Two-Model-Based Online Hand Gesture Recognition from Skeleton Data

Zorana Doždor[a], Tomislav Hrkać[b] and Zoran Kalafatić[c]
University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia

Keywords: Recurrent Neural Network, Gated Recurrent Unit, Online Gesture Recognition, Hand Skeleton, Sliding Window.

Abstract: Hand gesture recognition from skeleton data has recently gained popularity due to the broad areas of application and availability of adequate input devices. However, before utilising this technology in real-world conditions there are still many challenges left to overcome. A major challenge is robust gesture localization – estimating the beginning and the end of a gesture in online conditions. We propose an online gesture detection system based on two models – one for gesture localization and the other for gesture classification. This approach is tested and compared against the one-model approach, often found in literature. The system is evaluated on the recent SHREC challenge which offers datasets for online gesture detection. Results show the benefits of distributing the tasks of localization and recognition instead of using one model for both tasks. The proposed system obtains state-of-the-art results on SHREC gesture detection dataset.

1 INTRODUCTION

In recent years, hand gesture recognition has become an important part of human-computer interaction with a growing field of application, such as the video-game industry, medicine, sign language translation, automotive industry etc. Advancement of input devices has allowed scientists to develop methods for different input modalities, such as RGB, depth and hand skeleton data as well as different multimodal approaches.

Hand keypoints are easily obtained from sensors available in modern virtual reality devices (e.g. Leap Motion Controller, HoloLens 2), or alternatively using standard RGB input processed by a lightweight convolutional model such as MediaPipe Hands (Zhang et al., 2020). This allows for a variety of interaction mechanisms in virtual and mixed reality, making us rethink standard human-computer interaction. This inspired us to create an online gesture recognition system based solely on 3D hand skeleton coordinates.

The online gesture recognition (or detection) includes both the temporal localization of a gesture and its subsequent classification. However, up to this point a majority of research and datasets were focused on classification of already segmented gestures. Although this is an important part of online gesture recognition, it is less challenging than gesture extraction and recognition from continuous input data. Consequently, developed online systems were often application and/or dataset specific, and therefore not robust enough for general real world use.

There are several challenges regarding online gesture recognition in the real world systems. The system latency should be very low to give a feeling of an instant response. Also, the system should not have too many false positives; however, it should have a high detection rate, which often becomes a trade-off. Finally, the models should often be lightweight to be appropriate for real time use on embedded hardware.

A crucial part of online gesture recognition is gesture localization – estimating the beginning and end of each gesture. There are two possible approaches to this problem. In one approach, continuous input data is fed to the model by a (temporal) sliding window technique where the model labels each incoming input frame with one (or none) of the known gesture classes. Final gesture boundaries are then determined by different postprocessing strategies (Maghoumi and LaViola, 2019), (Emporio et al., 2021). Another approach first determines gesture boundaries (using often complex and time latent heuristics (Caputo et al., 2022), (Caputo et al., 2021)) and then classifies the segmented data into gesture classes.

We believe the latter is a more prudent approach because gestures lengths vary drastically, both within and between classes, which the former approach

[a] https://orcid.org/0000-0002-9194-4589
[b] https://orcid.org/0000-0002-4362-2489
[c] https://orcid.org/0000-0001-8918-9070

Doždor, Z., Hrkać, T. and Kalafatić, Z.
Two-Model-Based Online Hand Gesture Recognition from Skeleton Data.
DOI: 10.5220/0011663200003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP*, pages 838-845
ISBN: 978-989-758-634-7; ISSN: 2184-4321

struggles to handle. In this work, we propose a similar system; however, instead of using a complex heuristic for gesture localization, we propose training a recurrent model for this task. This results in a system comprised of two recurrent lightweight models – one for gesture localization and another for gesture classification.

Our contribution is as follows:

- a novel hand gesture recognition approach that uses two lightweight, easy to implement models;
- a hand gesture localization model based on trained recurrent network instead of using complex heuristics.

The proposed system achieves state-of-the-art results on SHREC gesture detection dataset.

2 RELATED WORK

Much of the published work in gesture recognition utilizes traditional computer vision methods where features are hand-crafted. However, deep learning methods have recently come to the forefront.

Gesture recognition methods differ based on the utilized input modalities, as they indicate the shape of the data upon which the algorithms are based. Out of the several prominent input modalities (RGB, depth, skeleton, segmentation mask), our work focuses on gesture recognition from hand skeleton data.

