A hierarchical privacy-preserving IoT architecture for vision-based hand rehabilitation assessment

Ali Nadian-Ghomsheh1 • Bahar Farahani1 • Mohammad Kavian1

Received: 15 September 2020 / Revised: 12 December 2020 / Accepted: 13 January 2021
Published online: 13 February 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract
The healthcare industry requires the integration of digital technologies, such as Artificial Intelligence (AI) and the Internet of Things (IoT), to their full potential, particularly during this challenging time and the recent outbreak of the COVID-19 pandemic, which resulted in the disruptions in healthcare delivery, service operations, and shortage of healthcare personnel. However, every opportunity has barriers and bumps, and when it comes to IoT healthcare, data privacy is one of the main growing issues. Despite the recent advances in the development of IoT healthcare architectures, most of them are invasive for the data subjects. In this context, the broad applications of AI in the IoT domain have also been hindered by emerging strict legal and ethical requirements to protect individual privacy. Camera-based solutions that monitor human subjects in everyday settings, e.g., for Online Range of Motion (ROM) detection, are making this problem even worse. One actively practiced branch of such solutions is telerehabilitation, which provides remote solutions for the physically impaired to regain their strength and get back to their normal daily routines. The process usually involves transmitting video/images from the patient performing rehabilitation exercises and applying Machine Learning (ML) techniques to extract meaningful information to help therapists devise further treatment plans. Thereby, real-time measurement and assessment of rehabilitation exercises in a reliable, accurate, and Privacy-Preserving manner is imperative. To address the privacy issue of existing solutions, this paper proposes a holistic Privacy-Preserving (PP) hierarchical IoT solution that simultaneously addresses the utilization of AI-driven IoT and the demands for data protection. Furthermore, the efficiency of the proposed architecture is demonstrated by a novel machine learning-based system that allows immediate assessment and extraction of ROM as the critical information for analyzing the progress of patients.

Ali Nadian-Ghomsheh
a_nadian@sbu.ac.ir

Bahar Farahani
b_farahani@sbu.ac.ir

Extended author information available on the last page of the article
Keywords  Internet of things · Machine learning · Privacy-preserving · Range of motion measurement · Physical rehabilitation

1 Introduction

Healthcare and technology have a long history of interaction. However, the rapid adoption and evolution of the Internet of Things (IoT), the development of wearable biosensors, as well as technological advancements in Artificial Intelligence (AI) have created new opportunities in the realm of smart and personalized healthcare. By tackling conventional healthcare obstacles, IoT healthcare enables a paradigm shift toward patient-centric care offering revolutionary benefits such as increased accessibility, availability, customization, cost efficiency, and precision. Despite this transformation, many challenging problems remain that will require more scientific breakthroughs. Particularly, IoT healthcare produces massive amounts of data around individual behavior and activities. Transmitted such data into the cloud through non-secure channels or processing and storing privacy-sensitive patient’s data in a centralized location generates many privacy concerns and pitfalls that will need to be addressed. In 2019, the Global Risk Report of the World Economic Forum (WEF) indicated data breaches and cyber-attacks are the fourth and fifth more prevalent risks facing the technology world. The healthcare arena, among others, is the target of many cyber-attacks, and many scenarios can create fertile ground for such attacks. In addition, centralized training creates another set of challenges for traditional cloud-based IoT computing. Indeed, the Variety, Volume, and Velocity (3Vs) of data coming in from billions of devices need huge amounts of bandwidth, which can be very expensive. In addition, IoT healthcare applications often need near real-time analytics and decision-making abilities or low latency [10, 12].

Physical rehabilitation addresses the illnesses or injuries that limit a person’s abilities to move and perform functional activities in their daily lives. Rehabilitation over the Internet, regarded as a branch of Tele-Rehabilitation (TR), allowing patients to interact with providers remotely and can be used both to assess patients and to deliver therapy. A recent literature study covering more than a thousand patients that engaged TR showed that the approach is less costly and equally as effective as clinic-based rehabilitation at improving functional outcomes in stroke patients [21]. Motor training exercises are the most commonly implemented modality, where the provider guides a patient through performing different motions and activities in order to regain strength and function. Performance analysis allows therapists to assess the current improvements and devise future rehabilitation plans. The quantitative method for assessment is the Range of Motion (ROM) measurement. ROM is a measure of movement flexibility around a specific joint or a body part. Computer Vision methods powered by AI enable cost-efficient methods for ROM measurement [20, 28]. In such applications, machine learning (ML) models are trained to learn patient’s treatment exercises and calculate the ROM for different joints. Resulted information can then be used to monitor patient progress and enable performance analysis.

