

Musculoskeletal Rehabilitation Guiding System

박세현
장지수
반지훈

1 Introduction

WHO announced that there are approximately 1.71 billion people having musculoskeletal conditions worldwide in 2021 [1]. The number of musculoskeletal patients, from severe cases to mild cases, is increasing enormously year by year. Patients need to visit doctors regularly for the treatment of diseases which are physiotherapy, exercise programs, chiropractic care, and more. This is a huge pressure for patients, especially for mild cases, in terms of cost and time. There has been unmet demand from those patients for a personalized in-home exercise program that does not require any care from doctors. With an elevating order on personalized healthcare systems, many researchers aim to provide a user space-free, time-free, and doctor-free system in any circumstance. However, it is a challenging task as the system needs to be robust and flexible in potential uncertainties. This project aims to build an in-home musculoskeletal exercise guidance system for users by providing a video of the movement with a skeleton so that the user can keep track of their movement. This report covers related works on personalized exercise guidance systems, the reasoning process of our program, and how this can be implemented on the user's side.

2 Related Works

Several ventures are carrying on a business to provide personalized healthcare applications to patients. Kaia Health is a leading digital therapeutics company that gives guidance on musculoskeletal conditions using artificial intelligence (AI). It has a movement coach using AI to support the movement and give real-time analysis to judge whether the movement is on the right track. The coach ensures tracking a skeleton of the patient, however, the technology focuses on the regional part of the body motion, which cannot give movement-specific feedback. Moreover, as this is certified as a medical device, it needs a medical prescription or doctor's assurance.

On top of the ventures, there has been a steady effort on implementing smart devices like smartwatches and smart sensors for using personalized exercise guidance applications as using the camera is not enough to detect the exact movement of users. Min Hun Lee et al., researched robotic coaches for improved engagement of patients on rehabilitation exercises [2]. As such, a number of research focusing on the usage of such devices with a rehabilitation program. Kaelin et al. pointed out that there are gaps between current research and development. One of them is a lack of intervention or guidance on rehabilitation. This means a patient needs ground truth that they can follow to take the correct movement of the

exercise [3]. The key deficiency in this field is that the user needs casual exercise guidance with ground truth that they can refer to. Moreover, as most people do not possess devices like sensors or smartwatches, casual applications like our project are not appropriate for the use of such devices. Therefore, we decided to establish an application that users can use via smartphone or laptop providing a ground truth video for reference and a skeleton of ground truth and user so that the user can keep track of their own movement to see if they are doing it in the right way.

3 Implementation Methodology

3.1 Scenario: Concept of Architecture and process

The initial plan to implement the musculoskeletal rehabilitation guiding system is building an application in real-time that shows an example video of the correct posture and a user camera surface view. An example video of the correct posture is to learn the data from learning UI-PRMD data sets. The goal of training the UI-PRMD data sets is to make GT(Ground Truth) of targeting skeleton landmarks, and angles. Based on these joint coordinates, it tells the user if the action is correct or not. From the user's point of view which is using a camera surface view, since the correct movement is shown as an example with GT, following the guide user can continue to see what the correct posture of the movement is while performing the movement. The application programming interface of building the camera surface is named CameraX. The function of CameraX is used by providing a preview function and a function that allows smooth access to the buffer for use in algorithms, such as ML kit delivery in image analysis. One of the goals to build this structure is making the system real-time. The requirement for building the real-time system is fps (frames per seconds) which is as fast as possible for following the fps of the existing guide video. Hence, we are determined to use OpenGL ES API and Mediapipe pose detection. For the real-time performance of the entire ML pipeline consisting of pose detection and tracking models, MediaPipe proceeds by expanding the blaze face, as each component must be very fast, using only a few milliseconds per frame. Our system of architecture is that image frames brought to CameraX are modeled by Mediapipe pose detection and represented by extracting target points using the OpenGL ES, a type of API. The OpenGL ES is a type of Graphics API for embedded systems, and most of the functions are physically implemented on GPUs. It is difficult to quickly depict the same level of the frame for seconds as the game fps. However, using the OpenGL ES API, describes the same level of graphics as games on the screen fast (30~ 60 fps). Each extracting an angle of target coordinate points from the guide video and camera surface view is compared. As the comparison angle between the GT video and the result of prediction from modeling by user surface view is matched, the result of the accuracy is getting increased.

