Personalized Facial Gesture Recognition For Accessible Mobile Gaming

Matteo Manzoni[1 [0009−0002−1946−5919]], Dragan Ahmetovic[1 [0000−0001−5745−1230]], and Sergio Mascetti[1 [0000−0002−8416−4023]]

Department of Computer Science, Università degli Studi di Milano, Milan, Italy
matteo.manzoni2@studenti.unimi.it, dragan.ahmetovic@unimi.it,
sergio.mascetti@unimi.it

Abstract. For people with upper extremity motor impairments, interaction with mobile devices is challenging because it relies on the use of the touchscreen. Existing assistive solutions replace inaccessible touchscreen interactions with sequences of simpler and accessible ones. However, the resulting sequence takes longer to perform than the original interaction, and therefore it is unsuitable for mobile video games. In this paper, we expand our prior work on accessible interaction substitutions for video games with a new interaction modality: using facial gestures. Our approach allows users to play existing mobile video games using custom facial gestures. The gestures are defined by each user according to their own needs, and the system is trained with a small number of face gesture samples collected from the user. The recorded gestures are then mapped to the touchscreen interactions required to play a target game. Each interaction corresponds to a single face gesture, making this approach suitable for the interaction with video games. We describe the facial gesture recognition pipeline, motivating the implementation choices through preliminary experiments conducted on example videos of face gestures collected by one user without impairments. Preliminary results show that an accurate classification of facial gestures (97%) is possible even with as few as 5 samples of the user.

Keywords: Upper extremity motor impairments · Mobile devices · Video games · Face gestures recognition.

1 Introduction

For people with Upper Extremity Motor Impairments (UEMI), the interaction with mobile devices is challenging because it largely relies on the use of the touchscreen interface and therefore on the manual ability of the user [9]. Specific challenges with touchscreen use may also vary based on the actual condition of each user with UEMI. Indeed, some conditions cause difficulties in performing or sustaining precise movements (*e.g.*, cerebral palsy). Other conditions may impair hand strength, and some users may not have any mobility in upper extremities.

Assistive technologies that replace touchscreen interactions with sequences of simpler, more accessible ones have been proposed. However, these sequential interactions are slow and, therefore, not suitable for time-constrained interaction (*e.g.*, games). In our previous work [3], we propose one-to-one remapping of touchscreen interactions to alternative inputs as a way to enable accessibility of existing games by people with UEMI. As a part of this prior work, we have conducted studies with people with UEMI on the use of external switches and vocal sounds with promising results. However, for those users who cannot access external switches and have a speech impairment (*e.g.*, anarthria), these interactions are still inaccessible.

We propose a new one-to-one interaction substitution method based on personalized Facial Gestures (FGs) recognition. To account for the specific needs of different users with UEMI, our approach relies on few-shot learning to allow the users to define and register their own FGs, with just a few samples of each gesture. The recorded FGs are then mapped to the touchscreen interactions needed to play a target game. Various existing mobile games can be used, including many popular ones that are available to users without disabilities.

In this work, we describe the FG recognition pipeline. In particular, we detail the processes of feature selection, few-shot learning, result aggregation, and fine-tuning. Preliminary experiments on videos of FGs collected by one user without UEMI yield a classification accuracy of 96.99% and the ability to process 8.26 ± 1.55 frames per second on a commodity Android device. As future work, we will conduct a thorough empirical evaluation with representative participants, focusing on confirming the applicability of the proposed approach and measuring its accuracy and appreciation by the target population.

2 Related Work

Mobile device accessibility for people with UEMI [9] is provided through accessibility services (ASs) [1], running in the background that completely replace the default touchscreen interaction paradigm, providing substitutive interactions for all mobile device functionalities. Scanning approaches [2] replace direct selection of a target User Interface (UI) element on the screen with sequential traversal of all UI elements, activating the target once it has been reached. Interactions may be provided through simplified touchscreen gestures, external switch peripherals [2], or FGs [14]. Direct activation of UI elements is also possible through gaze tracking [13] or verbal instructions [15]. However, all these approaches are slower than direct touchscreen access, which makes them unsuitable for mobile gaming [4]. To address this issue, in our previous work we have explored direct remapping of game UI elements to alternative user actions [3]. Specifically, we have proposed vocal interactions or external switch activations as alternative user actions, achieving comparable accuracy and reactivity with respect to touchscreen interaction. However, for users with UEMI who have a speech impairment and cannot access external switches (*e.g.*, cerebral palsy [**?**] or anarthria [8]), these interactions are still inaccessible. Hence, in this paper, we extend our previous approach [3] with a novel interaction modality based on FG recognition.