Computer vision methods can be extended through IoT architectures to provide TR solutions. However, training a machine to learn in a distributed environment such as an IoT architecture faces some limitations. Data for training should be collected from institutions and individuals which requires dealing with different storage and data privacy regularizations [44]. In addition, for the particular job at hand, data is a collection of images that should be captured from patients while practicing physical activities in their private living environment. Personalization is another important issue that should be
considered when users with different physical characteristics use the TR system and the learned model needs to be updated in order to meet individuals’ requirements. Hence, preserving the privacy of users and enabling personalization becomes an important factor in an AI-driven IoT Healthcare system for physical TR.

To address the above-listed shortcomings of the state-of-the-art techniques, in this paper, we propose a holistic IoT-based ROM assessment solution. The main contributions of our work include:

- A hierarchical edge-assisted IoT architecture to recognize and assess rehabilitation exercises.
- A complementary set of Privacy-Preserving (PP) techniques, e.g., Federated Learning, to train the machine learning models over privacy-sensitive data based on the proposed IoT architecture.
- A marker based method for visual recognition and measurement of ROM.

The rest of the paper is organized as follows: In Section 2, ROM measurement methods are reviewed, and the idea of the proposed ROM measurement approach is explained. In Section 3, Privacy-Preserving machine learning and multiparty computation methods are reviewed. The proposed IoT ecosystem and the ROM measurement methods are explained in Section 4. Experimental results and discussion are provided in Section 5, and the paper is concluded in Section 6.

### 2 Range of motion for monitoring physical rehabilitation

Current studies have shown promising results for applying computer vision techniques for improving physical TR through patient performance monitoring. However, several important issues, such as designing ML algorithms, privacy issues, and integrating the solution in a communication medium such as IoT, require further research. In this section, we explore conventional and recent computer vision-based methods for ROM measurement and discuss their major limitations.

Stroke, physical injury, and surgery may cause decreased strength and dexterity of the hand. Hand exercises can be regarded as an essential part of a rehabilitation process to help those in suffering reclaim strength and dexterity. Exercise therapy can be categorized objectively as goal-oriented, task-oriented, and repetitive task training. Task-oriented training is defined as single-joint and single plane movement exercises [41]. Repetitive task training refers to an active sequence performed repetitively within a single training session [43]. Goal-oriented exercises are designed based on performance analysis and identification of functional limiting variables [26]. Duration, exercise load, and feedback can be further used to characterize each method.

In the rehabilitation process, rapid performance monitoring and analysis are essential parameters that help create effective treatment plans. Patient’s performance is analyzed by quantifying the flexibility of the body in terms of angular displacement of the joints defined as ROM. Goniometry [13] is the conventional method practiced to obtain ROM values. Goniometry in health care is a technique to measure the position and range of motion of a desired joint in each of the three spatial planes [42]. The process involves placing an appropriate goniometer on the measured angle and reading the flexibility of the joint. Despite the popularity, the method has several drawbacks, including significant time consumption [6, 9], and being intrusive [5, 7, 14, 27]. However, Goniometry is a reference method for measuring ROM and is currently used in
rehabilitation clinics. Other mechanical devices used for ROM measurement include inclinometer [25], electro-goniometer [15, 37, 40], and wearable sensors [45]. These devices provide reasonable accuracy for ROM measurement, however share common disadvantages; namely, they are expensive, intrusive, time-consuming, and require high maintenance. Marker-based motion-recording technology has also been used for ROM measurement. Motion capture techniques are used over a very broad field of applications, ranging from digital animation for entertainment to biomechanics analysis for clinical and sports applications [11]. Although this is an accurate approach, it is very expensive and impractical, especially for TR [30].

