Universal Access in the Information Society
https://doi.org/10.1007/s10209-024-01139-6

LONG PAPER

Hand gesture recognition for user-defined textual inputs and gestures

Jindi Wang[1] · Ioannis Ivrissimtzis[1] · Zhaoxing Li[2] · Lei Shi[3]

Accepted: 26 July 2024

Abstract
Despite recent progress, hand gesture recognition, a highly regarded method of human computer interaction, still faces considerable challenges. In this paper, we address the problem of individual user style variation, which can significantly affect system performance. While previous work only supports the manual inclusion of customized hand gestures in the context of very specific application settings, here, an effective, adaptable graphical interface, supporting user-defined hand gestures is introduced. In our system, hand gestures are personalized by training a camera-based hand gesture recognition model for a particular user, using data just from that user. We employ a lightweight Multilayer Perceptron architecture based on contrastive learning, reducing the size of the data needed and the training timeframes compared to previous recognition models that require massive training datasets. Experimental results demonstrate rapid convergence and satisfactory accuracy of the recognition model, while a user study collects and analyses some initial user feedback on the system in deployment.

Keywords Human computer interaction · Hand gesture · User study · Personalization

1 Introduction

The use of palm and finger movements to convey thoughts and ideas is known as hand gesture [1]. Hand gestures allow users to communicate with a machine simply by moving their hands, eliminating the need for any additional input devices. They are becoming increasingly popular in a variety of Human-Computer Interaction (HCI) applications, benefiting from advances in both general computer technology and specialised hardware equipment [2, 3].

For a meaningful communication between human and machine, gestures of the human should be recognized by the machine in a process called hand gesture recognition (HGR). It is one of the central focal points of current research, with a wide range of application domains, including natural user interfaces [4], robot control [5], virtual gaming [6], and virtual reality [7]. Furthermore, systems for technology-assisted human communication, including sign language recognition, is another principal application of HGR [8].

Due to the variety of application domains of HGR, it is challenging to develop a standardised set of gestures that will become universally used and recognized. Instead, in practice, every user tends to develop their unique perspective on the posture of a given motion, and thus, ***how to convey to other systems or humans the intended meaning of a user's gesture, without having to teach them a new meaning for that gesture*** is one of the key challenges in the field [9–11]. User-defined hand gestures, with each user having their own interpreter, is a reasonable approach to the challenge, and the one we explore here is a fix for the current issue.

Several researchers have already studied processes by which users can possibly define a hand gesture vocabulary. Notably, Jahani et al. invited 20 non-technical users to participate in a study of the influence of hand gesture preferences of different users on hand gesture definitions [12]. Based on an experiment in which users described a simple chair and an abstract chair with their hand gestures, they found that in a virtual environment, users preferred to define a hand gesture vocabulary with two hands.

✉ Jindi Wang
jindi.wang@durham.ac.uk

✉ Ioannis Ivrissimtzis
ioannis.ivrissimtzis@durham.ac.uk

✉ Zhaoxing Li
zhaoxing.li@soton.ac.uk

✉ Lei Shi
lei.shi@newcastle.ac.uk

[1] Durham University, Durham, UK

[2] University of Southampton, Southampton, UK

[3] Newcastle University, Newcastle, UK

Published online: 02 August 2024

Piumsomboon et al. conducted research on hand gesture guessability in an Augmented Reality (AR) environment [13]. They invited 20 users to make 800 gestures corresponding to 40 tasks, and created user-defined gesture sets to guide the designers in implementing user-centered consistent hand gestures for AR. Because most surface computing prototypes use hand gestures created by system designers, their hand gestures do not always reflect user intention. Wobbrock et al. proposed a desktop-based hand gesture designing method by eliciting hand gestures from non-technical users [14], by first describing to them the effect of a gesture and then asking them to perform it according to their understanding. They recorded and analysed 1,080 gestures from 20 participants, finding that users were particularly concerned about the number of fingers they used. They also found that using one hand was more popular than two, whereas, in [12], users showed a preference towards using both hands. The above studies analyse user behaviour to design improved standardised hand gesture vocabularies, however, they do not address the problem that the definition of a hand gesture may vary widely among users from different e.g. cultural backgrounds.

The technical backbone of our method is the proposed camera-based HGR, which is another field that has already attracted considerable research interest. It is a multidisciplinary research field, combining elements of image processing, pattern recognition, and artificial intelligence [15]. The main challenge in camera-based HGR is the shielding of one hand from the other [16], which can lead to poor segmentation, low-quality feature vectors, and eventually a decline in recognition rates. To overcome this difficulty, Kishore et al. used a 4-camera system for recognizing gestures in Indian sign language [17]. Bauer et al. proposed a two-camera video-based HGR system based on a continuous density hidden Markov model and studied the effect of the various features of the hand gesture parameters on the recognition results [18]. Since the standard Chinese sign language contains more than 8000 words, Wang et al. considered dividing the recognition process into multiple sub-tasks [19]. Using few-shot learning for the sub-tasks, they could reduce the data acquisition cost and achieve short training times. However, recognizing a hand gesture from a single image or a sequence of images from a single camera remains a challenge, as it is generally difficult to infer 3D information from 2D data, and possible relationships between sequences may further complicate the task. Therefore, Ferreira et al. proposed a new model based on the Contrastive Transformer [20], which showed that learning rich representations from key-point sequences gives good discrimination between vector embeddings. Thus, here we use key point sequences generated by the Mediapipe tool [21], which generates a 21-point sequence for each RGB camera frame of a hand gesture.

Summarising the motivation for our work, while, as mentioned above, several studies have analysed user hand gesturing behaviour, with the aim of developing standardized hand gesture vocabularies, neither the users' diversity of cultural backgrounds nor their individual signing styles were taken into account. Moreover, most of the prior research on camera-based hand gesture recognition required data from multiple cameras to achieve high recognition accuracy rates. In reality, however, there is usually only a single screen and a single camera in front of a user, and thus improving single-camera-based hand gesture recognition will have immediate practical implications. We propose that a user interface with user-defined hand gesture textual inputs, operating with a single camera and requiring minimal data collection, will enable the individual user to use it in a way that is expressive to them, or, in other words, it will enable them to develop their own unique expressions.

To assess the usefulness of the proposed user interface with user-defined hand gesture textual inputs, the following two research questions were investigated:

RQ1: Is it feasible to train a light-weighted neural network as an interpreter, using user-defined small datasets for hand gesture words?
RQ2: Can users with no technical background accept a user interface with user-defined hand gestures?

The main contributions of this work are as follows:

1. A user interface with user-defined hand gesture textual inputs, enabling the user to define their own hand gestures according to their preferences;
2. A lightweight parallel Multilayer Perceptron (MLP) neural network based on the contrastive loss function, serving as the interpreter of user-defined hand gesture words, and achieving fast convergence and high accuracy;
3. A user study (N = 12) based on the analysis of a questionnaire to summarise user acceptability.

The remainder of this paper is organized as follows: Sect. 2 reviews related work, focusing on hand gesture detection and recognition, and user interfaces for user-defined hand gestures. Section 3 describes the user interface we designed and implemented, consisting of four modes: hand gesture verification, new hand gesture addition, hand gesture management, and model analysis. Section 4 describes the data we collected, data processing methods, and the proposed recognition model. Section 5 describes the experimental evaluation, including model performance analysis and the user study. In sect. 6, we discuss the advantages and the limitations of the proposed solution to the problem of hand

gesture variability, as well as future work. We briefly conclude in Sect. 7.

2 Related work

The proposed systems supporting user-defined hand gestures consist of two primary components: the first is concerned with the detection and gathering of user hand gesture data, and the second is the hand gesture recognition model. Here, we review prior research on hand gesture detection, hand gesture identification, and systems for creating customized hand gestures.

Hand gesture detection. Real-time detection of dynamic hand gestures from video streams is a challenging task since: (i) there is no indication when a hand gesture starts and ends in the video, (ii) a performed hand gesture should only be recognized once, and (iii) the entire system should be designed considering memory and computational power constraints. In [22], the user's hand was detected based on an estimation of the skin color of their face, through face detection. Pandey et al. demonstrated real-time egocentric hand gesture detection and localization on mobile headset-mounted displays [23]. Kopuklu et al. addressed these challenges by proposing a hierarchical structure enabling offline-working convolutional neural network (CNN) architectures to efficiently operate online by adopting the sliding window approach [24]. Neethu et al. proposed a hand gesture detection and recognition methodology using CNNs, achieving state-of-the-art performance [25]. Zhang et al. proposed MediaPipe [21], a real-time on-device hand tracking pipeline, to predict hand bone positions from a single RGB camera, aiming primarily at AR/VR applications. After reviewing the literature on camera-based hand gesture detection, we found Mediapipe to be a mature technology that is suitable for our purposes, and we thus employed it for the hand gesture detection component of our approach.

