ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 7, July 2023

PIANO：A Hand Training Serious Game

[1] Xiya Tao, [2] Martina Eckert

[1] [2] Research Center for Software Technologies and Multimedia Systems for Sustainability, (CITSEM) of the Universidad Politécnica de Madrid (UPM)
Corresponding Author Email: [1] xiya.t@alumnos.upm.es, [2] martina.eckert@upm.es

Abstract— ***Hands are very important body parts for humans, yet they are often affected by injuries or dysfunctions due to diseases. However, rehabilitation costs are high, and patients are often unable to receive the necessary treatment, so supervised self-training at home could be a good solution. This is possible through serious games that allow patients to exercise at their own pace. Many games have already been proposed and developed over the years, but most rely on additional equipment such as motion capture cameras or robotic devices. In this work, we developed a serious hand rehabilitation game called PAINO, based on open-source software and a regular webcam, to explore new ways of implementing low-cost, yet precise and effective exercises. We tested PIANO for feasibility with four users by analyzing the accuracy of the hand and finger detection, their questionnaire responses, and the emotions experienced while playing. The results indicate that this type of serious game is useful for guiding users towards targeted hand movements and could therefore be used as a tool for self-training of finger flexion.***

Index Terms— ***Finger motion capture, hand rehabilitation, hand detection, serious game, webcam.***

I. INTRODUCTION

Hands are very important body parts for humans because they are constantly in use. This fact makes them very susceptible to various types of injuries, in addition, they are often affected by certain diseases (e.g. Stroke, Parkinson’s, Cerebral Palsy or a cerebrovascular accident). With the development of technology and society, the cure rate for hand injuries is increasing, but after medication and surgery, most patients are still unable to return to their previous work and life due to various after-effects and psychological reasons. However, the effects of depression can be greatly reduced or even eliminated if, in time, the patient starts a scientific and effective rehabilitation program after surgery [1].

The problem is that not everyone can afford expensive postoperative rehabilitation in the hospital or have access to specialized rehabilitation programs in the laboratory. Here, the possibility of using a therapeutic serious game could be a good option for many patients, if self-training is considered being helpful for their recovery [2]. Remotely supervised self-training at home can significantly reduce rehabilitation costs and, because it is readily available, significantly reduce the time required for the rehabilitation process.

Injuries in the hand usually affect control and lead to difficulties in performing activities of daily living. In recent years, multiple virtual reality (VR) systems and robotics combined serious games have been developed for hand rehabilitation. One example is a VR game based on the Kinect Xbox One sensor that used the mean absolute trajectory error and hand speed movement to analyze the arm movement during gaming [3]. There are also hand rehabilitation games for stroke patients, based on a combination of the motion tracking device P5 Glove and Kinect [4]. RobHand (Robot for Hand Rehabilitation) is a system presented by [5], which is an exoskeleton that supports electromyography (EMG) driven assisted bilateral hand control by using a custom-made low-cost EMG real-time embedded solution. The combination of an interactive serious game interface and a hand exoskeleton device also stimulates physical mobility by placing the user in an immersive experience in an interactive virtual environment [6]. Also, a device developed for hand rehabilitation based on Soft Robotics has been proposed combined with a serious game, which provide users with the necessary exercise and guidance [7].

All these and many other works show that great efforts are being made to achieve hand rehabilitation at low cost, but in most cases, users still need to purchase devices such as depth cameras or wearable gloves. In addition, when using wearable devices, customization is likely to be required, a service that is not always available. Most digital rehabilitation systems require accurate real-time monitorization that reflects the true condition of a patient's hand, the progress of the rehabilitation process and the task completion. To achieve this, accurate real-time pose estimation is essential. Therefore, we believe that it is worth to investigate serious games based on open-source algorithms and open-source game engine (Unity) using regular webcams.

This study presents a hand training demo based on OpenCV [8] and MediaPipe [9] for hand detection via a regular web camera, integrated in a simple Unity game that simulates a virtual piano play. With this demo, we want to test the functionality of a low-cost application that captures hand movements without the need for additional devices and analyze its feasibility for finger exercises that could be used

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 7, July 2023

to improve manual dexterity and thus the post-operative mental health of a patient.