Computer vision applications in physical rehabilitation have been recently considered in several studies. For ROM analysis, most studies are focused on gate rehabilitation. The progress is mainly due to the advent of marker-less motion capture systems, especially the Microsoft Kinect camera, which comes with a Software Development Kit (SDK) that allows application development with the Kinect joints stream [39]. This technology applies an infrared imaging system to capture the scene depth and implements a pattern recognition algorithm to extract 20 landmarks on the body for Kinect v1 (or 25 for Kinect v2), regarded as Kinect joints. The relative angle between joints is used to extract various body angles. Several studies have confirmed the accuracy of Kinect against Goniometry and other standard devices for ROM measurement. The authors studied the accuracy of Kinect for hand ROM, and the results showed there is a low correlation between Kinect reading for hand ROM measurement. This is mostly due to the fact that Kinect was not designed for the extraction of hand joints exclusively [29, 35]. Marker-less hand motion capture technology has been less considered for hand rehabilitation. This is mainly due to the fact there are a few off-the-shelf solutions for the task. The Leap Motion device is one of the very few devices that is built to extract hand joint information through computer vision approaches. In a recent study, the authors investigated the validity of Leap Motion data for hand ROM measurement [20]. The result of this study showed that the error of ROM obtained from the Leap Motion is significant. The main reason for the low accuracy of the device was due to the complex structure of the hand and where the detected joints did not match with the real position of the joints on the hand.

Another critical challenge of the state-of-the-art ROM assessment technique is that they are very invasive to the privacy of users. Privacy and confidentiality are the central components of the healthcare industry and they should be adequately addressed. However, training an accurate and reliable ROM assessment technique requires retrieving hundreds of thousands image data from several patients locating in multiple sites with different regulations on the disclosure of health information. For instance, in a real-life scenario, images are typically captured by a camera in the patient’s private living area which includes significant amount of privacy-sensitive information. This problem is even compounded by uploading and storing all data in remote cloud servers which are susceptible to various cyber-security concerns, such as lack of transparency. Unfortunately, to the best of our knowledge, none of the previous work tackled this important issue.

To overcome the problems of marker-less motion capture systems for hand joints localization, a marker-based solution was considered. In this scheme, markers that are stuck on the hand should be recognized by, and then they should be identified to be used for ROM measurement. The markers can be any kind of an indicator with colors that are distinguishable from the color of the skin. Our computer vision algorithms extract the hand region and locate markers within the hand region. From marker information, the ROM values for several joints can be calculated. The method proposed here can work with conventional RGB cameras and, unlike previous methods, does not require expensive depth cameras. Based on our observations, several challenges in the proposed vision-based method for wrist and fingers ROM
analysis can be pointed out. Background colors can be misclassified as hand joints, and hence unwanted markers are detected. To solve this problem, we consider segmenting the hand region via human skin detection. The detected markers should be recognized in terms of the joint they represent. Several approaches using machine learning and data-driven techniques were presented and tested. The best solution for marker recognition was applied. To obtain ROM for a specific joint, the joint should be engaged in a specified rehabilitation exercise. Any measurement in other forms of motion will lead to undesired ROM values. In addition, to address privacy issues of the today’s techniques, images during the training as well as the inference phase, are remained on the local devices. We utilize a parameter mixture model (Gaussian Mixture Model) to obtain the human skin distribution. The parameters of the model are learned through Federated Learning (FL) for preserving the privacy of the users.

3 Privacy-preserving machine learning and multiparty computation

Privacy-Preserving Machine Learning (PPML) is a subcategory of ML that focuses on training models while safeguarding data privacy. Privacy-Preserving techniques enable several input parties to work together to train an ML model without providing original data that including private information. In addition, Multiparty Computation (MPC) empowers organizations to analyze private data owned by other institutions without revealing inputs. Therefore, MPC is a technique that analyzes data and completes computations using inputs from multiple parties. No single party receives extra information about another party’s inputs. Everyone receives the same public output available to all parties. In this section, we briefly review different PPML techniques.