Regarding non-camera based systems, Pan et al. presented a universal framework to achieve dynamic gesture detection and recognition from WiFi signals [26]. Wang et al. proposed a method for continuous gesture detection and recognition, based on a Frequency Modulated Continuous Wave (FMCW) radar [27]. Yang et al. proposed a gesture recognition system, which processes range-Doppler-angle trajectories with Reused Long Short-Term Memory (RLSTM) network, using data from a 77GHz FMCW Multiple-Input-Multiple-Output (MIMO) radar [28].

Hand gesture recognition. As one would expect, the recent state-of-the-art methods are almost exclusively based on deep learning. Koller et al. utilised an end-to-end embedding of a CNN into an Hidden Markov Model, interpreting the outputs of the CNN in a Bayesian framework [29]. Bantupalli et al. extracted and processed temporal and spatial features from video sequences and created an application for hand gesture to text translation, aiding communication between signers and non-signers[16]. Rao et al. proposed a system for Indian sign language recognition [30], using CNNs, and they brought hand gesture recognition closer to a real time application deployed on a mobile platform [31]. Joze et al. introduced the first real-life large-scale hand gesture data set, comprising over 25,000 annotated videos, which they validated with state-of-the-art methods from sign and related action recognition [32]. Liao et al. proposed a multimodal dynamic hand gesture recognition method based on a deep 3-dimensional residual ConvNet, and Bi-directional Long Short-Term Memory (LSTM) networks, which they named BLSTM-3D residual network (B3D ResNet) [33]. We note that the state-of-the-art dynamic hand gesture recognition methods still have their limitations, which could result in low recognition accuracy, especially with complex hand gestures, and on very dynamic or particularly long video sequences. For a thorough survey of techniques and methods in hand gesture detection and recognition, we refer the readers to [34].

Beyond hand gesture recognition, Camgoz et al. introduced the Sign Language Translation (SLT) problem [35], which takes into account the grammatical and linguistic structure of sign languages. To evaluate the performance of their proposed Neural SLT, they introduced the first publicly available Continuous SLT dataset, RWTH-PHOENIX-Weather 2014T [35]. Mittal et al. proposed a modified LSTM model for continuous hand gesture recognition, aiming at recognizing sequences of connected gestures [36]. Finally, Camgoz et al. introduced a novel transformer-based architecture that jointly learns continuous hand gesture recognition and translation, being trainable in an end-to-end manner [37].

A common theme in the work we have reviewed is the use large data sets to train their complex neural networks, aiming at high accuracy race. The drawback of this approach is the high computational cost of the training phase, which would affect negatively the user experience, had it to be done frequently in order to support a dynamic vocabulary. Instead, here we need lightweight methods for hand gesture recognition, aiming at establishing a simple recognition model with satisfactory accuracy and fast convergence on small data sets.

User interfaces for user-defined hand gesture recognition. A lot of work has already been done on user-defined hand gesture user interfaces, but most of them support limited functionalities, such as letting the user select one out of two established hand gestures as the one they want to use. Moreover, we note that when designing unique hand gestures, it is also important to take into account how long it will take to collect the necessary data, as long times could negatively impact the user experience. For example,

the GESTOP tool proposed by Sk et al. [38] takes 15 to 20 min to complete data collection for a single hand gesture. For users to define multiple hand gestures, it would require several hours, which would be detrimental to the user experience. The user interface we propose here intents to improve greatly on the timings reported for the GESTOP tool, which we used as our baseline. Finally, we note that several user interfaces have been proposed for specific application domains. Wu et al. proposed an interface for users to customize hand gestures and apply them to VR shopping applications [39], while they proposed a user-defined hand gesture interface that could be used on in-vehicle information systems [10].

The majority of the prior work we have reviewed is based on fixed vocabularies, and usually, the users are provided with a large number of hand gesture words to learn and use. However, in practice, the users would only need a small portion of the vocabulary, i.e., a small number of words. In contrast, we provide users with input-able ways to develop a vocabulary corresponding to their own hand gestures, assuming that only a small number of gestures would eventually be defined. This simplification assumption allows us to employ a lightweight neural network for recognition trained with contrastive learning, which, as it is based on data augmentation, it further reduces the time a user needs to define a hand gesture through the interface (Fig. 1).

3 User interface

In this section we provide an overview of the user interface. Its main elements are:

1. A user login function, ensuring that users can only use the system under their account;
2. The default mode of our user interface is mode 0, which allows users to verify hand gesture textual inputs. Figure 2 illustrates this mode with a user doing the **OK** hand gesture in mode 0 to verify whether it can be recognized by the model. Users can change mode by clicking the *Switch* button. The implementation of the main functionality of mode 0 was based on the OpenCV [40] and Mediapipe [21] libraries. The OpenCV is used to capture the with the camera the user's image data stream. Mediapipe is an open-source hand gesture detection framework from Google, offering hand gesture detection and localization with high accuracy. We used it to detect the user's hand and calculate the x and y coordinates of 21 feature points on that image;
3. The vocabulary extension module (mode 1), shown in Fig. 3, is one of the most critical components of the system as it allows the addition of new words. In this mode, the user can input a specific word or phrase and then assign a corresponding gesture to it within the system. The mode's workflow is shown in Fig. 4. The user can enter the textual description for a new hand gesture by clicking on the button *Add*. The system will first check and inform them whether this word or text exists in the current vocabulary or not. Even if the textual input already exists, the system will remind the user that they can provide additional gesture data, and upon clicking the button *Collection*, the system will start capturing gesture data for 5 s. When data acquisition is complete, a pop-up window will inform the user, and upon clicking the button *Retrain*, the gesture recognition model will be trained. During model training, users have the option to switch to the hand gesture management mode of the interface (mode 2), where they can also utilise a large online dictionary to search for hand gestures of other expressions and words. After model training, a pop-up window will remind the user that they can switch to hand gesture verification in mode 0 to verify the newly trained model;

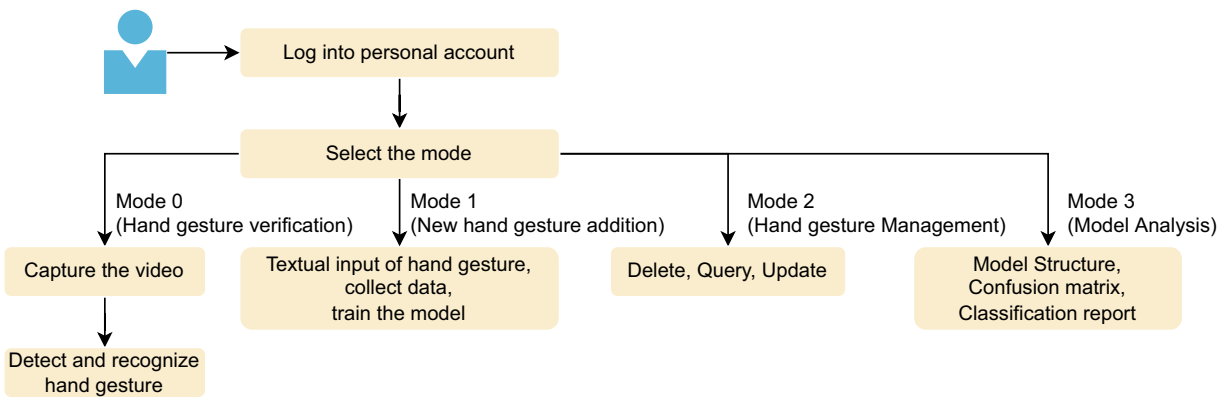

Fig. 1 The workflow of the user interface

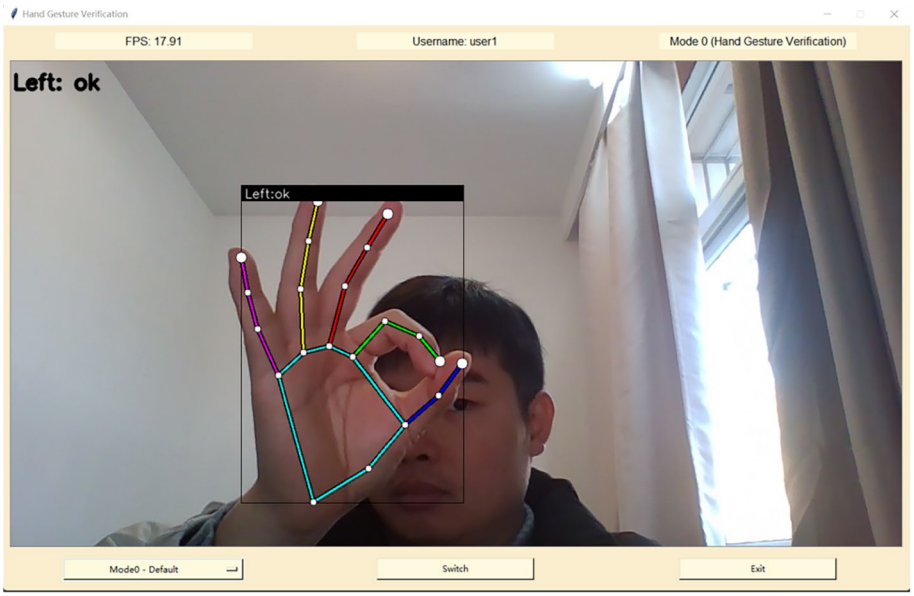

Fig. 2 Default mode of the user interface. Verification of user-defined hand gestures