Prior literature identifies two key approaches for FG recognition [10]. One frequently used approach is to use deep convolutional neural networks to extract facial features and classify images into FGs. The other common approach first detects geometric facial landmarks, such as the position of the eyes, nose, and mouth. Then it extracts meaningful high-level features, such as distances between the landmarks and areas of polygons defined by sets of them. Finally, simpler machine learning models, like Support Vector Machines (SVMs), are used to classify these features into FGs. We highlight that both approaches are designed to classify a fixed set of predefined FGs, which are same for all the users. Furthermore, to train generalizable classifiers that robustly recognize FGs for different users, a large amount of data is required.

However, for some people with UEMI, none of these methods may be appropriate due to the characteristics of their specific motor impairment that may prevent them from making predefined FGs recognized by the classifier [14]. To address this issue, we propose the use of user-defined gestures. This approach requires user-specific training of the FG recognition model, using only a few examples of FGs that can be collected from a given user. To account for this requirement, our approach combines two machine-learning models: a pre-trained deep-learning model to extract facial landmarks [6] and prototypical networks [12], a few-shot learning approach suitable for FG classification.

3 Methodology

The proposed pipeline is divided into four main steps (see Fig. 1): Landmark detection and normalization (Section 3.1), Feature extraction and selection (Section 3.2), Classification (Section 3.3), and Post-processing (Section 3.4).

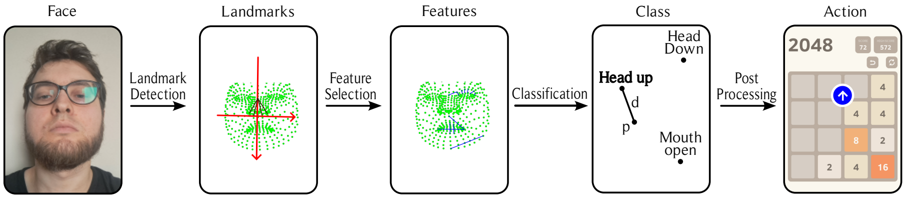


Fig. 1: Steps of the FG recognition pipeline

3.1 Landmark detection and normalization

To detect the face landmarks, we used a pre-trained network. Specifically, we used the *Mediapipe Face mesh* library [6] that is capable of real-time detection of 478 facial landmarks, whose position is determined in the image coordinate system. To robustly detect the FGs even if the user moves, the landmark coordinates are then made invariant with respect to the position of the face in the image or its distance from the camera using two approaches. First, we convert the landmarks into a face-centric coordinate system (the axes are the horizontal and the vertical lines shown in red in Fig. 1). Second, the landmark coordinates are normalized with respect to a set of reference distances between pairs of pre-defined landmarks (brown segments connecting the base and the tip of the nose). This set was obtained experimentally by selecting the pairs of points whose distances change the least in various FGs.

3.2 Feature extraction and selection

From the normalized landmarks, we extract a set of features to be used as input to the classifier. We considered different possible sets of features, among which we selected the one that yields the most accurate FG recognition (see Section 4.1). For our data, the final set of features included head rotation and distances between pairs of landmarks that are close to each other.

Since the detection runs on a resource-constrained device, we also apply the following feature selection procedure. At training time, the features are ordered according to their ability to discriminate different FGs using Fisher score [5]. Then, for each feature starting from the one with the highest score and iterating over the others, the technique computes correlation of that feature with all other features, removing the ones that have a correlation higher than a given threshold with the initial one. An example of features remaining after the selection is shown in Fig. 1 (blue segments above the left eye, on the mouth, and on the chin).

3.3 Classification

During the app setup, the user records a sequence of frames (5 in our tests) for each FG they intend to use to interact with a game. Features are extracted from these frames and used to train a Prototypical Networks model, a machine learning approach suitable for few-shot learning [12]. Specifically, a class prototype is defined for each considered FG using the extracted features as training data.

During inference, input video frames are classified using the trained network. Specifically, considering the features of the FG represented in the input frame, the closest prototype is set as its class. Additionally, we compute the distance value d between the input features and the closest prototype. Clearly, the smaller the value d, the higher the classification confidence with respect to the considered prototype. Based on this intuition, we empirically define three confidence levels (*High*, *Medium*, and *Low*) considering the distance d. If d is greater than the minimum distance between prototypes ($t = 0.8$ in our dataset), the FG is detected with *Low* confidence. Instead, to compute the threshold between *Medium* and *High* confidence, we examined how the classification accuracy varies considering different distance thresholds (see Section 4.2), finally selecting the value $T = 0.32$. Both thresholds can be tuned to the user-specific FG dataset.