As for gesture localization, there has not been much research or datasets for temporal gesture localization from continuous skeleton data – something which this work aims to build upon.

2.1 Skeleton-Based Hand Gesture Classification

In skeleton-based hand gesture classification, the input rarely consists of just raw 3D hand skeleton points (Maghoumi and LaViola, 2019). Defining additional features based on 3D points can significantly improve model performance. Various features extracted from the hand-skeleton have been proposed, based on joint velocity, acceleration, joint-to-joint distances, angles between selected joints, etc.

Recurrent networks are one of the dominant approaches in gesture recognition because of their ability to extract important temporal information from sequential data. Still, they underperform when it comes to extracting spatial features. To aid this, convolutional neural networks (CNNs) are often utilised.

In (Maghoumi and LaViola, 2019), a deep recurrent network with an attention module is proposed for skeleton-data gesture recognition. The main building blocks of the model are stacked gated recurrent units (GRUs). (Shin and Kim, 2020) also use GRU architecture, but they divide the input features into multiple parts to reduce the number of network parameters needed for optimization. In (Song et al., 2017), a recurrent network is constructed with LSTM units with spatial and temporal attention subnetworks. (Chen et al., 2017) also utilizes an LSTM network, but in addition to skeleton sequence input they construct a set of finger and global motion features to describe hand movement and improve classification accuracy.

As for the CNN, a simple convolutional network inspired by DDNet architecture (Yang et al., 2019) is proposed in (Emporio et al., 2021). Instead of using raw data, the authors constructed five different input features from the input skeleton: the Euclidian distances between pairs of joints, the palm orientation, the orientations of selected joint pairs and the joint speeds. In (Devineau et al., 2018) a CNN with parallel branches for feature extraction is presented. In (Hou et al., 2019), an end-to-end attention residual temporal convolutional network is proposed, tracking both spatial and temporal features.

A combination of CNN and LSTM is presented in (Núñez et al., 2018), with CNN being used as encoder part of the network, while LSTM is used for gesture classification. For that approach, a two-stage training strategy is presented, firstly adjusting the weights of the CNN, then training the combination of CNN and LSTM.

Additionally, there have been some attempts in creating an image representation of skeleton trajectories. In (Lupinetti et al., 2020) and (Caputo et al., 2020), color intensity is used to represent temporal information, and the final 2D image is created by projecting the 3D points onto a view plane. Images are then classified by a CNN. Similarly, in (Wang et al., 2016), hue, saturation, and brightness are used to represent spatial-temporal motion information. Joint trajectory maps are constructed for three orthogonal planes and processed by a CNN, then fused for the final result.

2.2 Temporal Gesture Localization

There has been a lot of work done on temporal localization in videos. However, localization for skeleton-based continuous data is yet to be researched. To the authors knowledge, the only non-segmented skeleton-based hand gesture recognition datasets have been proposed by Caputo et al. as a part of the SHape Retrieval Contest (SHREC).

The SHREC 2019 (Caputo et al., 2019) dataset contains only 5 dynamic gestures described by a hand trajectory. In SHREC 2021 (Caputo et al., 2021), there were 18 gesture classes divided into static, dynamic and fine dynamic gestures. Finally, the third dataset, SHREC 2022 (Caputo et al., 2022) was created to alleviate some of the problems of the previous datasets. These challenges resulted in several methods for gesture localization and recognition proposed by the contestants.

The results of four research groups participating in SHREC 2021 contest are presented in (Caputo et al., 2021). One group proposed a transformer-based architecture for gesture classification. The model makes predictions for every time step of the input and then a finite state machine is utilised for localisation of the gestures. Another group proposed an energy-based detection module for gesture localisation. With a sliding window approach, they calculated the energy for several consecutive windows and selected the ones with the minimum of energy as candidate gestures. Similar work was done in SHREC 2022, where one group applied an energy-based proposal module with the addition of a gesture localisation branch in the classification model for more precise localisation. Another group presented a method based on temporal convolutional network (TCN), where the model is fed with windows of length *n*, and the final per-frame prediction is made by a voting strategy with *n* votes for each frame. An additional post-processing strategy is applied to determine the gesture start and end from the per-frame predictions.