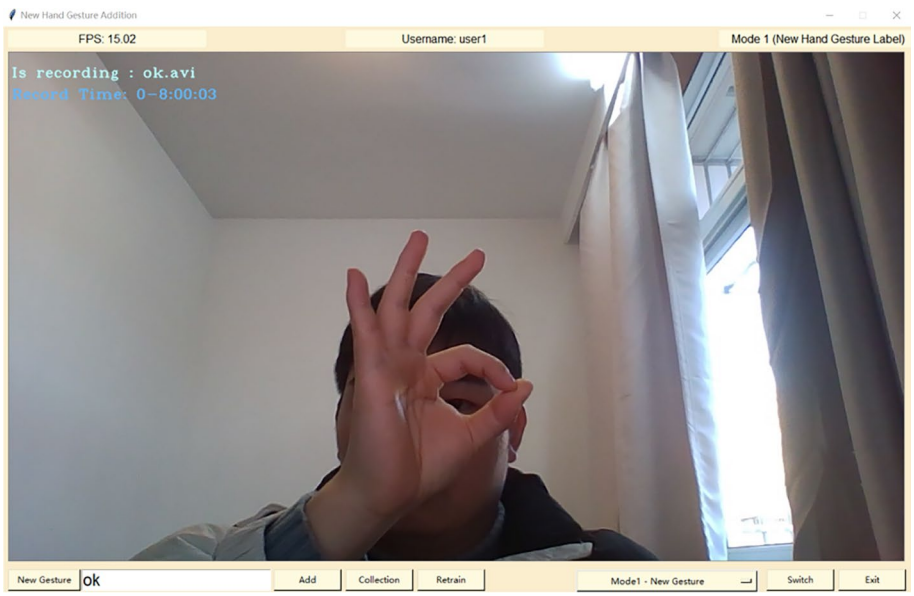

Fig. 3 Custom hand gesture mode (mode 1) of the user interface, supporting the addition of new hand gestures

4. A vocabulary management module (mode 2), detailed in Appendix A. In this mode, using the *query* function, a user can enter a word or phrase and watch videos with the corresponding gestures from the American Sign Language Lexicon Video Dataset (ASLLVD). This function is important since our primary purpose is not to enable users to invent new gestures, but rather allow them perform common gestures in their own personalised way. In mode 2, there is also a *delete* function, clicking which can erase a word or a phrase from the vocabulary, together with the collected gestures, and the recognition model is retrained;
5. A model analysis component (mode 3), detailed in Appendix B. In this mode, the user can generate analytical data on the recognition system and its performance, including training logs, confusion matrices, and a classification report with accuracy rates. For the effective use of this mode, the user would need to have some technical background, or be coached by someone with technical background.

The key elements of the user interface are depicted in the workflow diagram in Fig. 1. After signing up and logging

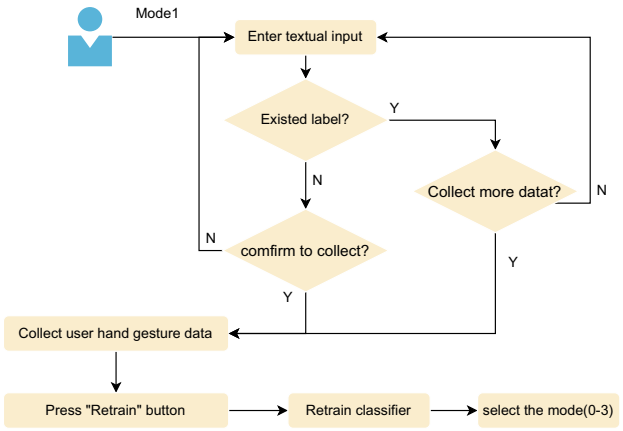

Fig. 4 Workflow of the custom hand gesture interface

in to the system, the user can choose between four modes. A new user should first choose the vocabulary extension mode (mode 1), in order to define a basic vocabulary. When entering a word or a phrase in that mode, if that textual input already exists, the system will prompt the user to decide whether they want to add more gestures corresponding to it. If they do, gesture data will be gathered for an additional 5 s, and the user would then choose between continuing creating gesture data or retraining the model. After the model has been trained once, the user can switch between the different modules of the interface, including mode 0 for hand gesture detection and recognition, mode 2 for exploring new hand gestures or deleting existing ones, and mode 3 where analytical data about the system's performance can be viewed.

4 Gesture recognition

This section describes the main aspects of the gesture recognition model, including data acquisition, augmentation, and normalisation, as well as the MLP's architecture and loss function.

4.1 Hand gesture data

4.1.1 Data Acquisition and Augmentation

Typically, the training of hand gesture detection and recognition tools utilises large datasets such as the American Sign Language Lexicon Video Dataset (ASLLVD) [41], the Chinese Sign Language Recognition Dataset (CSLRD) [42], or the Swiss German Sign Language Dataset [43]. Motivated by the observations that most of the vocabulary in these large datasets is rarely used in everyday life and that each user develops their own way to perform a hand gesture, here we use instead data collected from each user. Figure 5 shows the collection of the gesture data from a user for a small number of words or phrases chosen by them, here nine commonly used words. As the system records hand gestures for 5 s, and its camera has a frame rate of 18 fps, during

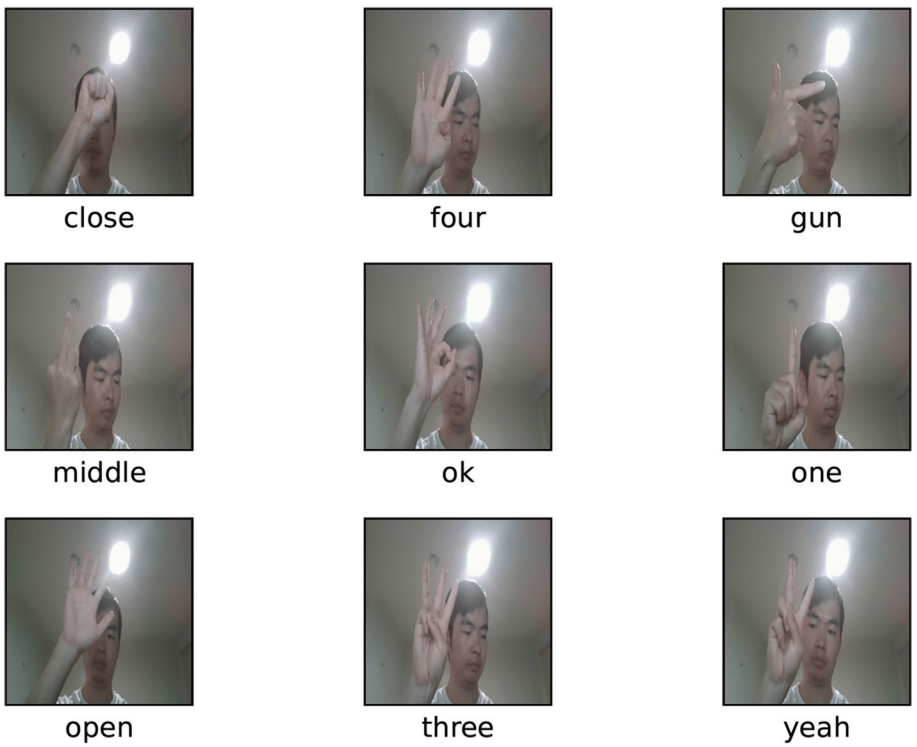

Fig. 5 User-defined hand gesture data

an acquisition session we collect about 90 images for each hand gesture.

As we train the model using contrastive learning [44], data augmentation is used to create positive samples. Each original video frame is rotated by 180 degrees about the origin and then scaled down by a factor of 0.8 to reduce the image size and thus, decrease the computational load and processing time. The origin of the frame is defined as the p_0 Mediapipe point - see Data Normalization below. Assuming that the amount of screen recording data collected by each user is N frames, the final amount of training data will be $2N$ frames, that is, twice the original.