Secret Sharing was introduced by Shamir and Blakely in 1979. Secret Sharing means that one party distributes a secret to all other parties in such a manner so that none of the other parties can obtain the whole secret alone. The secret is shared so that the information of at least \( t \) (\( t \) is a preset threshold) of the \( n \) parties is required to obtain the complete secret. Put simply, as depicted in Fig. 1a, a party breaks the secret into pieces and disperses a piece to various parties. Each party performs computations individually, having never seen the complete input values. In the end, the combined results reveal the complete, correct output [36].

Suppose that three parties, Bob, Alice, and Carol, would like to collaboratively calculate the summation of their salaries in a Privacy-Preserving manner; meaning that none of them should be able to find out the other’s salary. Assume that \( x, y, \) and \( z \) are the values of Bob’s, Alice’s, and Carol’s salary, respectively. In this case, the salary of Bob (\( x \)) is divided into three parts of \( x_1 + x_2 + x_3 \). He keeps \( x_1 \) for himself and shares \( x_2 \) with Alice and \( x_3 \) with Carol. Indeed, neither one knows the salary of Bob. Similarly, Alice and Carol conduct the same procedure and share only a part of their salary with others. In the next step, each party calculates the summation of numbers in her/his hand. In this example, Bob determines public1 = \( x_1 + y_1 + z_1 \), Alice calculates public2 = \( x_2 + y_2 + z_2 \), and Carol performs public3 = \( x_3 + y_3 + z_3 \). Then, each participant sends her/his public answer to the other parties. Finally, each party recognizes the summation of the salaries by performing public1 + public2 + public3. This example shows that in the secret sharing approach, computations are completed at the local level, and none of the parties has access to complete inputs.

Secret sharing technique suffers from several shortcomings such as long computation time due to multiple back-and-forth operations between participants, no support for categorical data,
and low accuracy for complex operations such as non-linear functions due to their approximated implementation.

**Homomorphic Encryption** (HE), a centralized ML technique, enables computation with ciphertexts. It generates results that can be decrypted to reveal results that match those that would have been generated if the operation had been completed using plaintext. As illustrated in Fig. 1b. Contributors encrypt raw data before sending it to a server for analysis. Operations are completed using encrypted data, and results are encrypted using the same key. The data does not have to be decrypted prior to analysis [1, 3, 22].

**Differential Privacy** (DP) is a technique initially developed by Cynthia Dwork [17]. DP safeguards user privacy by incorporating random noise with data (i.e., adding years to your age when completing a survey). This makes it impossible to determine a user’s identity. DP is innovative because it protects privacy while simultaneously permitting thoughtful data analysis. While adding noise protects individual data, when the data points are aggregated, the noise is averaged, revealing a result nearer to the original (See Fig. 1c).

For example, if a survey includes controversial yes/no questions, then participant data should not be shared. DP can be implemented using a few easy steps. Simply flip a coin. If the coin lands on heads, then document the original answer, or flip the coin again. If it lands on heads, record "yes" or "no." Flipping the coin ensures responses are randomized and incorporates noise to ensure data privacy. At a basic level, this trades accuracy for privacy. If the probability that the coin will land on heads is 1, then the original response is always documented. No noise in the data means that the data’s accuracy will be high, but the data will lack privacy. If the probability of landing on heads is 0, then responses are randomized, and the data noise will be maximized, but the data is not accurate enough to be useful. Researchers must choose the probability value that secures privacy and maintains an acceptable level of data accuracy.

---

Fig. 1 Privacy-preserving machine learning and multi-party computation techniques. a Secret sharing, b Homomorphic encryption, c Differential privacy, d Federated learning
Generally, DP systems can work using local privacy or global privacy. A globally private system requires that a trusted party (i.e., curator) have access to the raw data of several people. This party then analyzes the data and adds noise after analysis. Locally private systems do not contain a trusted party. Every party is responsible for including noise with data before sharing. Locally private systems usually include an aggregator responsible for gathering data from a large group simultaneously. Global private systems often have a higher level of accuracy because clean data is analyzed, and minimal noise is added at the end. For such a system to be successful, all parties must trust the curator. Local privacy models are safer because all data points are noisy, making the individual data points useless. Data noise can be removed, and aggregators that gather the local data can analyze trends within the entire dataset [2, 8].