The following efforts have been made to achieve our objectives:

1. Comparison of the joint angle calculation accuracy of four currently available algorithms, for their applicability in a hand recognition system.
2. Development of a serious game to restore manual dexterity in patients with limited manual functionality and hand movement deficits.
3. User evaluation based on a questionnaire and the analysis of the emotional reactions during play.

The rest of the article is structured as follows. In section 2 we show the project development process and describe the selection of an adequate gesture recognition algorithm, the design of the game and the integration of the algorithm in the game. Section 3 presents the results of the prototype evaluation and section 4 contains conclusions and future work.

II. MATERIALS AND METHODS

The development of the application was divided into two parts, which were first finding the best hand recognition algorithm, and second the development of the serious game that incorporates the algorithm. Fig. 1 shows the organization and sequential development of the modules.

Visual Studio code (VScode) v.1.71.2 was chosen as runtime environment, with Python v.3.9.13 as programming language for the hand recognition part. The serious game has been implemented in Unity v.2021.3.0f1, with C# as programing language. The following hardware configurations were used for all experiments: x64 Intel(R) Core (TM) i7-10750H CPU@ 2.60GHz 2.59 GHz, Graphics Cards: NVIDIA GeForce RTX 2070 GDDR6 8GB (256 bits), with Windows 10 operating system.

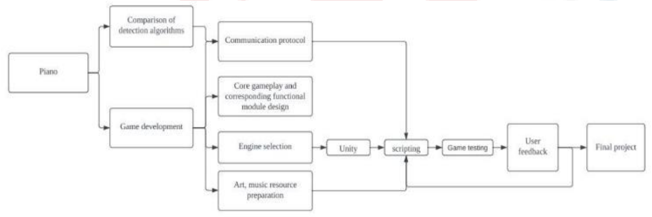

Piano
Comparison of detection algorithms
Game development
Communication protocol
Core gameplay and corresponding functional module design
Engine selection
Art, music resource preparation
Unity
scripting
Game testing
User feedback
Final project

Figure 1. Workflow of the development process

A. Design of the gesture recognition algorithm

Multiple open-source gesture detection algorithms can be found in the literature, so the aim was to find the most suitable one and build the final algorithm on it, adding the necessary modifications and enhancements. For rehabilitation purposes, the most important factor is to achieve an accurate recognition of the finger movements in real time. Therefore, in the first step, we compared four estimation models about their suitability: AlphaPose (v.0.4.0) [10], MediaPipe (v.0.8.9.1), OpenPose (v.1.6.0) [11], and Openpifpaf (v.0.13.2) [12]. We used an English sign language video [13] as a database for these pre-tests. MediaPipe gave excellent results without training the algorithm, Openpifpaf's hand keypoints were in a jump state, and AlphaPose's hand failed to detect finger keypoints without keypoint inference when it was partially occluded. With these factors in mind, we decided to do a closer comparison between MediaPipe and Openpose to find out which one can be used as a basis for developing the hand and finger recognition for this project.

To compare the accuracy of MediaPipe and OpenPose, we used a Chinese sign language video from Fudan University to conduct hypothetical experiments [14].

In order to better analyze the accuracy of the two algorithms for motion detection, we edited the Chinese sign language videos, from which we selected the frontal movements, the lateral movements and the hand-hand overlapping movements to make three short videos, and named them as Video 1, Video 2 and Video 3.

The key parameters to be compared within this project are the following:

- Frontal hand movements
- Lateral hand movements
- Hand overlap detection
- Kinematics analysis (detection accuracy)
- Speed of motion detection

For the comparison of frontal hand movements, we chose video 1, which contains completely unobstructed views of the back of the hands. It shows ten clearly intact fingers, with a few frames of overlapping hands and faces (the similarity of hand and face skin tones may have a slight effect on detection). The detection was performed separately using MediaPipe and OpenPose, and the results were stored in a hand coordinate dataset for further analysis.