4.1.2 Data Normalization

In machine learning, it is important to ensure that the training and test data follow similar distributions. Thus, in most applications, data is usually normalized, or in deep learning, each layer of the network will be normalized to ensure consistent data distributions.

In this paper, we normalise the data extracted by the Mediapipe, which is a sequences of 21 feature points (p_0, p_1, p_2,...,p_{20}) for each frame. An example of the 21 Mediapipe points is shown in Fig. 2. We set p_0, the point at the bottom of the palm near the wrist, as the origin of the frame's coordinate system. Let (x_i, y_i) be the coordinates of the point p_i. It is normalized by

$$x_i = \frac{x_i - x_0}{x_{max}}, \quad y_i = \frac{y_i - y_0}{y_{max}}, \qquad i = 1, 2, \ldots, 20. \tag{1}$$

where

$$\begin{aligned} x_{max} &= max(\,|x_1 - x_0|,\ |x_2 - x_0|,\ \ldots, |x_{20} - x_0|\,) \\ y_{max} &= max(\,|y_1 - y_0|,\ |y_2 - y_0|,\ \ldots, |y_{20} - y_0|\,) \end{aligned} \tag{2}$$

That is, for each frame we use a local coordinate system centered at p_0, and scale in the x and y directions such that the largest absolute value of a coordinate is 1.

4.2 The MLP architecture

Contrastive Learning focuses on learning common characteristics between similar instances and distinguishing the differences between non-similar instances [44]. It trains an encoder to encode instances within the same class with codes that are as similar as possible, while instances belonging to different classes should correspond to codes that are as different as possible [45]. Compared with generative learning [46], contrastive learning does not need to pay attention to the details of samples but only needs to learn to distinguish data in the feature space at the abstract semantic level. Therefore, model optimization becomes a simpler problem, and the solutions could have higher generalisation ability. Formally, the workflow of contrastive learning aims at maximising the distance between *positive* and *negative* samples, by decreasing the distance between positive samples and anchor points and increasing the distance between negative samples and anchor points. That is,

$$d(f(x), f(x^+)) << d(f(x), f(x^-)) \tag{3}$$

where x^+ denotes positive samples, x^- negative samples, and x anchor points.

We use two MLPs as the skeleton of the proposed contrastive learning model, as shown in the Fig. 6. Some previous works use images from the original dataset for contrastive learning and have achieved good results [47–49]. However, in our application we need a lightweight model, which should be able to be re-trained in real-time with user-provided data. Thus, the video stream is processed through the Mediapipe library, and as we discussed, 21 hand key points are obtained, significantly reducing the dimension of input data. In addition, considering that contrastive learning method will also used, we augment the Mediapipe data by rotating by 180 degrees and scaling. The original Mediapipe data is used as input to the first MLP. The augmented data is used as input to the second MLP, which has the same architecture as the first, and it is trained with contrastive learning using Eq. 4 as the loss function:

$$L_{out}^{sup} = \sum_{i \in I} L_{out,i}^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i * z_p / \tau)}{\sum_{a \in A(i)} \exp(z_i * z_a / \tau)} \tag{4}$$

For each class $i \in I$, that is, for each gesture i in the vocabulary I, $P(i)$ is the set of positive samples created from the frames in class i, $|P(i)|$ is their number, and $A(i)$ is the set of negative samples created from the frames in the other classes $I \setminus \{i\}$. The z_x are the feature vectors of the network just before the final Relu and Softmaxing steps, $*$ denotes inner product, and τ is a positive scalar parameter which is set at 0.1. The minimization of the loss function aims at making samples belonging to the same class closer and samples in different classes farther away.

5 Experimental evaluation

This section describes the details of the experimental evaluation of the system, which covers two main aspects. The first is the performance of the model. Considering that user customisation is at the heart of our approach, it is necessary to make the model converge fast to prevent long training times affecting user experience. The second aspect is the user experience itself. We invited users to use the system and provide feedback in the form of questionnaire scoring.

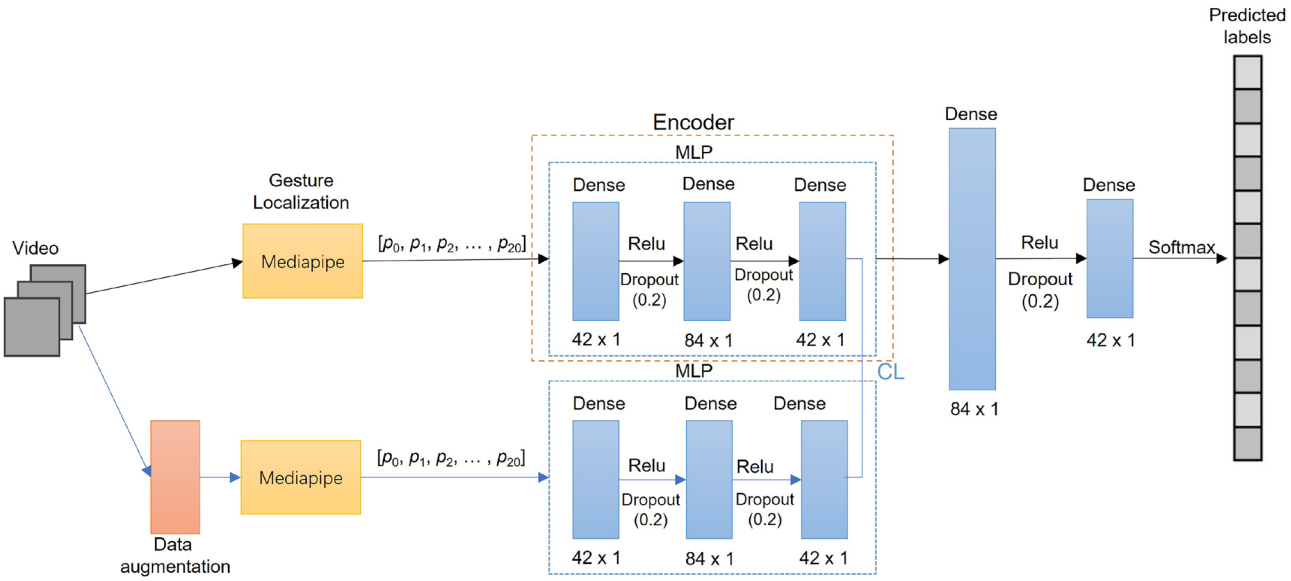

Fig. 6 The architecture of the two parallel MLPs

Our experiments were conducted on a high-end PC, running on an AMD Ryzen 9 5900HX processor. This CPU is well-suited for high-performance computing tasks due to its advanced multi-core architecture. The system was equipped with 32GB of RAM, providing ample memory for handling large datasets and complex computations. For graphical processing, we utilized an NVIDIA GeForce RTX 3080 GPU with 16GB of VRAM. This GPU is known for its high computational power and efficiency, making it particularly suitable for machine learning and other computationally intensive tasks.

5.1 RQ1 Evaluation - Model performance analysis

RQ1: ***can a lightweight network be trained with a small dataset?***

Because all the training data have to be generated by the individual user, each hand gesture is captured for only 5 s. That keeps the data capture period reasonably short, however, we have to analyse whether fast convergence with acceptable accuracy can be achieved. In the experiment, we compare three different model architectures. The first is a single 3-layer MLP with *"Sparse_Categorical_Crossentropy"* (SCC) as loss function, the second is a double 3-layer MLP with SCC, and the third is a double 3-layer MLP with *Contrastive loss* (CL) as loss function. We compare the performance of the three architectures on datasets with various number of gestures ($N_g = 4, 6, 8, 10, 12$). Table 1 shows that, compared to the two double MLP architectures, the accuracy of the single MLP architecture is significantly lower. Moreover, the double MLP achieves higher accuracy rates with the CL loss rather than the SCC loss.

Table 1 Model performance comparison

Model (Loss function)	Accuracy (%)	Loss	Convergence time (Epochs)
Gesture number = 4			
Single MLP (SCC)	88.39	0.2866	417
Double MLP (SCC)	92.90	0.2265	190
Double MLP (CL)	**96.13**	**0.1040**	**143**
Gesture number = 6			
Single MLP (SCC)	82.83	0.4251	479
Double MLP (SCC)	87.88	0.3367	387
Double MLP (CL)	**98.48**	**0.0430**	**369**
Gesture number = 8			
Single MLP (SCC)	71.00	0.6639	473
Double MLP (SCC)	87.67	0.4589	530
Double MLP (CL)	**96.00**	**0.1287**	**276**
Gesture number = 10			
Single MLP (SCC)	76.88	0.5070	**332**
Double MLP (SCC)	80.06	0.5862	708
Double MLP (CL)	**93.06**	**0.1674**	448
Gesture number = 12			
Single MLP (SCC)	81.07	0.5125	256
Double MLP (SCC)	87.62	0.3406	489
Double MLP (CL)	**90.29**	**0.3132**	**217**

Bold fonts represent the best performance under the current conditions (best accuracy, least loss, fastestconvergence)

The results show that the double MLP model with CL loss can achieve fast convergence and acceptable accuracy rates, at least in environments where limited vocabularies are used. Indeed, we notice that as the number of gestures in

the vocabulary rises, the accuracy rates decline. This is to be expected as the difficulty of the gesture recognition problem increases with the size of the vocabulary.