3.4 Post-processing

The post-processing phase has three objectives: to smooth fluctuating classification results, to convert the classification results into actionable events (start/stop a tap) and to take into account the situation in which the user does not make any recorded FG (the "other" case). The proposed solution is based on two logical components: a filter and a finite-state machine. The filter takes in input the classification result and, if its confidence level is *high*, returns its class. If the confidence level is *medium*, the filter adds the class to a buffer. The buffer has a pre-defined size, which is a system parameter. When more than half of the buffer spaces contain the same class, the filter returns that class. If the confidence level is *low* a "other" class is added to the buffer. The finite state machine defines a state for each class, plus a "other" state. It takes in input the class returned by the filter and changes the state (if needed) to the corresponding class. Upon leaving/entering a state, the system triggers a *end tap*/*start tap* action. The only exception is the "other" state, which does not trigger any action.

4 Results

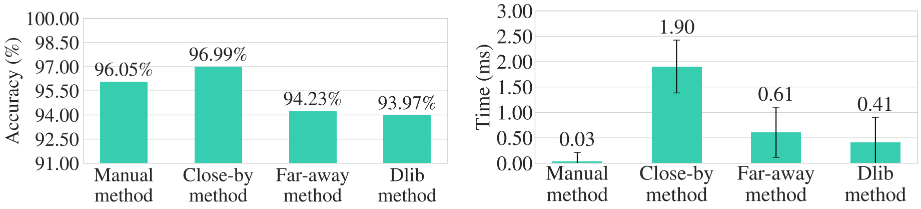
We conducted an extended set of performance tests aimed at tuning the system parameters. The experiments were carried out using a set of 120 videos, each representing a person making one of 12 representative FGs[1] (10 videos for each FG). The videos were collected by one user without UEMI.

4.1 Feature extraction

We experimented with various sets of features, with the aim of balancing accuracy and the computational cost due to the large number of features. Specifically, we considered: manually selected features based on prior literature [11] including selected distances between landmarks, areas defined by sets of landmarks, and vertical inclinations of segments defined by pairs of landmarks (**Manual method**); distances between pairs of close-by landmarks (**Close-by method**); – distances between pairs of far-away landmarks (**Far-away method**); – distances between pairs of *Dlib* landmarks[2] (***Dlib* method**). Head rotations were considered as an additional feature in all cases. The best overall results (96.99% accuracy) were obtained considering close-by landmarks, with a set of 7098 features (see Fig. 2a), and a processing time overhead of 0.41ms (see Fig. 2b). The intuition for using close-by landmarks is that facial expressions are characterized by local changes (*e.g.*, the landmarks around the mouth when it is open or closed). Using these features and implementing the system on a *Samsung Galaxy A53 5G*, the system can process 8.26 ± 1.55 frames per second.

[1] FGs considered in the experiment: smile, open mouth, close left/right eye, curve eybrows, wrinkle nose, turn left/right, incline left/right, raise/lower head.

[2] Subset of *Mediapipe Face Mesh* [6] landmarks that are also defined in *Dlib* [7].



(a) Accuracy

(b) Mean processing time

Fig. 2: Feature extraction accuracy and processing time by extraction method

4.2 Classification

Fig. 3 shows how classification accuracy varies considering different values of the distance threshold T. For each threshold, the graph shows the percentage of images ("Support") whose distance d falls within that threshold and the accuracy achieved on those images. Based on these results, we select the threshold value $T = 0.32$. This value is data-specific, and can be tuned for each user.

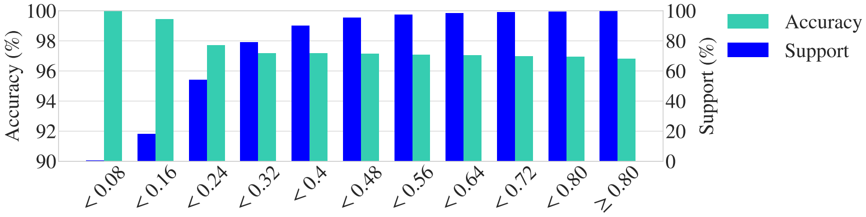


Fig. 3: Aggregate accuracy by intervals of d (distance from closest prototype)

5 Discussion

5.1 Detection Robustness, Personalization, and Generalizability

We designed the FG detection pipeline with three goals in mind: 1) accurate and timely interaction, suitable for mobile gaming, 2) personalization of FGs according to the user's needs, and 3) need for limited training data. The results obtained from the preliminary tests show that our approach satisfies all three criteria. By design, the proposed technique is robust to user movement and distance from the camera. It is able to process more than 8 frames per second on a commodity mobile device, with 97% accuracy. It enables a user to define personalized FGs and use them as interactions to play existing mobile video games. Finally, it only requires a short video of each FG for the training.