For the comparison of **lateral hand movements** with video 2, a one-handed pose was chosen. The palm of the hand is turned sideways towards the camera, with slightly overlapping fingers. MediaPipe and OpenPose were used to detect them separately, and the detection results were also stored in the hand coordinate dataset.

In the comparison of **hand overlap detection**, we used video 3 to detect them by MediaPipe and OpenPose, and output the detection results into hand coordinate dataset for analysis.

Both algorithms also produce output videos that show the markers of the detected keypoints (see Fig. 2, 4, and 6). For the **kinematics** analysis, we used the locations the points detected in video 1 to calculate the joint angles listed in Table 1. All angles obtained have been stored in a database to calculate the root mean square error (RMSE) (1) between both algorithms to analyze their accuracy.

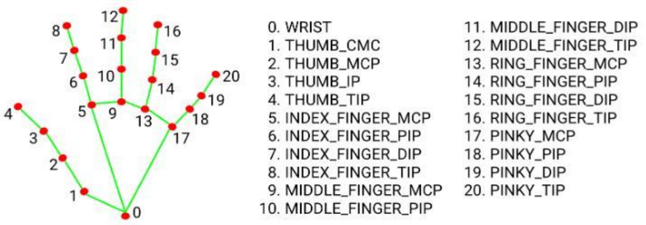


Figure 2. List of 21 hand key points

Table 1: Correspondence between joint angles and key point groups

Angle	Point lists
1	(2,3,4)
2	(6,7,8)
3	(10,11,12)
4	(14,15,16)
5	(18,19,20)
6	(1,2,3)
7	(5,6,7)
8	(9,10,11)
9	(13,14,15)
10	(17,18,19)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}} \quad , \qquad (1)$$

with $X_{obs,i}$ = position of OpenPose and $X_{model,i}$ = position of MediaPipe.

To compare the **motion detection speed**, we compared the frame rates in frames per second (FPS) of the output videos, additionally, motion blur was analyzed as differentiating criteria.

For the **frontal hand movements**, the results showed that OpenPose is capable to detect a hand accurately when it is not overlapping with the face. In case of an overlap, the key points were lost due to the similar skin tones of face and hand. MediaPipe detected both hands accurately without losing any key points and the overlapping part of the hand and face were also accurately distinguished. So, it can be concluded that MediaPipe generally performs better for frontal hand movements.

For the **lateral hand movements**, when not all fingers are visible, OpenPose does not automatically reconstruct the finger positions and produces poor results, while MediaPipe automatically estimates other store positions to build the hand skeleton based on the points that can be detected.

For the **hand overlap detection**, OpenPose lost some or all key points, while MediaPipe worked perfectly in most cases, as shows the example in Fig. 3. Both algorithms present a significant lack of detection in the case of quick hand movements, but OpenPose even performs worse than MediaPipe.

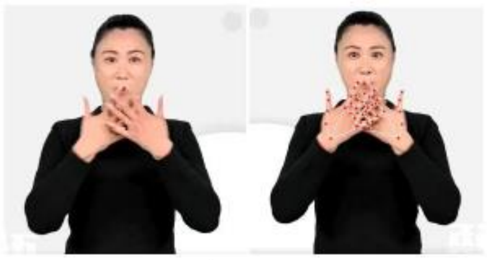


Figure 3. Detection results in the hand-overlap scenario. Left: OpenPose, right: MediaPipe

For **Kinematics analysis**, we compared the average angles detected by both algorithms for different groups of key points.

In Table 2, the results of five joints are presented as examples. The coordinates were cleaned up manually, removing the missing and wrong values, and grouped to calculate the mean value. Joint angle 10 could not be used for comparison, because keypoint 17 was covered most of the time and OpenPose did not accurately infer its location.