From the performance analysis of the model, we can see that small data sets can train a model with a good performance. Hence, users can use this user interface with user-defined hand gesture textual inputs to define their hand gestures, and this evaluation experiment answers the first research question.

5.2 User study

The user study aims at answering **RQ2:** ***can users with no technical background accept a user interface with user-defined hand gestures?***

We adopted the user experience investigation template proposed by [50], which assesses six key factors: **Attractiveness**, **Efficiency**, **Perspicuity**, **Dependability**, **Stimulation**, and **Novelty**. We invited 12 users to interact with the system and then complete a questionnaire. The users aged from 21 to 45. Half of them were male and half female. The survey comprised 6 parts, corresponding to Schrepp's six key factors, each of which consists of 6 or 7 questions. The users rated each question on a scale of 1 to 10.

The results of the survey are shown in Fig. 7. The overall score of the six factors is 6.96 (SD = 1.49), indicating that users are generally satisfied with the system. The average scores of **Novelty** and **Attractiveness** are 7.63 (SD = 1.06) and 7.24 (SD = 1.42), respectively, indicating that the system design is innovative and the users are willing to use it, although the system's security still needs improvement as the **Dependability** factor was scored the lowest - 6.32 (SD = 1.34).

For the **Attractiveness** factor (Fig. 8), most users felt pleasant when using the system, their average score being 8.03 (SD = 1.16). However, some users felt the interface not friendly and gave scores lower than 5 in the relevant question. This indicates that the system should be more

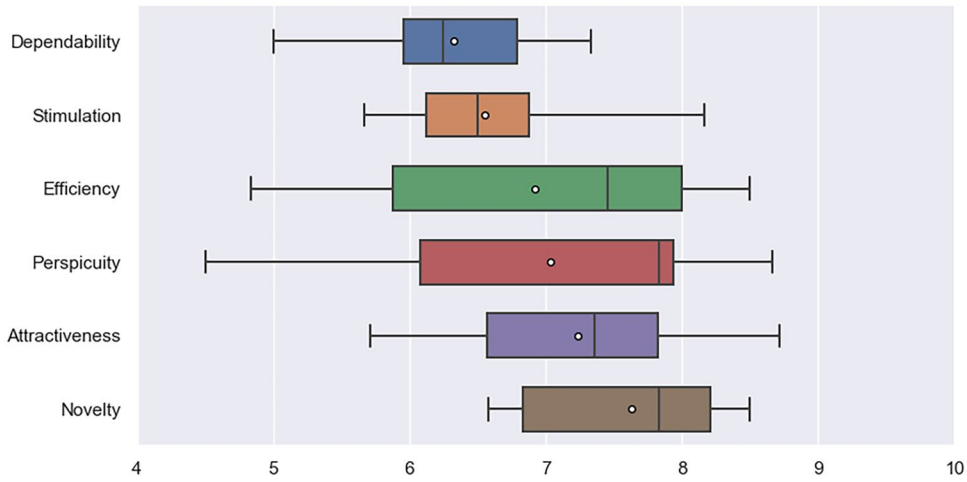

Fig. 7 A weighted composite score of six factors of the user questionnaire

Fig. 8 The scores of attractiveness factor

interactive, and that users need more detailed instructions on how to use it.

For the **Efficiency** factor (Fig. 9), the average score is 6.92 (SD = 1.65), which suggests that users generally perceived the system to be efficient. Each question in this factor has an average score of around 7.00, but some users gave a score of 4 on the question of how practical/impractical the system is. This may reflect their expectations that the system should have been embedded in other platforms, such as a VR scene, in order to give users a richer experience.

For the **Perspicuity** factor (Fig. 10), while the average score reaches 7.03 (SD = 1.49), individual low scores appear in all six questions, indicating that a small number of users still finds it difficult to use the product. This could be caused by the multiple steps required in the execution of some tasks. For example, to add a hand gesture, the user needs to enter their name, then click the add button and then the collection button to start recording the hand gesture video. Improvements could be made by simplifying the operations, or by giving more prompts and hints to the user, where necessary.

For the **Dependability** factor (Fig. 11), the average score is 6.33 (SD = 1.34). It suggests that most users feel that the execution steps of the system are predictable. However, some users gave low scores in the question of feeling secure/not secure when using the system. This could be a result of the user information management system being too simple and straightforward, lacking for example security verification processes such as login information, registration information, and password changes. We should therefore improve the system by adding a secure user information management system to make sure that the users' personal information is safer.

For the **Stimulation** factor (Fig. 12), the average score of 6.56 (SD = 1.19) indicates a generally positive attitude from most of the users, however, the score on the questions regarding excitement and motivation are lower than the rest. This might be caused by the lack of feedback in the system. For instance, currently, no feedback is provided during window loading or model training. Furthermore, the prototype only supports static user-defined hand gestures, which limits

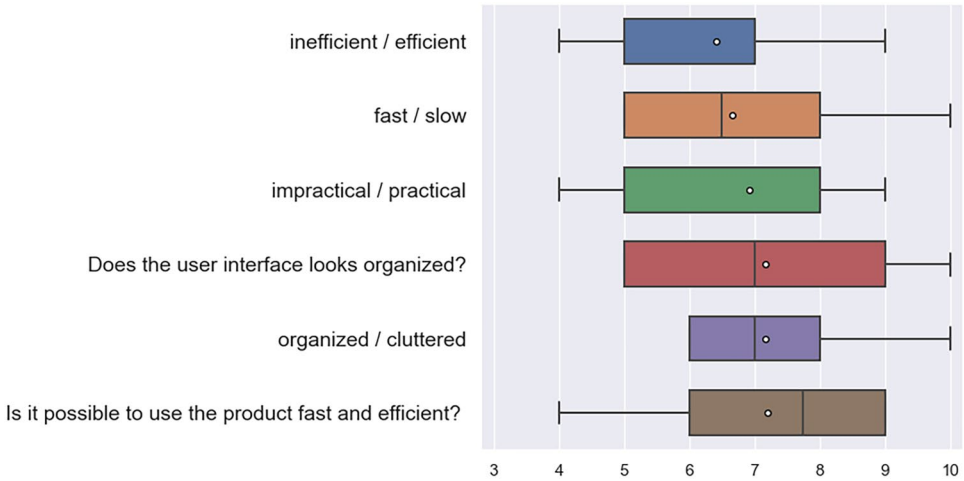

Fig. 9 The scores of efficiency factor

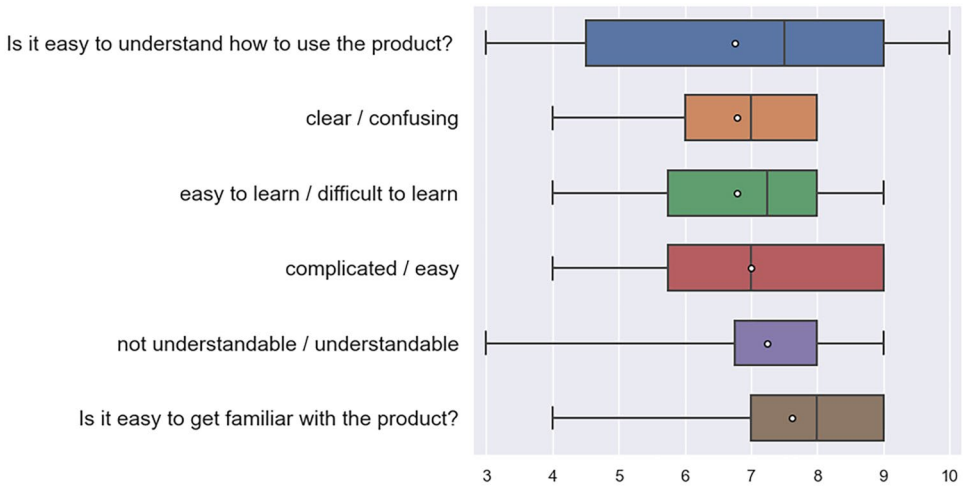

Fig. 10 The scores of perspicuity factor

its potential for user engagement. In the future, dynamic hand gestures will be supported, and users will be able to explore a richer, more capable system.