3 PROPOSED SYSTEM

Figure 1 shows the general flow of the proposed two-model system. The first model (gesture localizer) is a binary classifier, predicting whether a gesture is happening, for every time step of input data. The input is constructed by a temporal sliding window, where the classifier makes its prediction for the last time step of the input window. The model outputs a binary prediction: 1 if a gesture is detected, and 0 otherwise.

Based on the sequence of predictions from the localizer, segments which contain gestures are extracted from the data, resampled to a fixed length, and provided as input to the second model (gesture recognizer). The second model then classifies the segments into one of the known gesture classes, or the non-gesture class. The purpose of the non-gesture class is to filter out any gesture segments that do not actually contain a gesture, i.e. the false positives of the localizer.

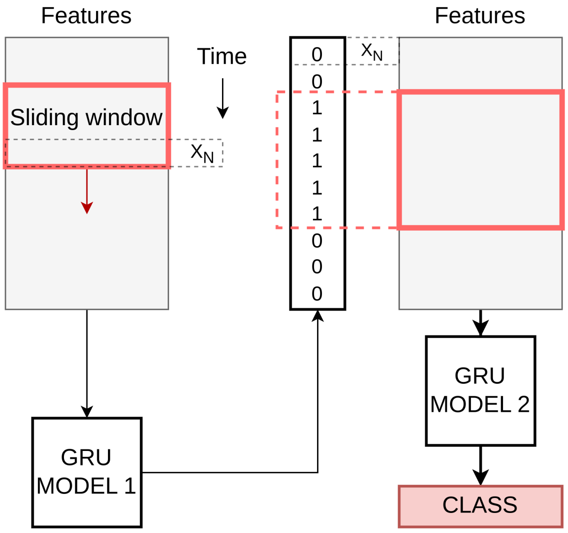


Figure 1: General flow of the system. Using a sliding window approach, the gesture localization model (GRU model 1) extracts gesture segments, while the gesture classification model (GRU model 2) classifies them into gesture classes.

3.1 Input Preprocessing

The input for both models is preprocessed as follows: first, for each input sample, the per-axis coordinates are normalized to have zero mean and unit variance; then, additional features are derived from the coordinates and added to the input – joint velocity and pairwise Manhattan distance. The final feature vector for each time step is obtained by flattening and concatenating the aforementioned features.

Velocity is important as different gestures and different parts of the same gesture are made with varying speeds. For time step *t* and joint *i*, per-axis velocity is calculated as the first derivative of the joint's positions approximated by finite differences:

$$v_{i,x} = x_{i,t} - x_{i,t-1}, \tag{1a}$$
$$v_{i,y} = y_{i,t} - y_{i,t-1}, \tag{1b}$$
$$v_{i,z} = z_{i,t} - z_{i,t-1}. \tag{1c}$$

Pairwise Manhattan distance is introduced to add spatial information. We chose Manhattan distance instead of the more popular Euclidian because experimental results have shown better performance. For joint pair *i* and *j*, Manhattan distance for time step *t* is calculated as follows:

$$d_{i,j,t} = |x_{i,t} - x_{j,t}| + |y_{i,t} - y_{j,t}| + |z_{i,t} - z_{j,t}|. \tag{2}$$

3.2 Architectures of the Models

Figure 2 shows the architectures of the recurrent models used. The encoder part of both architectures is essentially the same: it consists of a fully connected layer for reduction of feature dimensionality, followed by two gated recurrent unit (GRU) layers with hyperbolic tangent activation. The only differences are that in the second model batch normalization is added after the fully connected layer, and dropout with the probability of 0.2 is applied after each GRU layer, to reduce overfitting. Inspired by (Maghoumi and LaViola, 2019) we selected GRU as a building block of the encoder because it has less training parameters than LSTM, and still performs well for sequences that are not very long. We found experimentally that stacking just two GRU layers is enough for our application.

For time step t, based on the input vector x_t and the previous hidden state h_{t-1}, the hidden output for the GRU unit is calculated as follows:

$$u_t = \sigma(W_{uxh}x_t + W_{uhh}h_{(t-1)} + b_{uh}), \tag{3}$$

$$r_t = \sigma(W_{rxh}x_t + W_{rhh}h_{(t-1)} + b_{rh}), \tag{4}$$

$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{(t-1)}) + b_h), \tag{5}$$

$$h = u_t \odot h_{(t-1)} + (1 - u_t) \odot \tilde{h}_t, \tag{6}$$

where u and r are update and reset gates, respectively, while $\tilde{h}$ and h represent intermediate hidden state and output hidden state, respectively. Weights and biases are denoted as W and b, while σ denotes the sigmoid function. Symbol $\odot$ represents the element-wise product, know as the Hadamard product.