**Federated Learning** (FL), also known as collaborative learning, is a decentralized ML technique originally created by Google. It is an instance of bringing code to data rather than bringing data to the code. It depends on a central server as well as multiple client servers and offers the advantage of keeping private data in its original location. As demonstrated in Fig. 1.d, FL uses a central server to train the model using proxy data that is made available in advance. The initial model is then provided to clients for localized training. Individual clients use local data to train the model for multiple iterations. Then, each contributor receives a copy of the model’s parameters, trains it with private data, and sends the computed results of the model back to the centralized server. The coefficient/gradient information received from the clients is averaged by the central server to create a global model. The updated model is then returned to client servers. These steps are repeated across multiple iterations until the level of accuracy reaches an appropriate level. In general, there are two types of Federated Learning: Horizontal Federated Learning and Vertical Federated Learning [4, 23, 47].

- **Horizontal Federated Learning** – Also known as sample-based FL, this type is used in situations where datasets have identical feature space but have differing space within samples. For example, two banks may have different customer groups based on region, so the intersection of users is small. But, because both are similar businesses, the feature spaces are the same. Participating parties must have the whole feature set and all labels available to train a local model. The horizontal model recognizes shared feature space and amalgamates the model’s results. The subsequent model is comparable to stacking datasets with similar features spaces to train one model. For example, one pharmacy chain may have stores on the coasts, while another pharmacy chain has stores located primarily in the center of the country. This results in very diverse user bases, but all customers are still shoppers of pharmacy. Both pharmacy chains gather similar data about purchases; therefore, it is possible to lessen geographic bias by combining the data. FL can be utilized to train models and expand understanding of customer needs and desires.

- **Vertical Federated Learning** – Also known as Feature-Based Federated Learning, is useful in situations where datasets have the same sample ID space but different feature spaces. For example, an eCommerce company and a bank in the same city will have datasets containing many of the area’s residents, so their user spaces largely overlap. However, the bank’s data is focused on customer data like credit ratings, spending habits, and saving patterns, while the eCommerce company’s data focuses on customer purchase or browsing information. This means that the two companies’ feature spaces are very different. Vertical Federated Learning enables the creation of a purchase prediction model based on product and user data. Vertical Federated Learning allows different parties to gather data with diverse feature spaces, but only one party can access the label so that
others cannot train a model independently. This makes the aggregator more complex because sample spaces have to be aligned. Training the model also requires that partial model updates be exchanged. For example, if an eCommerce company and a credit card company both wanted to improve a fraud detection algorithm, vertical Federated Learning enables both organizations to develop a single feature vector to train the model while improving security for users.

4 Proposed solution

In this section, the proposed method is explained in detail. First, the IoT architecture for the rehabilitation process is presented. Then, the computer vision process responsible for assessing the rehabilitation exercise is presented.

4.1 IoT architecture

The proposed Privacy-Preserving hierarchical IoT eHealth architecture and its main building blocks, an extension of our recent works published in [10] is illustrated in Fig. 2. The components of this architecture are organized in a distributed manner across the following three layers:

1. Edge Device Layer: This layer comprises both wearable devices (e.g., smart wristband) and ambient sensors (e.g., camera) connected to the network either by wireless/wired technologies providing a 360-degree view of individuals.

![Fig. 2 Proposed hierarchical privacy-preserving IoT architecture for ROM measurement](image)
2. Fog Layer: The fog layer is composed of several geographically distributed smart gateways. The main role of this layer is to bridge the gap between the endpoint IoT devices and the cloud by enhancing the performance of the system, such as interoperability, context-awareness, energy-efficiency as well as latency by locating data storage and computing resources closer to the data-producing sources.

3. Cloud Layer: The convergence of IoT and Cloud computing offers i) higher processing capabilities for complex data analytics and machine learning; ii) unlimited storage to tackle big data (characterized with 4Vs: high volume, high variety, high veracity, and high velocity) generated by IoT devices; iii) novel “Things as a Service” business model, and iv) a cost-effective solution to connect, monitor, and manage any eHealth IoT devices from any location at any time.