For the **Novelty** factor (Fig. 13), the average scores on all questions are over 7, showing that the novelty of the system was appreciated by the majority of users. Nevertheless, two users gave a neutral score of 5 on the question on whether the system grabs users attention. This may be due to the simplicity of the user interface design. The current version of the system mainly focuses on functionality, and future work will work on the usability and the aesthetics of the user interface to make for a more stimulating experience.

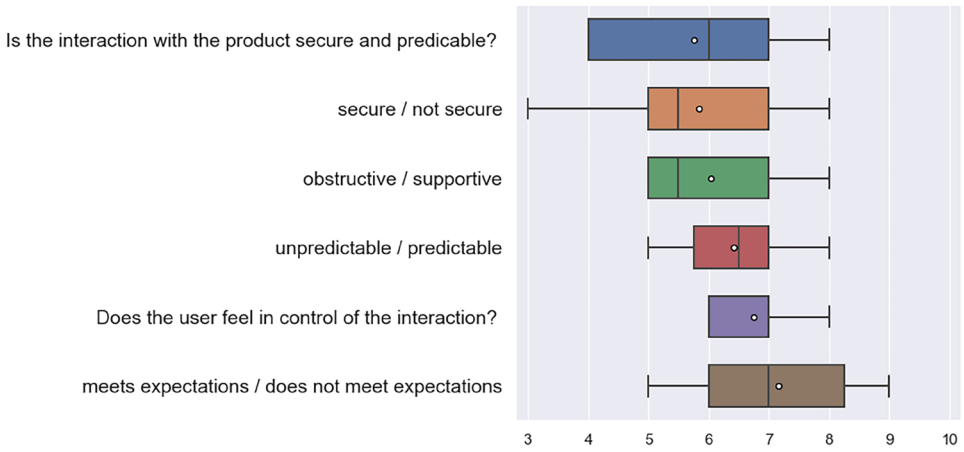

Fig. 11 The scores of dependability factor

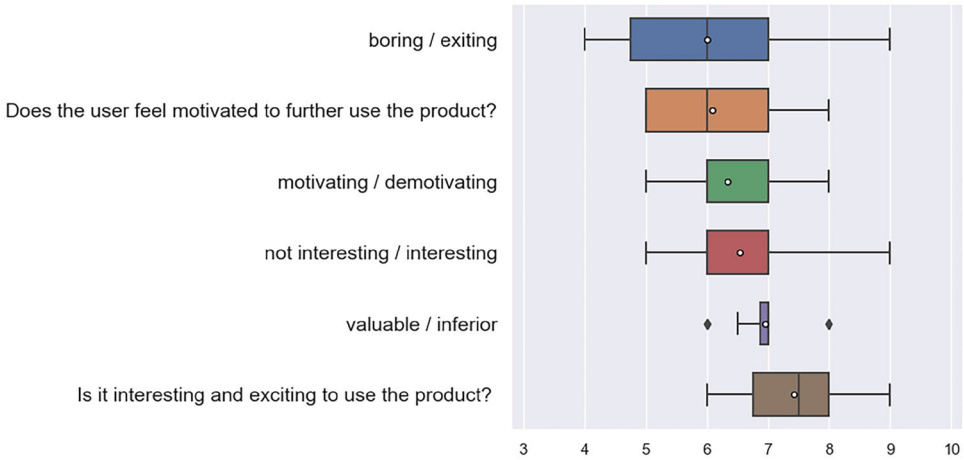

Fig. 12 The scores of stimulation factor

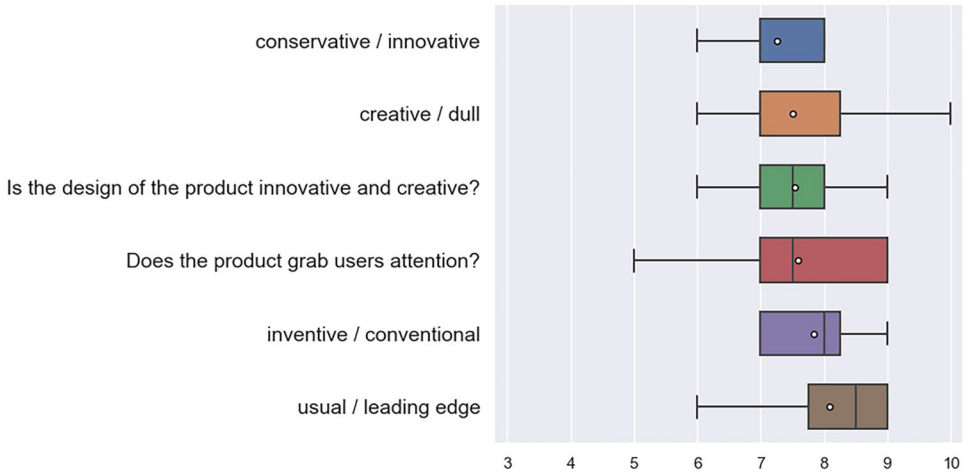

Fig. 13 The scores of novelty factor

6 Discussion and future work

In this section, we summarize the strong points and the limitations of the proposed system, as revealed by our qualitative and quantitative analysis and by the user survey. We also discuss some potential application scenarios and, briefly, our plans for future work on the extensions of the system.

6.1 Advantages

In our system, each user has their personal account and can develop their own, unique vocabulary and recognition model. The user interface allows them to enter any words or phrases and create a distinctive hand gesture representation of that textual input, in contrast to existing user-customized interfaces, which offer a limited set of options to achieve some limited customization. After the extraction of feature vectors with Mediapipe, we use a lightweight parallel MLP architecture and contrastive learning with data augmentation, achieving quick convergence and yet high accuracy. In the user survey, the majority of the respondents gave to most of the questions scores of 6 or more, that is, above the neutral limit of 5. This result indicates that many users will likely find the proposed system to be a viable solution to their communication needs.

6.2 Limitations

The main limitation of the proposed system is that, in its current implementation, it is based on static hand gestures. We believe that more research is required before deciding the best way forward, towards a system that will support dynamic hand gestures, and simultaneously retain all the advantages of the current system afforded by its lightweight recognition model. There are several other aspects to be improved, including a lack of interactivity of the user interface, and a perceived lack of system security. The aesthetics and the usability of the interface are also not optimal. For example, as we mentioned in Sect. 5, users found it somewhat tedious to add user-defined hand gestures as they had to click multiple buttons.

Another issue we identified during the evaluation is the demand for a relatively large amount of user data, which can have a negative effect on the user experience, especially on those using the interface for the first time. We note that this limitation is a direct consequence of the support for user-defined hand gestures which, in turn, is the key advantage and the central motivation for developing the system. Indeed, with the proposed system a user has the freedom and flexibility to define their own sign language and use it to communicate with the world, using the system as their interpreter. On the other hand, as we found during evaluation, for some at least of the users, the requirement to define more than six gestures and provide training and test data for each one of them, was seen as demanding and challenging.

In addition, in the development and assessment of our model, training was conducted on static datasets. Consequently, the knowledge base of the model remains unchanged post-training, lacking the capability to incorporate new information, unless retrained from scratch with an updated dataset. Thus, upon completion of its training phase, the model does not adapt or evolve in response to subsequent data, maintaining a fixed state of knowledge until a retraining process is initiated externally by user intervention. This design choice was dictated by the specific requirements and constraints of our project, including limitations in computational resources and the characteristics inherent to the data utilized, primarily its high-dimensional nature encompassing 21 distinct Mediapipe extracted points.

Finally, we mention are some rather modest but unavoidable limitations regarding hardware requirements; in particular, the system's inability to function without a camera.

6.3 Future work

Regarding our upcoming work, our first priority is to expand the system to support dynamic and mixed gestures. Additionally, we would like to assess and readdress the balance between a user's freedom and flexibility to define vocabulary and the associated hand gestures, and their need for structure and instruction in their interactions with the system. Thus, we will further study the degree to which a global hand gesture standard vocabulary is shared, and gather a significant quantity of data to create a pre-trained model that will aid, especially the new users, in their first steps.

In subsequent work, we would also like to extend our evaluation to include other sign language dictionaries for the users to get cues, which could provide further insights into the model's versatility and effectiveness across varied contexts.

We would also like to test and further develop our system on multiple scenarios in multi-device environments. In particular, we would like any updated system to have been tested under cross-device validation protocols.