The classification part for the first model consists of only one fully connected layer with a sigmoid activation function. The second model has three fully connected layers: two with ReLu activation, and the last one with softmax activation, where the number of units is equal to the number of classes, plus one for the non-gesture class. The hyperparameters of the models are chosen experimentally.

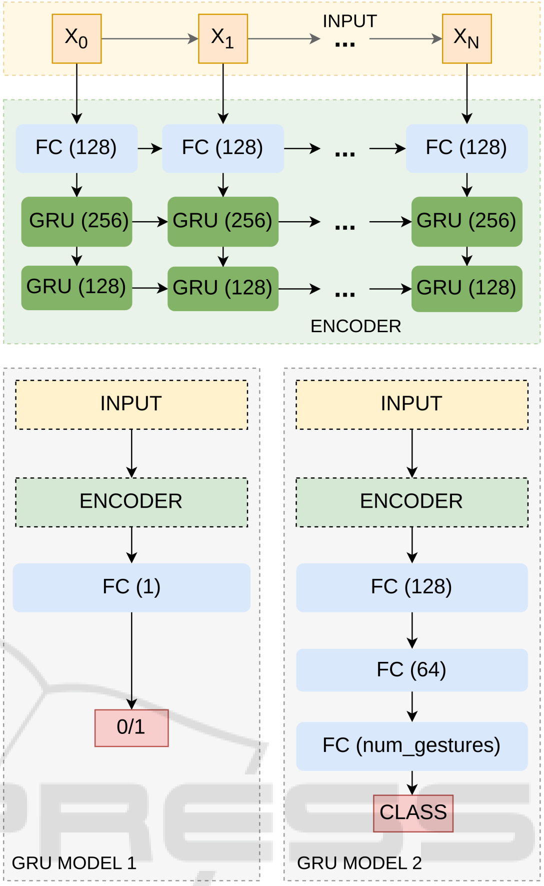


Figure 2: Architecture of the gesture localization model (GRU model 1) and the gesture classification model (GRU model 2). The models share the same encoder structure while they differ in the classifier architecture.

4 EXPERIMENTS

The proposed system is trained and evaluated on two similar datasets: the SHREC 2021 and SHREC 2022 online gesture detection benchmarks.

SHREC 2021 dataset consist of 16 gestures divided into static *(one, two, three, four, ok, menu)*, dynamic *(left, right, circle, v, cross)* and fine-grained dynamic *(grab, pinch, tap, knob, expand)* gestures. Hand skeleton data consisting of 20 joints was collected by a Leap Motion sensor at 50 fps. Total of 180 sequences is divided into train and test, with 108 and 72 sequences respectively. The sequences contain variable number of gestures (3-5), and the gestures are interleaved with other hand movements. The beginning and the end of each gesture is annotated, as well as the gesture class.

SHREC 2022 consists of 16 gestures divided into four classes: static *(one, two, three, four, ok, menu)*, dynamic *(left, right, circle, v, cross)*, fine-grained dynamic *(grab, pinch)*, and dynamic periodic gestures *(deny, wave, knob)*, as illustrated in Figure 3. The dataset has a total of 288 sequences divided evenly into training and test set. For each time step the coordinates (x,y,z) of 26 hand joints are captured by a HoloLens 2 device. The gesture dictionary is slightly changed compared to SHREC 2021 to avoid some ambiguities that were noticed.

The evaluation considers several measures: the detection rate, the number of false positives, the detection latency, and the Jaccard Index.