One of the main important features of the hierarchical IoT healthcare architectures is the edge-fog-cloud interplay in a way that the intelligence and data processing tasks can be distributed across the edge, fog, and cloud. Particularly, task allocation and computation offloading techniques can be utilized to optimize problems, such as application execution, energy consumption, and bandwidth by transferring resource-intensive tasks to another computing node (e.g., fog nodes or cloud servers) considering the current dynamic of the system in real-time. To find the best task, time, and location of task migration, offloading techniques can take advantage of reinforcement learning, deep learning, genetic algorithms, and game theories, to name a few.

As mentioned in the prior sections, the public embracement of healthcare IoT devices, particularly connected cameras, is largely debated in light of data privacy, ethical, and legal considerations. To address this important challenge, we propose to incorporate and map the Federated Learning techniques with the aforementioned hierarchical architectures. Thereby, data collected by IoT devices do not need to leave its original location. The Federated Learning approach can be divided into the following steps: 1) An initial ML model can be trained at the cloud server using publicly available datasets. 2) Next, the model will be transferred to the local nodes. Note that local nodes can be either endpoint IoT devices owned by a patient or even to a fog node, e.g., located in a hospital or at home trusted by several patients. Each local processing node is responsible for training the model individually by its local data and then sends the trained parameters (e.g., the weights of the deep learning models or coefficients of the ML model) to the server. 3) The server is responsible for combining all trained parameters received from all local nodes into one model and then send it back to local nodes. This procedure is iteratively executed until the model converged. Federated Learning by distributing the model training across several local nodes enables multiple parties/patients to jointly train a shared model while preserving the privacy of the data owner. Although integrating the Federated Learning eliminated the need for sharing raw data, in the end, all ML parameters should be collected and combined in a central server. Recently, a few studies demonstrated that malicious parties could attack the privacy of owners by analyzing the collected ML coefficient and reverse engineering to gain insight into local raw data. To mitigate this threat, differential privacy or secret sharing techniques can be integrated to perform Step 3 of the Federated Learning in a more secure manner. In this way, there is no need to send the plain/raw ML parameters from local nodes directly to the server, and as a result, it leads to higher security and improved privacy. FL can also be combined with DP in a way that each client injects a certain amount of noise to the locally trained parameter before uploading them to the central server.
4.2 Vision-based ROM measurement

In this section, the ML engine responsible for wrist and fingers ROM analysis is explained. The process consists of several computer vision and ML tasks that are trained jointly on the edge device and the cloud server to preserve the privacy of the user. The objective of the computer vision system is to locate the position of the markers on the hand via personalized skin detection and Federated Learning. The ROM for a specific joint can be obtained by using the position of three markers. These markers consist of the rotating joint and two other markers chosen by definition of the specific exercise. The markers create two vectors on the 2D plane of the hand. The inner product of the vectors was used to extract the ROM value. Misclassification of the joints or inaccurate detection of the marker position leads to incorrect ROM measurement; hence, accurate detection and localization of markers is crucial. The presented method for marker localization is designed so that the privacy of the users is preserved by avoiding image transmission in the network. Furthermore, through Federated Learning, local information of the input image can be used along with the global model to extract the hand region accurately. This helps to avoid selecting background pixels as a hand region or markers. A marker can be any stick-on material that identifies a joint on the hand skeleton structure. Some examples of the hand with markers are shown in Fig. 3. Positioning the markers on the hand is a trivial task as the hand joints are easily detectable. The only consideration here is to
select markers with colors different compared to skin color. Green and Blue are more appropriate for the task because human skin color has a significant amount of Red. This study considers Blue or Green stickers. Task allocation in the proposed hand rehabilitation monitoring system is presented in Table 1.

### 4.2.1 Personalized skin detection for marker localization

In order to extract the hand region in the image, a skin detection step was implemented, and the largest skin region in the image was used to segment the hand. Skin detection was considered in this study because the skin color distribution can be closely generalized with the Gaussian distribution model, and local adaptation steps can be applied to reduce the shift domain problem.