These results were computed on a preliminary dataset collected by one user without impairments. The aim was solely to assess the feasibility of the proposed technique and to tune the system parameters during the process. Thus, we cannot consider these results to be representative for users with UEMI. However, since the FGs are personalized and trained separately for each user, the process in itself should apply to users with UEMI without modifications.

5.2 Limitations

Despite the positive results, we acknowledge the technical and practical limitations of our approach. First, the applicability of the approach is limited by the ability to discern facial landmarks in the video frame, and therefore it is not possible in adverse luminosity conditions such as extreme dark or light glare. The presence of others in the camera frame, as well as occlusion of the facial features, may also influence the ability of the system to correctly detect the user's FGs.

Clearly, the applicability of the approach is also limited by the user's ability to make FGs. For users with limited mobility or difficulties in controlling their head and facial features (*e.g.*, people with dystonia), this approach is not suitable. In such cases, other input modalities could be used, such as non-verbal voice interaction or external switches [3].

Finally, the main methodological limitation of our work is the lack of an evaluation by users with UEMI. The overall interaction substitution approach, with other types of input (non-verbal voice input or external switches), has been evaluated with representative users [3]. However, to assess the generalizability of the FG interaction modality, additional user studies with representative users are needed to robustly assess the actual recognition accuracy of the technique.

6 Conclusions and Future Work

This paper describes a pipeline for recognizing personalized facial gestures, selected by the users themselves, to be used as a new interaction approach for making existing mobile games accessible to people with upper extremity motor impairments. After detecting facial landmarks [6] corresponding to a facial gesture, a set of robust features is selected and used to train a Prototypical Network model [12]. Only a short configuration step is required to register the personalized gestures and associate them to the desired game. Afterwards, the system will recognize the registered gestures and trigger the associated game interactions, thus enabling users with UEMI to play the configured games.

As future work we will assess the validity of the proposed approach with representative users, using existing mobile games. In particular, we will assess the accuracy of the approach, the reaction time of the users when using the new interaction modality, and the cognitive and physical load associated to its use. Furthermore we will assess its usability and appreciation by users with UEMI, as well as the applicability of the approach to users with different abilities.

Disclosure of Interests. The authors have no competing interests.

Bibliography

[1] Android accessibility services, https://developer.android.com/guide/topics/ui/accessibility/service

[2] Android accessibility suite, https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback

[3] Ahmetovic, D., Riboli, D., Bernareggi, C., Mascetti, S.: Replay: Touchscreen interaction substitution method for accessible gaming. In: Human Computer Interaction with Mobile Devices and Services. ACM (2021)

[4] Bierre, K., Chetwynd, J., Ellis, B., Hinn, D.M., Ludi, S., Westin, T.: Game not over: Accessibility issues in video games. In: Universal Access in Human-Computer Interaction (2005)

[5] Gu, Q., Li, Z., Han, J.: Generalized fisher score for feature selection. In: Conference on Uncertainty in Artificial Intelligence (2011)

[6] Kartynnik, Y., Ablavatski, A., Grishchenko, I., Grundmann, M.: Real-time facial surface geometry from monocular video on mobile gpus. arXiv preprint arXiv:1907.06724 (2019)

[7] King, D.E.: Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research (2009)

[8] Lecours, A., Lhermitte, F.: The "pure form" of the phonetic disintegration syndrome (pure anarthria); anatomo-clinical report of a historical case. Brain and Language (1976)

[9] Naftali, M., Findlater, L.: Accessibility in context: understanding the truly mobile experience of smartphone users with motor impairments. In: Computers & Accessibility. ACM (2014)

[10] Revina, I.M., Emmanuel, W.S.: A survey on human face expression recognition techniques. Journal of King Saud University-Computer and Information Sciences (2021)

[11] Rozado, D., Niu, J., Lochner, M.: Fast human-computer interaction by combining gaze pointing and face gestures. Transactions on Accessible Computing (2017)

[12] Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems (2017)

[13] Zhang, X., Kulkarni, H., Morris, M.R.: Smartphone-based gaze gesture communication for people with motor disabilities. In: Human Factors in Computing Systems. ACM (2017)

[14] Zhao, X., Fan, M., Han, T.: "i don't want people to look at me differently" designing user-defined above-the-neck gestures for people with upper body motor impairments. In: Human Factors in Computing Systems (2022)

[15] Zhong, Y., Raman, T., Burkhardt, C., Biadsy, F., Bigham, J.P.: Justspeak: enabling universal voice control on android. In: Web for All. ACM (2014)