- *Detection rate (DR)*: A gesture is considered correctly detected if the overlap between the ground truth gesture temporal boundaries and the predic-

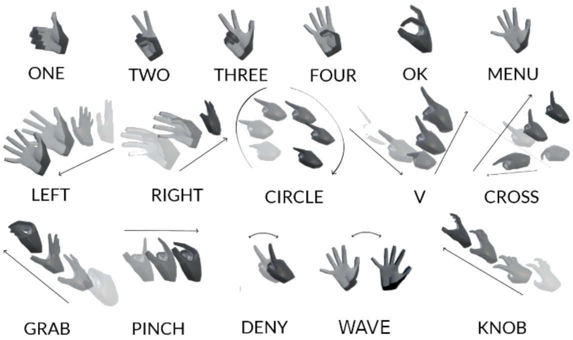


Figure 3: SHREC 2022 gestures (first row - static, second row - dynamic coarse, third row left - dynamic fine, third row right - periodic)(Caputo et al., 2022).

tion boundaries is larger than a predefined threshold (usually 50%), and the predicted class label is correct. The detection rate is the percentage of the ground truth gestures (of some class) that are correctly detected and classified.

- *False positive rate (FP)*: False positives include segments detected as gestures, not overlapping with any of the ground truth gestures, as well as the misclassified gestures. The false positive rate is the ratio between the number of false positives and the number of gestures in the test set.
- *Detection latency (DL)*: expresses the difference between the predicted gesture start and the last time step used for that prediction.
- *Jaccard Index (JI)*: measures the relative overlap between the ground truth and the predicted gestures.

The Jaccard Index is given by the expression:

$$JI_{s,i} = \frac{GT_{s,i} \cap P_{s,i}}{GT_{s,i} \cup P_{s,i}}, \tag{7}$$

where $GT_{s,i}$ and $P_{s,i}$ are ground truth and prediction binary vectors for sequence s. Vector components assume values one or zero, based on the timesteps where the i-th gesture is being performed.

We used both SHREC'21 and SHREC'22 datasets to test the benefits of our two-model approach. Additionally, we compared the obtained results with the published state-of-the-art results for the SHREC'22 dataset (Caputo et al., 2022). We omit the comparison with the SHREC'21 results from (Caputo et al., 2021) due to the recent modification of the labels.

4.1 Training

Models were trained with the Adam optimizer and a learning rate of 0.001 for 200 epochs. Reduction of learning rate on plateau is applied with patience of 20 epochs. First model is optimised with the binary cross entropy loss, while for the second model the categorical cross entropy is used.

Training data for both models was split into training and validation in the ratio of 80:20 by a stratified split. The length of sliding window for the first model is set to 40. The positive data for the second model are the extracted gesture segments, while negative examples are randomly sampled with a variable length from the non-gesture segments of the sequences. All extracted segments are resampled to a length of 20.

Data augmentation was applied for the training of both models. For each axis (x,y,z) of the input sample, with 10% probability, a random number between -0.1 and 0.1 is added to the joint coordinates in a randomly chosen segment of the sequence. The length of the segment is determined randomly, ranging from 0 to 25% of the input length. The position of the modified segment in the input sequence is also selected randomly.

4.2 Online Gesture Detection

When using the proposed system in an online setting, the boundaries of the candidate gestures sent to the second model are determined as follows: the beginning of the gesture is detected if 5 consecutive time steps are labelled as a gesture, while the end of the gesture is determined when 10 consecutive time steps are labelled as a non-gesture. That means, the system has a delay of 10 time steps between the end of the gesture and making the final gesture classification. Detection latency is 5 time steps, because decision about the start time step is made after 5 consecutive gesture predictions. The output of the first model, p, is a number between 0 and 1 because of the sigmoid activation function. We select time steps with $p > 0.5$ as frames that contain a gesture.

A single processing step for the first model includes sliding window data preprocessing and classification, while for the second model it consists of extracted gesture segment resampling and classification. For both models the measured execution time for a single processing step is approximately 20 ms. The experiments were conducted on a single GPU, *Nvidia GeForce RTX 2060 SUPER (8GB)* with an *Intel Core i9 (8 cores)* CPU. The system can process about 50 fps and, therefore, is appropriate for real-time applications.

Table 1: Comparison of one-model and two-model approaches on SHREC'21 and SHREC'22 gesture recognition challenges.

Dataset	Method	DR	FP	JI
SHREC'21	One model	0.7279	0.1728	0.6570
SHREC'21	Two models	**0.7831**	**0.0919**	**0.7371**
SHREC'22	One model	0.8524	0.1528	0.7519
SHREC'22	Two models	**0.8542**	**0.0920**	**0.7881**

Table 2: Comparison of our results with state-of-the-art on SHREC'22 gesture recognition challenge.

