAGHOUMI M., TARANTA E., RAZMJOO A., LAVIOLA JR J., ET AL.: Online gesture recognition. In *Eurographics Workshop on 3D Object Retrieval* (2019), The Eurographics Association. 2

[CGG*20] CAPUTO A., GIACHETTI A., GIANNINI F., LUPINETTI K., MONTI M., PEGORARO M., RANIERI A.: Sfinge 3d: A novel benchmark for online detection and recognition of heterogeneous hand gestures from 3d fingers' trajectories. *Computers & Graphics 91* (2020), 232–242. 2

[CGS*21] CAPUTO A., GIACHETTI A., SOSO S., PINTANI D., D'EUSANIO A., PINI S., BORGHI G., SIMONI A., VEZZANI R., CUCCHIARA R., RANIERI A., GIANNINI F., LUPINETTI K., MONTI M., MAGHOUMI M., AU2 J. J. L. J., LE M.-Q., NGUYEN H.-D., TRAN M.-T.: Shrec 2021: Track on skeleton-based hand gesture recognition in the wild, 2021. arXiv:2106.10980. 2, 4, 5, 7, 8

[CvMG*14] CHO K., VAN MERRIËNBOER B., GULCEHRE C., BAHDANAU D., BOUGARES F., SCHWENK H., BENGIO Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, Oct. 2014), Association for Computational Linguistics, pp. 1724–1734. URL: https://www.aclweb.org/anthology/D14-1179, doi:10.3115/v1/D14-1179. 2

[CZP*19] CHEN Y., ZHAO L., PENG X., YUAN J., METAXAS D. N.: Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. *BMVC* (2019). 2

[DSWV16] DE SMEDT Q., WANNOUS H., VANDEBORRE J.-P.: Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2016). 2

[DSWV*17] DE SMEDT Q., WANNOUS H., VANDEBORRE J.-P., GUERRY J., LE SAUX B., FILLIAT D.: Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset. In *3DOR-10th Eurographics Workshop on 3D Object Retrieval* (2017), pp. 1–6. 1, 2, 3, 6

[DXMY18] DEVINEAU G., XI W., MOUTARDE F., YANG J.: Convolutional Neural Networks for Multivariate Time Series Classification using both Inter- and Intra- Channel Parallel Convolutions. In *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP'2018)* (Marne la Vallée, France, June 2018). URL: https://hal-mines-paristech.archives-ouvertes.fr/hal-01888862. 6

[GHYBK18] GARCIA-HERNANDO G., YUAN S., BAEK S., KIM T.-K.: First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 2

[GHZ*21] GUO F., HE Z., ZHANG S., ZHAO X., FANG J., TAN J.: Normalized edge convolutional networks for skeleton-based hand gesture recognition. *Pattern Recognition* (2021), 108044. 2

[GLY21] GUO L., LU Z., YAO L.: Human-machine interaction sensing technology based on hand gesture recognition: A review. *IEEE Transactions on Human-Machine Systems* (2021). 1

[HWC*18] HOU J., WANG G., CHEN X., XUE J.-H., ZHU R., YANG H.: Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops* (2018), pp. 0–0. 2

[HWC*19] HOU J., WANG G., CHEN X., XUE J.-H., ZHU R., YANG H.: Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition. In *Computer Vision – ECCV 2018 Workshops* (Cham, 2019), Leal-Taixé L., Roth S., (Eds.), Springer International Publishing, pp. 273–286. 6

[LHY*19] LI Y., HE Z., YE X., HE Z., HAN K.: Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition. *EURASIP Journal on Image and Video Processing 2019*, 1 (2019), 1–7. 2

[LLG*21] LI C., LI S., GAO Y., ZHANG X., LI W.: A two-stream neural network for pose-based hand gesture recognition. *arXiv preprint arXiv:2101.08926* (2021). 2

[MDZ16] MARIN G., DOMINIO F., ZANUTTIGH P.: Hand gesture recognition with jointly calibrated leap motion and depth sensor. *Multimedia Tools and Applications 75*, 22 (2016), 14991–15015. 2

[MLJ18] MAGHOUMI M., LAVIOLA JR J. J.: Deepgru: Deep gesture recognition utility. *arXiv preprint arXiv:1810.12514* (2018). 2

[MZW*20] MA C., ZHANG S., WANG A., QI Y., CHEN G.: Skeleton-based dynamic hand gesture recognition using an enhanced network with one-shot learning. *Applied Sciences 10*, 11 (2020). URL: https://www.mdpi.com/2076-3417/10/11/3680, doi:10.3390/app10113680. 6

[NBLB21] NGUYEN X. S., BRUN L., LÉZORAY O., BOUGLEUX S.: Learning recurrent high-order statistics for skeleton-based hand gesture recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)* (2021), IEEE, pp. 975–982. 2

[NCP*18] NEZ J. C., CABIDO R., PANTRIGO J. J., MONTEMAYOR A. S., VLEZ J. F.: Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recogn. 76*, C (Apr. 2018), 80–94. URL: https://doi.org/10.1016/j.patcog.2017.10.033, doi:10.1016/j.patcog.2017.10.033. 6

[UBG*20] UNGUREANU D., BOGO F., GALLIANI S., SAMA P., DUAN X., MEEKHOF C., STÜHMER J., CASHMAN T. J., TEKIN B., SCHÖNBERGER J. L., ET AL.: Hololens 2 research mode as a tool for computer vision research. *arXiv preprint arXiv:2008.11239* (2020). 4

[WZZ*16] WAN J., ZHAO Y., ZHOU S., GUYON I., ESCALERA S., LI S. Z.: Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2016), pp. 56–64. 7

[XWG*19] XINGHAO C., WANG G., GUO H., ZHANG C., WANG H., ZHANG L.: Mfa-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data. *Sensors 19* (01 2019), 239. doi:10.3390/s19020239. 6

[YSWN19] YANG F., SAKTI S., WU Y., NAKAMURA S.: Make skeleton-based action recognition model smaller, faster and better. In *ACM International Conference on Multimedia in Asia* (2019). 2, 3, 4, 6

[ZBV*20] ZHANG F., BAZAREVSKY V., VAKUNOV A., TKACHENKA A., SUNG G., CHANG C.-L., GRUNDMANN M.: Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214* (2020). 1

[ZCCL18] ZHANG Y., CAO C., CHENG J., LU H.: Egogesture: A new dataset and benchmark for egocentric hand gesture recognition. *IEEE Transactions on Multimedia 20*, 5 (2018), 1038–1050. 7

[ZLX17] ZHANG S., LIU X., XIAO J.: On geometric features for skeleton-based action recognition using multilayer lstm networks. doi:10.1109/WACV.2017.24. 6