As can be seen, the angles are very similar, which also was confirmed by visual inspection of the videos, so MediaPipe and OpenPose are comparable in terms of kinematic analysis. Nevertheless, in frames without a hand present on the screen, OpenPose occasionally detected wrong keypoints, whereas MediaPipe produced no output. In cases when some keypoints were covered, OpenPose was unable to infer the corresponding joints, which had a significant impact on OpenPose's angle detection.

Table 2. Results obtained for selected Joint Angles

Average Joint Angles calculated per group of key points		
Key point group	MediaPipe	OpenPose
(2, 3, 4)	174.29°	179.32°
(6, 7, 8)	174.76°	177.48°
(10, 11, 12)	175.68°	178.53°
(14, 15, 16)	175.42°	179.6°
(18, 19, 20)	177.21°	177.56°

Table 3 shows an overview of the results presented so far, As can be seen, MediaPipe performs overall better in the detection accuracy, while OpenPose often fails to detect key points.

Table 3. Comparison results of hand algorithms

Key Parameters	**OpenPose**	**MediaPipe**
Frontal Hand Movements	Bad	Good
Lateral hand movements	Loses some points	Good
Speed of motion detection	Slow	Fast

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 7, July 2023

Key Parameters	OpenPose	MediaPipe
Hand overlap detection	Loses all points	All good
Kinematics analysis	Loses some points and gets wrong data	Area of the overlapping needs to be less than 60%

Finally, analyzing the motion detection speed, Table 4 shows that MediaPipe completes the detection about 18 times faster than OpenPose for the same number of frames. Also, the frame rates of the produced videos differ significantly See Table 5). OpenPose, calculates only 2 frames per second, while MediaPose produces high frame rates of 30 to 38 FPS.

Table 4. Processing time per test video

Inspection Completion	Video 1 (30 s)	Video 2 (25 s)	Video 3 (27 s)
OpenPose	631.08 s	598.27 s	587.49 s
MediaPipe	34.38 s	29.56 s	32.53 s

Table 5. Average frame rates of output videos

Average frame rate(FPS)	Video 1	Video 2	Video 3
OpenPose	2.1	1.8	1.7
MediaPipe	38.6	36.4	30.7

As an outcome of those different tests, it can be concluded that MediaPipe performs generally better than OpenPose, with quicker and more accurate detection results. In terms of motion detection speed, MediaPipe is about 20 times faster than OpenPose.

From the perspective of the algorithm and the required running environment, the reasons for the poor OpenPose results are most likely the following: On the one hand, insufficient algorithm training in the pre-project period, resulting in low recognition accuracy. On the other hand, the hardware base configuration of the project probably did not reach the standard required by OpenPose. We used only pure CPU to compare the algorithms, while OpenPose runs much faster on the GPU and requires more memory.

Consequently, for the final algorithm, we decided to use MediaPipe for implementation in the Unity game.

B. Implementation of the hand detection algorithm in the Unity game

To correctly capture the hand movements in Unity and make them usable for playing a game, we first need to get the complete information of the hands. Therefore, we added several functions for left- and right-hand differentiation, distance measurement from hand to webcam, as well as measurement of the joint angles. These functions will make the hands generated in Unity more accurate and realistic.

Fig. 4 shows an example of finger flexion detection with MediaPipe, where the numbers represent the angles in each joint.