Method	DR	FP	JI	DL (fr)
Stronger	0.7188	0.3299	0.5915	14.79
2ST-GCN 5F	0.7378	0.1042	0.6720	13.28
Causal TCN	0.8003	0.2552	0.6845	19.00
TN-FSM+JD	0.7708	0.1823	0.6582	10.00
Ours	**0.8542**	**0.0920**	**0.7881**	**5.00**

4.3 One-Model and Two-Model Approach Comparison

To validate the benefits of our two-model approach, an experiment has been conducted comparing a typical one-model approach to our own.

First, we trained only the gesture classification model from our system (GRU model 2) to make predictions based on sliding window input instead of the already extracted gestures. The model predicts one of the gesture classes (or a non-gesture), for every time step. All of the post-processing and training strategies are the same as explained above.

Then, we trained the proposed system, where the first model predicts gesture segments based on sliding window input, and the second model classifies the extracted segments. We trained and tested both approaches on SHREC'22 and SHREC'21 datasets. The detection rate, the number of false positives and the Jaccard Index are shown in Table 1 and are calculated as a mean of per class results.

For the SHREC'22 dataset the detection rate is approximately the same for both approaches, while the Jaccard Index is better for the two-model approach. The number of false positives is lower for the two-model approach by 6%. We also compared the number of false positives across all examples. For the two-model approach the total number of false positives is 9%: 3% are misclassified examples, and 6% are real false positives. For the one-model approach the total number of false positives is 16%: 4% are misclassified examples, and 12% are real false positives. This suggests that the two-model approach has 50% less false positives.

While in the SHREC'22 the results are mostly similar, for the SHREC'21 dataset the two-model approach shows better results across all measures. We suppose that this is due to the fact that SHREC'21 dataset labels for start and end of gestures are noisier, and gestures differ more in length so that the generalization properties of the two-model approach are more pronounced. Figure 4 shows the detection rate (which includes both correct localization and classification) in relation to the chosen threshold on overlap ratio between the predictions and the ground truth for both approaches on SHREC'21. The two-model approach is consistently better. The graph shows that both approaches have much higher detection rate for lower overlap ratio values. This indicates that gesture classification is usually successful even with poor localization.

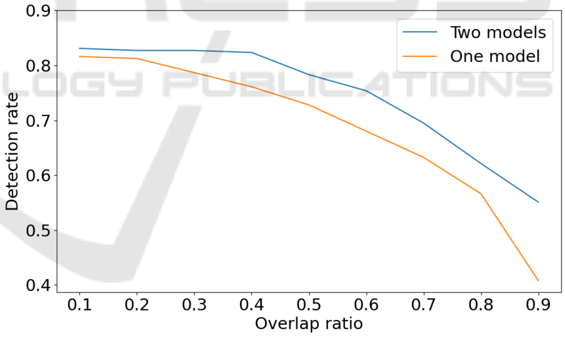
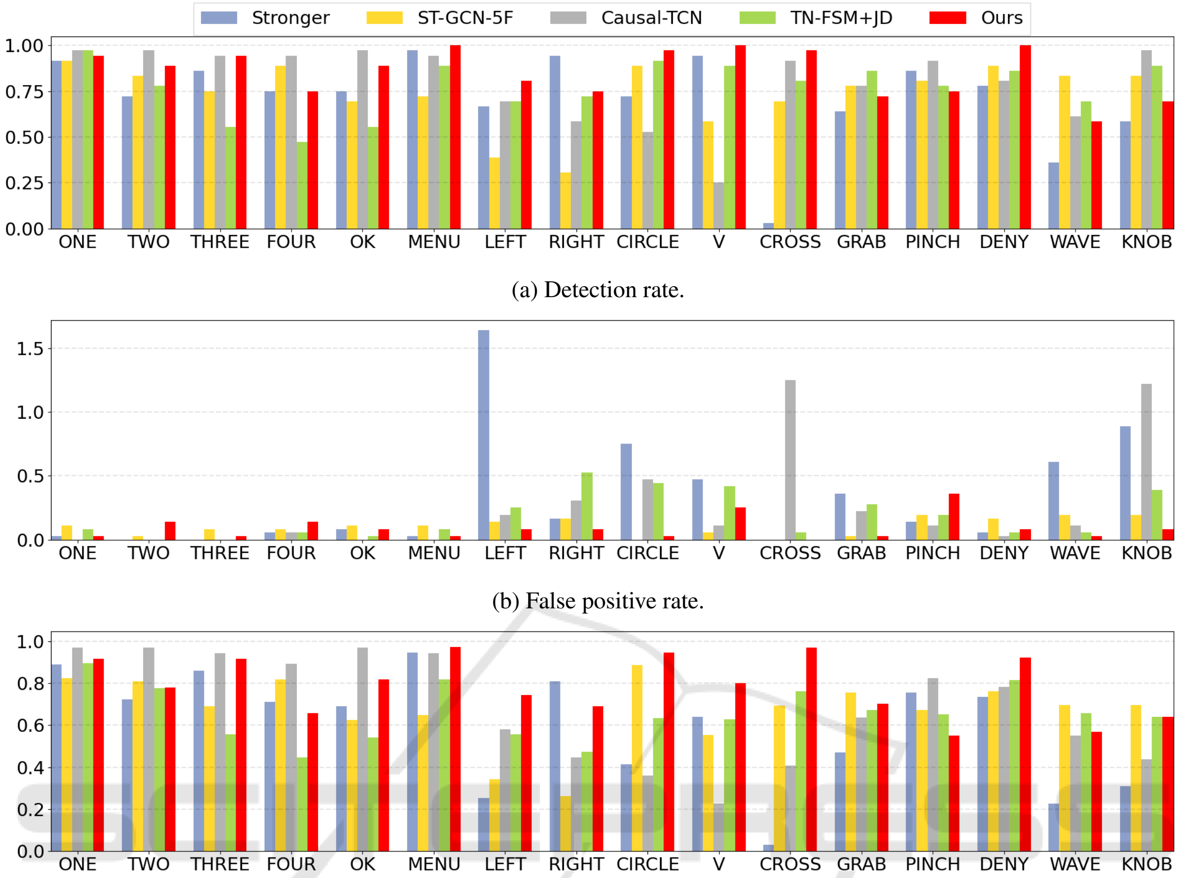