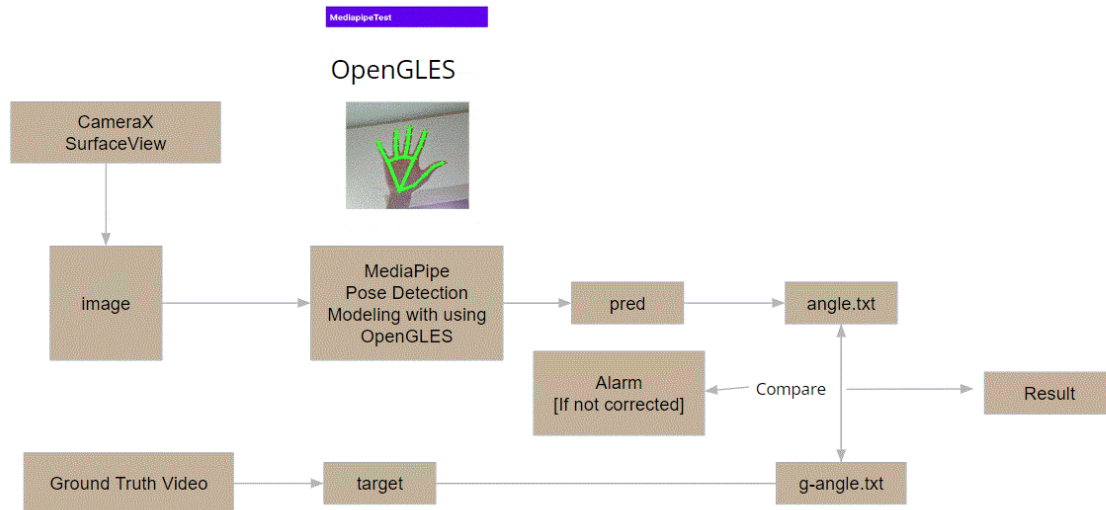


Fig.[1] The Pipeline of the Rehabilitation Treatment Guide System

Overall, the process proceeds to see a short video of the guide of exercise when a user enters the frame for rehabilitation exercise. After watching the guide, it gives five seconds to prepare to follow the movement. As a user follows the movement in the frame, the real-time user frame with joint coordinates and the video frame with GT are compared. After evaluation by frame, the application shows the result.

3.2 The difficulty in building a mobile system

Since the architecture of building the mobile application system is made, the challenge to implement the mobile system is complicated. The main reason is the synchronization of CameraX API and Mediapipe ML kit. The process of setting up the environment to Import the CameraX was fastidious. In the process of setting up the environment, open the build.gradle file for the CameraX module and add the CameraX dependencies, compile options and build features. The building gradle file for adding the CameraX module is not working as importing the Mediapipe from cloning the GitHub. This means the function of CameraX does not work to synchronize the MediaPipe ML kit. In addition to this environmental setting problem, if the Media Pipe ML kit is imported from GitHub, it cannot be updated into the Media Pipe pose detection that we modified. Consequently, given time to master android technology, we will be able to solve the above technical problems. When we felt a technical barrier to building the application we mentioned above, we converted the mobile system to a web-based system.

4 Implementation

We implemented our system using the Streamlit and OpenCV packages in Python. Since it is a web-based system, any camera can be used. Here, we tested on MacBook Pro (15 inch, 2017) with 2.8GHz Quad Core Intel Core i7 CPU for the inference of the MediaPipe post detection and 720p FaceTime HD camera to receive the user's pose.

The following sections are the web-based system process.

4.1 START

The user will first see the instruction video of the exercise. After it, he/she will do a short movement video repeatedly (roughly 3 seconds). We show the user's live video on the left, and the instruction video on the right. However, the instruction video is hidden on the first, we start the FIT phase first and then run the Evaluation phase to see whether the user is doing the right job.

4.2 FIT Phase

When the user clicks on the "START A SET" button, he/she will see a red frame bounding box(called it FIT box) on the real-time video captured from the camera. The FIT box is the instructor's position bounding box in the movement short video, which can be extracted by the MediaPipe pose estimation segmentation results. We draw the bounding box with the minimum top-left and maximum bottom-right coordinates.

The purpose of the FIT box is to let the user prepare to do the movement. If the user is in the bounding box in a certain amount of frames, it will turn green, then the movement instruction video will start automatically by showing the short movement video with inference post-estimation.

So, the MediaPipe program is also running on the user's video inputs at this phase to judge whether the user is in the FIT box. We used three landmark points(elbow, shoulder, hip) to judge it. If these points are in the FIT box in a certain number of frames sequentially then we judge whether the user is ready to go.

When the user is ready, we show an intermediate video on the right side of the web page and count 3 seconds.

4.3 Evaluation Phase

At the evaluation stage, the user's video and the instruction video will be shown on the screen by a certain inference frequency(IF) as a hyperparameter. We infer both video frames with the MediaPipe pose estimation in a certain interval frequency value. If $IF=1$, it means inference each frame in a certain fps video. For example, in our demo, both videos have 30 fps, and if $IF=1$, we run the model 30 times in a second. However, the lower IF will lead the video looking dramatically slow. When it is too high, the video will show in a fragile way, since we only show part of the video frames to the user.