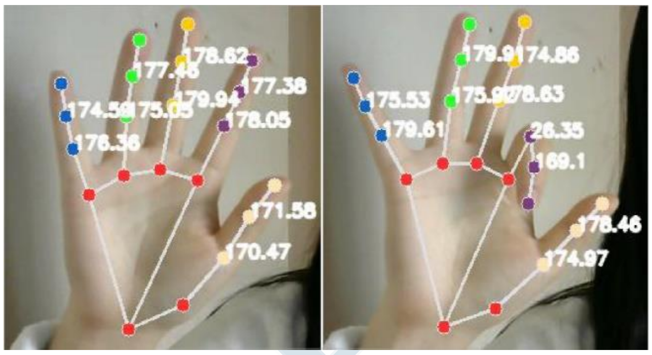
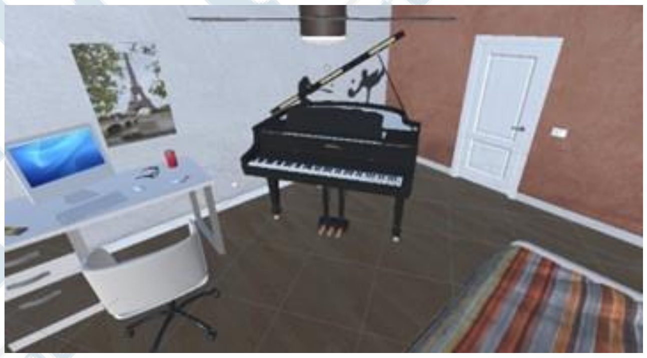
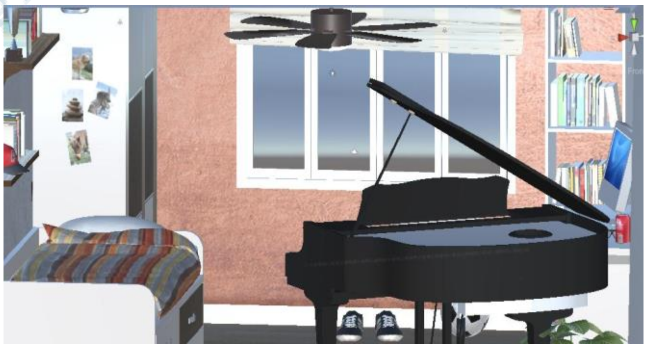


Figure 4. Finger curvature test

For the feasibility testing of the algorithm, a simple piano playing game has been created in Unity, where a virtual 88-key piano, placed in a living room, can be played by moving the fingers up and down in front of the web camera (see Fig. 5).



(a)



(b)

Figure 5. Room environment facing the piano (a) and showing its back view (b).

The piano was created as an entire object with the keys being separate elements to achieve the individual sounds. Fig. 6 shows the basic model that appears when using the game, as well as the hand skeleton that is created in real time. In the lower left part of Fig. 6, a window shows the view of the camera acquisition interface, which is also stored as a data set in order to facilitate the analysis by the therapists.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 7, July 2023

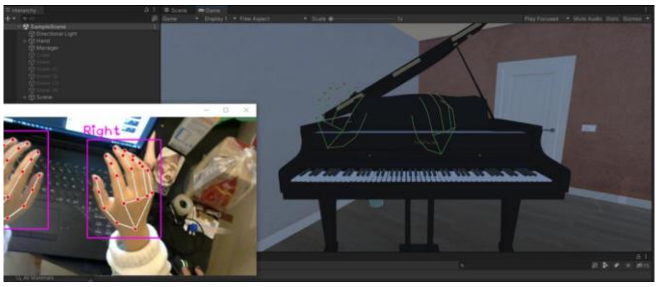


Figure 6. Real-time testing scene of the game

The game detects the collision of the dots representing the fingers (see red dots in Fig. 6) with the black and white piano keys and produces the corresponding sound effect automatically. In view of a possible lack of finger mobility of the patient and to avoid psychological barriers to rehabilitation, there is no need to hit the keys hard to produce the sound effect, the patient simply touches the piano keys with his fingers to get the desired sound. The aim of the game is to improve the patient's finger dexterity as much as possible, so there are no requirements for the intensity of the tapping.

III. RESULTS OF FEASIBILITY TESTS

The Piano game has been tested for feasibility with four abled users between 12 and 40 years with the aim to obtain some subjective feedback and usability results that can help to improve the application and reach a final version that could be tested extensively, also with affected people.

Information about the participants is shown in Table 6.

Table 6. Participants' information

Participant	Age	Gender
P1	12	Male
P2	24	Male
P3	25	Female
P4	40	Male

The participants played the game for 20 minutes and answered afterwards a questionnaire of 20 questions. The questions were divided into four groups of five questions each about comfortability of the interface, ease of play, attractiveness, and overall perception (visual quality, music, experience etc.). Participants answered on a scale from 0 to 10. The average results of the answers per group are listed in Table 7.