Figure 4: Detection rate in relation to overlap ratio threshold on SHREC'21 test set for one-model and two-model approach.

4.4 Comparison with SHREC 2022 Challenge State-of-the-Art Results

Our results are compared to the best runs of the SHREC'22 challenge participants. Table 2 shows the results. Participant results reported here differ a little from (Caputo et al., 2022) because the evaluation script has been fixed by the contestant organisers since the publishing of the results. Our system has the highest detection rate and Jaccard Index, while the number of false positives is slightly lower than the



(a) Detection rate.

(b) False positive rate.

(c) Jaccard Index.

Figure 5: Comparison of per class (a) detection rate (b) false positive rate and (c) Jaccard Index with state-of-the-art of SHREC 2022 gesture recognition challenge.

best result in the contest.

Figure 5 shows per class detection rates, false positive rates, and Jaccard Indexes compared to the challenge contestants. As visible in figure 5a, we have the best detection rate results for most of the gestures in the dynamic category (*left, circle, v, cross*), and comparable results for the rest of the categories. Other methods usually have inconsistent results across gesture categories. For instance, *Causal-TCN* is the best in the static category (*one, two, three, four, ok, menu*), but their performance in dynamic category is significantly lower. For our system, these differences are far less emphasized.

As for the Jaccard Index in 5c, it is also the best for dynamic category and comparable with the best results for other categories, which is to be expected as it is correlated with the detection rate.

Lastly, figure 5b shows the number of false positives. It can be observed that out model has consistently low false positive rate across all gesture classes. This is important because some of the other methods, like *Stronger* or *Causal-TCN*, have spikes with high number of false positives for certain gesture classes. These spikes are in some cases even larger than one, meaning that the number of false positives exceeds the number of samples for that class.

5 CONCLUSION

In this work we proposed a method for gesture localization and recognition from hand skeleton data in an online environment. The presented results demonstrate the benefits of distributing the tasks of gesture localization and recognition between two models, rather than training just one model for both. The two-model approach reduces the number of false positives and can improve generalization when gestures are considerably diverse in length, or when the dataset labels are noisy.

Although our results are better than those of the SHREC 2022 challenge contestants, they should be further improved for real-word use. With the detection rate of 85% and 9% of false positives, the system is still not robust enough. The number of false positives should be close to zero because every response activated by a false gesture detection would deteriorate the user experience. As shown, using two models slightly alleviates this problem, but further improvements are needed to enable smooth system usage.