Using user's and instruction frames, we extract the elbow, shoulder, and hip coordinates to calculate the angles among these points. If the difference in angles between the user and instructor is lower than certain criteria, we treat this frame as a correct frame. We measure the matching accuracy as the sum of correct frames divided by the total number of frames.

Our demo shows an average of 1 to 3 FPS in the FIT phase and nearly 0 FPS in the Evaluation phase. One of the reasons is that inferring two frames at the same time requires a lot of computation in CPU environments. We tried to pre-calculate the instructor's video first and only infer the user's movement after that. We can save some computation time this way, but we still need to add matching frames between the user and the instructor to determine whether they are making the same semantic movement.

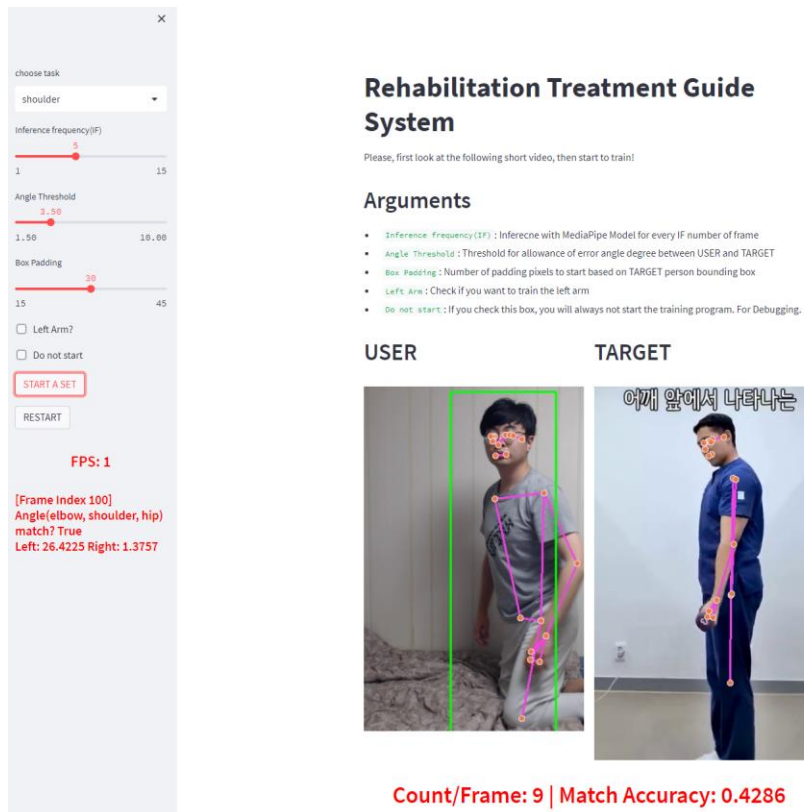


Fig.[2] The Interface of the Rehabilitation Treatment Guide System:
USER is the current user and TARGET is the instructor.

Fig. [2] shows our interfaces, there are several parameters that can be determined:

- Inference frequency(IF): Inference with MediaPipe Model for every IF number of frames
- Angle Threshold: Threshold for allowance of error angle degree between USER and TARGET(instructor)
- Box Padding: Number of padding pixels to start based on TARGET(instructor) person bounding box
- Left Arm: Check if you want to train the left arm

Conclusion

We switched to a web-based system and successfully created a rehabilitation exercise guide system in real time. However, the performance of implementing the system in real-time is limited to us. The limitation is that if the process is performed systematically in real-time, the matching algorithm may be inaccurate in the case of a delay. Moreover, since it is difficult for the patient to follow the guide movement well, we consider that it is necessary to add a function to perform one movement and analyze the image. As a result, our future plan, including the above we mentioned adding function, is to build MediaPipe pose detection with CameraX and OpenGL ES surface view in Android for performing better in real-time. Besides, we will consider reducing latency when running a real-time application in order to improve Android performance.

Citation

[1] "Musculoskeletal health," *World Health Organization*. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/musculoskeletal-conditions>. [Accessed: 17-Dec-2022].

[2] Lee, M.H., Siewiorek, D.P., Smailagic, A. *et al.* Enabling AI and Robotic Coaches for Physical Rehabilitation Therapy: Iterative Design and Evaluation with Therapists and Post-stroke Survivors. *Int J of Soc Robotics* (2022). <https://doi.org/10.1007/s12369-022-00883-0>

[3] Kaelin VC, Valizadeh M, Salgado Z, Parde N, Khetani MA. Artificial Intelligence in Rehabilitation Targeting the Participation of Children and Youth With Disabilities: Scoping Review. *J Med Internet Res*. 2021 Nov 4;23(11):e25745. doi: 10.2196/25745. PMID: 34734833; PMCID: PMC8603165.