Table 7. User Satisfaction Questionnaire

Participant	Interface	Ease	Attractiveness	Overall
P1	8	8	8	8
P2	7	7	8	7
P3	7	5	6	7
P4	8	3	8	5
Average	7.5	5.75	7.5	6.75

During the play, also the facial expressions of the participants have been recorded in parallel, as they give valuable information about the entertainment factor and difficulty of play. These videos have been analyzed afterwards with the software "Emotion recognition v2.1" [15]. As Table 8 shows, most of the time, happy and calm faces were recognized. The eldest participant had some more disgusting moments than the others, which were due to 3D vertigo.

Table 8. User Emotional Feedback

Participant	Happy	Disgusted	Calm
P1	36%	8%	56%
P2	41%	5%	54%
P3	68%	3%	29%
P4	46%	13%	41%
Average	47.75%	29%	45%

IV. CONCLUSIONS

In order to provide a cheap and effective rehabilitation tool for hand and finger exercises, this study achieved the successful implementation of a simple serious game that could be played at home. It has been created with the open-source packages OpenCV, MediaPipe and Unity, and runs on standard personal computers using a regular web camera. This allows access to any user avoiding the need to achieve expensive electronic devices.

The program can record data for each exercise, making it easy for the therapist or family to access the records and analyze whether the patient is making progress in the exercises. The patient's facial expressions are also recorded each time the game is played.

Nevertheless, the application still has some limitations and needs further development. For example, a server is needed to store the data obtained during use in a database. This will make it easier to recall pre-treatment data, data analysis, and user profiling. The project has attempted to do this with local storage, but when the users reached a certain level of usage, the previous data was overwritten, which resulted in data loss.

Preliminary results with four abled users showed that the game can detect the hands and the joint angles of the patient's fingers accurately. The user's found the game entertaining and easy to use, but the small sample size greatly limits the generation of conclusions. The tests did not include therapists nor patients for the time being, so extensive tests must be performed with a future version. This project needs to be supported by more rehabilitation-related researchers and the participation of a variety of users from different backgrounds, which will allow for more accurate user feedback and professional feedback, as well as feedback from users and observations of user behavioral impairments across different disorders and levels, which will allow for more realistic user

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 7, July 2023

needs for new versions and more effective treatments.

In future versions, more features will be added to the game to provide a better a better user experience. For example, some hand training mini games could be integrated to make the game more interesting. A cue bar could be introduced to indicate which key has to be pressed, where the length of the bar can show the duration of the tune. This would break the musical limitations of the game and allow people who can't play piano to play beautiful tunes. The aim is to integrate the game into the Blexer-Med environment [16], which will enable the possibility to adjust the game difficulty and to observe the users results from a distance.

Also, game levels should be created to achieve immersion, motivation, and flow, to reduce the patients' resistance to unassisted rehabilitation and psychological barriers, and to shift patients' attention from pain to enjoying the game [17] to an acceptable level.

REFERENCES

[1] A. Towfighi *et al.*, 'Poststroke Depression: A Scientific Statement for Healthcare Professionals From the American Heart Association/American Stroke Association', *Stroke*, vol. 48, no. 2, Feb. 2017, doi: 10.1161/STR.0000000000000113.

[2] A. Garcia *et al.*, 'A feasibility study to assess the effectiveness of Muvity: A telerehabilitation system for chronic post-stroke subjects', *Journal of Stroke and Cerebrovascular Diseases*, vol. 31, no. 11, p. 106791, Sept. 2022, doi: 10.1016/j.jstrokecerebrovasdis.2022.106791.