6.4 Future application scenarios

The user interface we developed in this work can be used in a wide range of scenarios, as it is not affected by cultural variations, or by natural variations in human behaviour. Thus, it can be used as a camera-enabled interface in a variety of contexts, such as video conferencing [51], computer games [52], or virtual reality [53]. In particular, we note that as Metaverse develops, a growing number of scholars

is becoming interested in virtual reality. We hope that with the continued development of our system, eventually, we will be able to use it for hand gesture interaction in virtual reality applications, noting that hand gesture detection and recognition is a crucial component of that technology.

7 Conclusion

We presented a system where users can develop their own set of textual inputs and then correspond to them their own, personalized hand gestures. The user interface allows for the quick acquisition of a small set of hand gestures for training a lightweight parallel MLP architecture with contrastive learning, which it was shown to be sufficient for our purposes, leading to rapid convergence and acceptable accuracy rates.

Specifically, we first designed and implemented a graphical user interfaces supporting four types of functionality: hand gesture addition, hand gesture management, hand gesture verification, and model information inquiry. In a typical data acquisition session, the system gathers hand gesture data for 5 s; uses the Mediapipe library to locate 21 feature points; normalizes and saves these data in a local file. Next, the collected data are rotated and scaled to produce positive contrastive learning samples for training a lightweight parallel MLP architecture, which our experiments show it swiftly converges to acceptable accuracy rates. Finally, in order to assess user satisfaction and determine the acceptability of our interface, we conducted a user study (N = 12).

The results show that our main initial goal of providing users with an interface for defining a set of textual inputs and associating with them their own, personalized hand gestures, depending on their preferences and inventiveness, has been achieved. In the future, we would like to extend the system to support dynamic and mixed gestures, as well as develop the current prototype into a small module that can be integrated with a variety of devices and work in a variety of application scenarios.

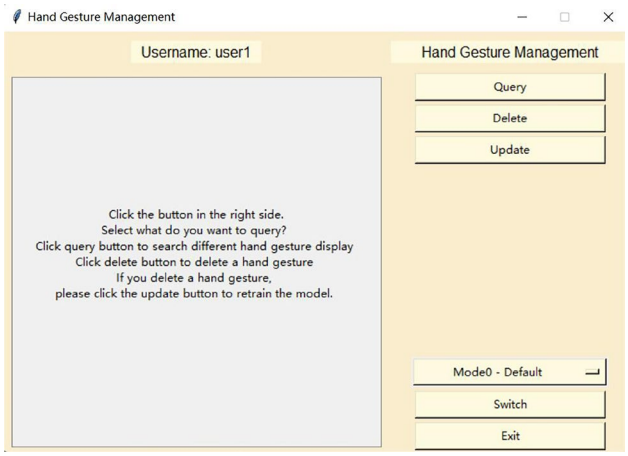

Fig. 14 The hand gesture management mode of the user interface

Appendix A Mode 2 - Hand gesture management

Mode 2 for hand gesture management, is also an essential part of the user interface. The user can query for other hand gesture expressions by clicking the button *Query* and can also click the button *Delete* to delete an already defined hand gesture, as shown in Fig. 14.

The query function returns results from the American hand gesture Lexicon Video Dataset (ASLLVD) [41] as suggested hand gesture representations. This dataset contains 132 hand gesture sequences, corresponding to various words, in 2 sessions, and we segmented them according to the number of frames in the corresponding video clips, as shown in Fig. 15.

After clicking the *Delete* button, the system will pop up a small window prompting the user to input the label to delete, as shown in Fig. 16. If the user has typed in an existing sign word which they want to delete, clicking the *OK* button will delete all data related to that sign word. After deleting the hand gesture words, the user will be reminded to click the *Update* button to update the model.

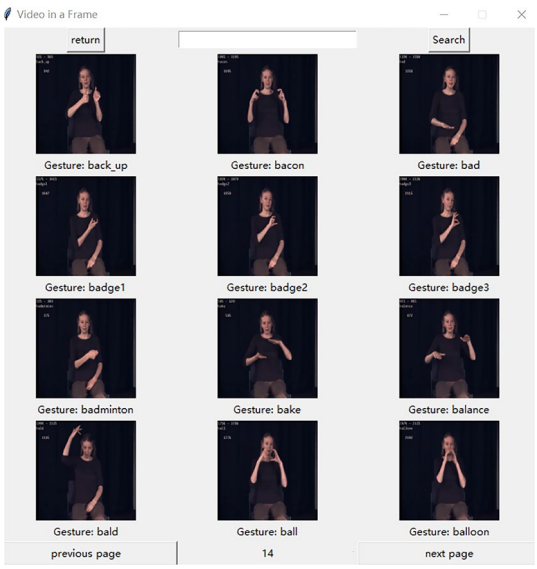

Fig. 15 Search result after clicking the query button

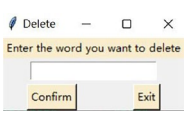

Fig. 16 The delete window, appearing after clicking the delete button

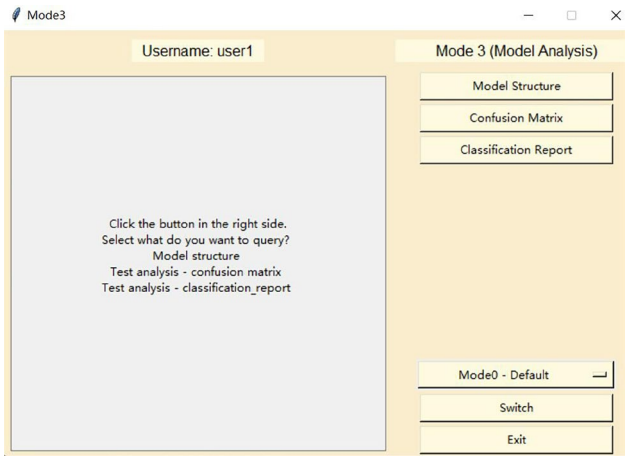

Fig. 17 Model Analysis Mode of the user interface

Appendix B Mode 3 - Model analysis

Mode 3 is the Model Analysis interface which is mainly for researchers and users with technical background. As shown in Fig. 17, the user can query the model structure by clicking on the button *Model Structure*, as well as compute the confusion matrix by clicking on the button *Confusion Matrix*, or receive the classification report by clicking the button *Classification Report*.

Author Contributions Conceptualization, J.W. and I.I.; Data collection and analysis, J.W.; Methodology, J.W., I.I., Z.L. and L.S.; Writing-original draft, J.W.; Writing-review and editing, J.W., I.I, Z.L. and L.S.; Supervision, I.I. and L.S.; All authors have read and agreed to the published version of the manuscript.

Funding This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through a Turing AI Fellowship (EP/V022067/1) on Citizen-Centric AI Systems (https://ccais.ac.uk/) and through the AutoTrust Platform Grant (EP/R029563/1);

Declarations

Conflict of interest The authors declare that they have no Conflict of interest;

Ethics approval This work has received ethics approval from Durham University;

Participants consent The work has received consent from the participants;

References

1. Wu, Y., Huang, T.S.: Human hand modeling, analysis and animation in the context of hci. In: proceedings 1999 international conference on image processing (Cat. 99CH36348), vol. 3, pp. 6–10 (1999). IEEE
2. Just, A.: Two-handed gestures for human-computer interaction. Technical report, IDIAP (2006)
3. Wu, C.-H., Chen, W.-L., Lin, C.H.: Depth-based hand gesture recognition. Multimed. Tools Appl. **75**(12), 7065–7086 (2016)
4. Ng, W.L., Ng, C.K., Noordin, N.K., et al.: Gesture based automating household appliances. In: international conference on human-computer interaction, pp. 285–293 (2011). Springer
5. Tao, L., Zappella, L., Hager, G.D., Vidal, R.: Surgical gesture segmentation and recognition. In: international conference on medical image computing and computer-assisted intervention, pp. 339–346 (2013). Springer
6. Kulshreshth, A., Pfeil, K., LaViola, J.J.: Enhancing the gaming experience using 3d spatial user interface technologies. IEEE comput. gr. appl. **37**(3), 16–23 (2017)
7. Sagayam, K.M., Hemanth, D.J.: Hand posture and gesture recognition techniques for virtual reality applications: a survey. Virtual Real. **21**(2), 91–107 (2017)
8. Lichtenauer, J.F., Hendriks, E.A., Reinders, M.J.: Sign language recognition by combining statistical dtw and independent classification. IEEE trans. pattern anal. mach. intell. **30**(11), 2040–2046 (2008)
9. Vatavu, R.-D.: User-defined gestures for free-hand tv control. In: proceedings of the 10th European conference on interactive Tv and Video, pp. 45–48 (2012)
10. Wu, H., Wang, Y., Liu, J., Qiu, J., Zhang, X.L.: User-defined gesture interaction for in-vehicle information systems. Multimed. Tools Appl. **79**(1), 263–288 (2020)
11. Wang, J., Ivrissimtzis, I., Li, Z., Zhou, Y., Shi, L.: User-defined hand gesture interface to improve user experience of learning american sign language. In: international conference on intelligent tutoring systems, pp. 479–490 (2023). Springer
12. Jahani, H., Kavakli, M.: Exploring a user-defined gesture vocabulary for descriptive mid-air interactions. Cognit. Technol. Work **20**(1), 11–22 (2018)
13. Piumsomboon, T., Clark, A., Billinghurst, M., Cockburn, A.: User-defined gestures for augmented reality. In: IFIP conference on human-computer interaction, pp. 282–299 (2013). Springer
14. Wobbrock, J.O., Morris, M.R., Wilson, A.D.: User-defined gestures for surface computing. In: proceedings of the SIGCHI conference on human factors in computing systems, pp. 1083–1092 (2009)
15. Cooper, H., Holt, B., Bowden, R.: Sign language recognition. In: Visual Analysis of Humans, pp. 539–562. Springer, ??? (2011)
16. Bantupalli, K., Xie, Y.: American sign language recognition using deep learning and computer vision. In: 2018 IEEE international conference on big data (Big Data), pp. 4896–4899 (2018). IEEE