There are several directions future work can take to further improve our results. Firstly, our models are trained independently. We believe they could benefit from end-to-end training. The accuracy of the second model is dependent on the output of the first model, however, higher accuracy of the first model alone does not necessarily lead to the better overall performance.

Secondly, the choice of hand-crafted features derived from hand skeleton has a large influence on model performance. We believe it should be further explored to fully utilize model capacity.

Lastly, we selected the models' parameters based on single stratified split. Although time consuming, it could prove beneficial to do grid search in combination with k-fold cross validation for the selection of parameters.

ACKNOWLEDGEMENTS

This work was supported by the ”Development of an advanced electric bicycles charging station for a smart city” project co-funded under the Operational Program from the European Structural and Investment Funds.

REFERENCES

Caputo, A., Emporio, M., Giachetti, A., Cristani, M., Borghi, G., D'Eusanio, A., Le, M.-Q., Nguyen, H.-D., Tran, M.-T., Ambellan, F., Hanik, M., Navayazdani, E., and von Tycowicz, C. (2022). SHREC 2022 track on online detection of heterogeneous gestures. *Computers & Graphics*, 107.

Caputo, A., Giachetti, A., Giannini, F., Lupinetti, K., Monti, M., Pegoraro, M., and Ranieri, A. (2020). SFINGE 3D: A novel benchmark for online detection and recognition of heterogeneous hand gestures from 3d fingers' trajectories. *Comput. Graph.*, 91:232–242.

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., Jr, J., Le, M.-Q., Nguyen, H.-D., and Tran, M.-T. (2021). SHREC 2021: Skeleton-based hand gesture recognition in the wild. *Computers & Graphics*, 99.

Caputo, F. M., Burato, S., Pavan, G., Voillemin, T., Wannous, H., Vandeborre, J.-P., Maghoumi, M., Taranta, E. M., A., Razmjoo, LaViola, J. J., Manganaro, F., Pini, S., Borghi, G., Vezzani, R., Cucchiara, R., Nguyen, H., Tran, M.-T., and Giachetti, A. (2019). SHREC 2019 track: Online gesture recognition. In *Eurographics Workshop on 3D Object Retrieval*.

Chen, X., Guo, H., Wang, G., and Zhang, L. (2017). Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. In *IEEE International Conference on Image Processing (ICIP)*, pages 2881–2885.

Devineau, G., Moutarde, F., Xi, W., and Yang, J. (2018). Deep learning for hand gesture recognition on skeletal data. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 106–113.

Emporio, M., Caputo, A., and Giachetti, A. (2021). STRONGER: Simple TRajectory-based ONline GEsture Recognizer. In *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association.

Hou, J., Wang, G., Chen, X., Xue, J.-H., Zhu, R., and Yang, H. (2019). Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition. In Leal-Taixé, L. and Roth, S., editors, *Computer Vision – ECCV 2018 Workshops*, pages 273–286, Cham. Springer International Publishing.

Lupinetti, K., Ranieri, A., Giannini, F., and Monti, M. (2020). 3d dynamic hand gestures recognition using the leap motion sensor and convolutional neural networks. In De Paolis, L. T. and Bourdot, P., editors, *Augmented Reality, Virtual Reality, and Computer Graphics*, pages 420–439, Cham. Springer International Publishing.

Maghoumi, M. and LaViola, J. J. (2019). DeepGRU: Deep gesture recognition utility. In *ISVC*.

Núñez, J. C., Cabido, R., Pantrigo, J. J., Montemayor, A. S., and Vélez, J. F. (2018). Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognit.*, 76:80–94.

Shin, S. and Kim, W.-Y. (2020). Skeleton-based dynamic hand gesture recognition using a part-based GRU-RNN for gesture-based interface. *IEEE Access*, 8:50236–50243.

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*.

Wang, P., Li, Z., Hou, Y., and Li, W. (2016). Action recognition based on joint trajectory maps using convolutional neural networks. *Proceedings of the 24th ACM international conference on Multimedia*.

Yang, F., Sakti, S., Wu, Y., and Nakamura, S. (2019). Make skeleton-based action recognition model smaller, faster and better. In *ACM International Conference on Multimedia in Asia*.

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C., and Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. *CoRR*, abs/2006.10214.