[3] B. N. Cahyadi, W. Khairunizam, S. Diny Syarifah, W. A. Mustafa, A. B. Shahriman, and M. R. Zuradzman, 'Rehabilitation Progress of Arm VR Game Based on Hand Trajectory', in *Intelligent Manufacturing and Mechatronics*, M. S. Bahari, A. Harun, Z. Zainal Abidin, R. Hamidon, and S. Zakaria, Eds., in Lecture Notes in Mechanical Engineering. Singapore: Springer Singapore, 2021, pp. 599–607. doi: 10.1007/978-981-16-0866-7_51.

[4] N. J. Seo, J. Arun Kumar, P. Hur, V. Crocher, B. Motawar, and K. Lakshminarayanan, 'Usability evaluation of low-cost virtual reality hand and arm rehabilitation games', *J Rehabil Res Dev*, vol. 53, no. 3, pp. 321–334, 2016, doi: 10.1682/JRRD.2015.03.0045.

[5] A. Cisnal, J. Perez-Turiel, J.-C. Fraile, D. Sierra, and E. de la Fuente, 'RobHand: A Hand Exoskeleton With Real-Time EMG-Driven Embedded Control. Quantifying Hand Gesture Recognition Delays for Bilateral Rehabilitation', *IEEE Access*, vol. 9, pp. 137809–137823, 2021, doi: 10.1109/ACCESS.2021.3118281.

[6] Y. Bouteraa, I. B. Abdallah, and A. M. Elmogy, 'Training of Hand Rehabilitation Using Low Cost Exoskeleton and Vision-Based Game Interface', *J Intell Robot Syst*, vol. 96, no. 1, pp. 31–47, Oct. 2019, doi: 10.1007/s10846-018-0966-6.

[7] A. A. Reymundo, E. M. Munoz, M. Navarro, E. Vela, and H. I. Krebs, 'Hand rehabilitation using Soft-Robotics', in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Singapore, Singapore: IEEE, Jun. 2016, pp. 698–703. doi: 10.1109/BIOROB.2016.7523708.

[8] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, 'A brief introduction to OpenCV'.

[9] C. Lugaresi et al., 'MediaPipe: A Framework for Building Perception Pipelines'. arXiv, Jun. 14, 2019. Accessed: May 12, 2023. [Online]. Available: http://arxiv.org/abs/1906.08172

[10] H.-S. Fang *et al.*, 'AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time'. arXiv, Nov. 07, 2022. Accessed: May 16, 2023. [Online]. Available: http://arxiv.org/abs/2211.03375

[11] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, 'OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields'. arXiv, May 30, 2019. Accessed: May 12, 2023. [Online]. Available: http://arxiv.org/abs/1812.08008

[12] S. Kreiss, L. Bertoni, and A. Alahi, 'OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association'. arXiv, Sep. 21, 2021. Accessed: May 16, 2023. [Online]. Available: http://arxiv.org/abs/2103.02440

[13] *SIGN LANGUAGE FOR BEGINNERS*, (Feb. 19, 2019). [Online Video]. Available: https://www.youtube.com/watch?v=p0ufyoe0URA

[14] *Sign Language Companion 'Chinese Sign Language Tutorial' by Ni Lan - Fudan University*, (May 28, 2021). [Online Video]. Available: https://www.youtube.com/watch?v=VmURfc7PUhE&list=RDCMUC4FsUmSz5jFYIlhmelCsMOA&start_radio=1&rv=VmURfc7PUhE&t=3

[15] 'Introduction to facial expression recognition system - python implementation'. https://www.cnblogs.com/sixuwuxian/p/16163014.html

[16] M. Eckert et al., "The Blexer system – Adaptive full play therapeutic exergames with web-based supervision for people with motor dysfunctionalities," EAI Endorsed Transactions on Serious Games, vol. 5, no. 16, 2018. doi: 10.4108/eai.13-7-2018.155085.

[17] E. S. A. Majid, J. A. Garcia, A. I. Nordin, and W. L. Raffe, 'Staying Motivated During Difficult Times: A Snapshot of Serious Games for Paediatric Cancer Patients', *IEEE Trans. Games*, vol. 12, no. 4, pp. 367–375, Dec. 2020, doi: 10.1109/TG.2020.3039974.