17. Kishore, P., Prasad, M.V., Prasad, C.R., Rahul, R.: 4-camera model for sign language recognition using elliptical fourier descriptors and ann. In: 2015 international conference on signal processing and communication engineering systems, pp. 34–38 (2015). IEEE
18. Bauer, B., Hienz, H.: Relevant features for video-based continuous sign language recognition. In: proceedings Fourth IEEE international conference on automatic face and gesture recognition (Cat. No. PR00580), pp. 440–445 (2000). IEEE
19. Wang, F., Li, C., Zeng, Z., Xu, K., Cheng, S., Liu, Y., Sun, S.: Cornerstone network with feature extractor: a metric-based few-shot model for chinese natural sign language. Appl. Intell. **51**(10), 7139–7150 (2021)
20. Ferreira, S., Costa, E., Dahia, M., Rocha, J.: A transformer-based contrastive learning approach for few-shot sign language recognition. arXiv preprint arXiv:2204.02803 (2022)
21. 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)
22. Ikegami, S., Premachandra, C., Sudantha, B., Sumathipala, S.: A study on mobile robot control by hand gesture detection. In: 2018 3rd international conference on information technology research (ICITR), pp. 1–5 (2018). IEEE
23. Pandey, R., White, M., Pidlypenskyi, P., Wang, X., Kaeser-Chen, C.: Real-time egocentric gesture recognition on mobile head mounted displays. arXiv preprint arXiv:1712.04961 (2017)
24. Köpüklü, O., Gunduz, A., Kose, N., Rigoll, G.: Real-time hand gesture detection and classification using convolutional neural networks. In: 2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019), pp. 1–8 (2019). IEEE
25. Neethu, P., Suguna, R., Sathish, D.: An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks. Soft Comput. **24**(20), 15239–15248 (2020)
26. Pan, X., Jiang, T., Li, X., Ding, X., Wang, Y., Li, Y.: Dynamic hand gesture detection and recognition with wifi signal based on 1d-cnn. In: 2019 IEEE international conference on communications workshops (ICC Workshops), pp. 1–6 (2019). IEEE
27. Wang, Y., Ren, A., Zhou, M., Wang, W., Yang, X.: A novel detection and recognition method for continuous hand gesture using fmcw radar. IEEE Access **8**, 167264–167275 (2020)
28. Yang, Z., Zheng, X.: Hand gesture recognition based on trajectories features and computation-efficient reused lstm network. IEEE Sens. J. **21**(15), 16945–16960 (2021)
29. Koller, O., Zargaran, S., Ney, H., Bowden, R.: Deep sign: enabling robust statistical continuous sign language recognition via hybrid cnn-hmms. Int. J. Comput. Vis. **126**(12), 1311–1325 (2018)
30. Rao, G.A., Syamala, K., Kishore, P., Sastry, A.: Deep convolutional neural networks for sign language recognition. In: 2018 conference on signal processing and communication engineering systems (SPACES), pp. 194–197 (2018). IEEE
31. Rao, G.A., Kishore, P.: Selfie video based continuous indian sign language recognition system. Ain Shams Eng. J. **9**(4), 1929–1939 (2018)
32. Joze, H.R.V., Koller, O.: Ms-asl: A large-scale data set and benchmark for understanding american sign language. arXiv preprint arXiv:1812.01053 (2018)
33. Liao, Y., Xiong, P., Min, W., Min, W., Lu, J.: Dynamic sign language recognition based on video sequence with blstm-3d residual networks. IEEE Access **7**, 38044–38054 (2019)
34. Cheok, M.J., Omar, Z., Jaward, M.H.: A review of hand gesture and sign language recognition techniques. Int. J. Mach. Learn. Cybern. **10**(1), 131–153 (2019)
35. Camgoz, N.C., Hadfield, S., Koller, O., Ney, H., Bowden, R.: Neural sign language translation. In: proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7784–7793 (2018)
36. Mittal, A., Kumar, P., Roy, P.P., Balasubramanian, R., Chaudhuri, B.B.: A modified lstm model for continuous sign language recognition using leap motion. IEEE Sens. J. **19**(16), 7056–7063 (2019)
37. Camgoz, N.C., Koller, O., Hadfield, S., Bowden, R.: Sign language transformers: Joint end-to-end sign language recognition and translation. In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10023–10033 (2020)
38. SK, S., Sinha, N.: Gestop: Customizable gesture control of computer systems. In: proceedings of the 3rd ACM India joint international conference on data science & management of data (8th ACM IKDD CODS & 26th COMAD), pp. 405–409 (2021)
39. Wu, H., Wang, Y., Qiu, J., Liu, J., Zhang, X.: User-defined gesture interaction for immersive vr shopping applications. Behav. Inf. Technol. **38**(7), 726–741 (2019)
40. Bradski, G., Kaehler, A.: Opencv. Dr. Dobb's journal of software tools **3**, 120 (2000)
41. Athitsos, V., Neidle, C., Sclaroff, S., Nash, J., Stefan, A., Yuan, Q., Thangali, A.: The american sign language lexicon video dataset. In: 2008 IEEE computer society conference on computer vision and pattern recognition workshops, pp. 1–8 (2008). IEEE
42. Zhang, J., Zhou, W., Xie, C., Pu, J., Li, H.: Chinese sign language recognition with adaptive hmm. In: 2016 IEEE international conference on multimedia and expo (ICME), pp. 1–6 (2016). IEEE
43. Ebling, S., Camgöz, N.C., Braem, P.B., Tissi, K., Sidler-Miserez, S., Stoll, S., Hadfield, S., Haug, T., Bowden, R., Tornay, S., et al.: Smile swiss german sign language dataset. In: proceedings of the 11th international conference on language resources and evaluation (LREC) 2018 (2018). The European Language Resources Association (ELRA)
44. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Adv. Neural Inf. Process. Syst. **33**, 18661–18673 (2020)
45. Park, T., Efros, A.A., Zhang, R., Zhu, J.-Y.: Contrastive learning for unpaired image-to-image translation. In: European conference on computer vision, pp. 319–345 (2020). Springer
46. Wittrock, M.C.: Generative learning processes of the brain. Educ. Psychol. **27**(4), 531–541 (1992)
47. Dai, B., Lin, D.: Contrastive learning for image captioning. Adv. Neural Inf. Process. Syst. **30,** 898–907 (2017)
48. Madhusudana, P.C., Birkbeck, N., Wang, Y., Adsumilli, B., Bovik, A.C.: Image quality assessment using contrastive learning. IEEE Trans. Image Process. **31**, 4149–4161 (2022)
49. Wang, X., Qi, G.-J.: Contrastive learning with stronger augmentations. IEEE trans. pattern anal. mach. intell. **45**(5), 5549–5560 (2022)
50. Schrepp, M., Hinderks, A., Thomaschewski, J.: Applying the user experience questionnaire (ueq) in different evaluation scenarios. In: international conference of design, user experience, and usability, pp. 383–392 (2014). Springer
51. Rautaray, S.S., Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey. Artif. intell. rev. **43**(1), 1–54 (2015)
52. Ren, Z., Meng, J., Yuan, J.: Depth camera based hand gesture recognition and its applications in human-computer-interaction. In: 2011 8th international conference on information, communications & signal processing, pp. 1–5 (2011). IEEE
53. Oudah, M., Al-Naji, A., Chahl, J.: Hand gesture recognition based on computer vision: a review of techniques. J. Imaging **6**(8), 73 (2020)