Skin detection has been a long-standing challenge in the field of computer vision, and it has been used in many studies that involve decision-making about humans [33, 38]. Several survey papers have explored various methods for skin detection [19, 32]. The skin detection method presented in this paper considers the parametric skin detection approach. In a pioneer study [18], it was shown that the Gaussian Mixture Model (GMM) is an efficient tool for skin detection. Recently, the authors have shown that the Multivariate Gaussian Models (MGM) can be applied to improve the skin detection accuracy [31]. In this paper, we combine both models to arrive at a Multivariate Gaussian Mixture Model (MGMM) model with advantages of multivariate distribution and its GMM version. The MGMM model is formulated as:

\[
P(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x|\mu_i, \Sigma_i),
\]

where \(P(x)\) is the skin-ness value of a pixel with feature vector \(x\), and \(\mathcal{N}(x|\mu_i, \Sigma_i)\) is the \(i^{th}\) Gaussian component of the MGMM formulated as:

\[
\mathcal{N}(x|\mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_i|}} \exp\left(-\frac{1}{2} (x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)\right),
\]

where \(\mu\) and \(\Sigma\) are the expected value and covariance matrix of the distribution. The dimension of the \(x\), and model parameters equals to the number of color channels considered to represent a skin pixel \(x\). The mixture component weights are defined as \(\phi\) for each component, with the constraint that \(\sum_{i=1}^{K} \phi_i = 1\), so that the total probability distribution normalized to 1. At inference time, each pixel with feature vector \(x\) and \(P(x) > \theta\), where \(\theta\) is a predefined threshold, is labeled as skin.

| Task                                    | Edge Device | Cloud |
|-----------------------------------------|-------------|-------|
| Marker Localization                     | ✓           |       |
| Marker Recognition                      | ✓           |       |
| ROM Analysis                            | ✓           |       |
| Hand Region Extraction                  | ✓           | ✓     |
| Rehabilitation Exercise Recognition     | ✓           | ✓     |
| Global Hand Extraction Model            | ✓           |       |
| Rehabilitation Exercise Recognition Model | ✓         |       |

Table 1 Image analytic task allocation in the proposed hand rehabilitation monitoring system

Multimedia Tools and Applications
In order to learn the MGMM model, a global model is learned via pre-labeled data and then refined using local models obtained from input videos. The initial local skin models can be obtained using a pre-annotation step via a graphical user interface, or it can use the global model.

At the global scale, MGMM is used to find the initial skin pixel for an input image on the edge device. Then, the initial skin pixels will be used to estimate the skin color distribution of the input skin image. Once the skin model of the patient is obtained, the model will remain unchanged through the rest of the session. The following steps have been applied to extract the hand regions using skin detection.

1. The global skin color model $P_{\text{global}}(x)$ was extracted. Several public datasets are available for modeling the skin color distribution. HGR dataset with 899 hand images with pixel-level labels was considered for this study [16]. The MGMM model uses different color components for modeling skin color. Each image was transformed into YCbCr and HSV color spaces. Each color component provides a different encoding of color information, which can benefit the accuracy of skin detection. The best combination of color spaces for representing skin was found through experiments. The global model was stored in the cloud device. Also $P_{\text{global}}(x)$ is dispatched to edge devices for initializing skin detection.

2. $P_{\text{global}}(x)$ was used in the edge device to acquire the initial skin model. The initial skin pixels are then used to build a univariate Gaussian model for skin distribution. Univariate Gaussian model, $P_{\text{local}}(x)$, was considered for modeling the skin color distribution of an individual image based on previous studies.

3. The aggregator collects all the local models. For each local model, the global model is updated as:

$$P_{\text{updated}}^{\text{global}} = \alpha P_{\text{local}}(x) + (1-\alpha) P_{\text{global}}(x) \otimes C,$$  \hspace{1cm} (3)

where $C$ is the Gaussian cluster $P_{\text{local}}(x)$ belongs to in the MGMM model, and $C$ is one hot coded vector that has the same length as the components of the MGMM model.

![Fig. 4 General Procedure of Hand Recognition](image)

---

Multimedia Tools and Applications
4. The training process is repeated until the global model does not change in two consecutive iterations.

After detecting the skin pixels, the hand region is extracted using some morphological operation in the binary image domain. The process is shown in Fig. 4. The largest skin area

Algorithm 1. The Federated Skin Learning Procedure.

1. If \( \| P_{global}^k - P_{global}^k \| > \varepsilon \) model needs update:
2. In the edge device:
3. Import Global Skin \( P_{global}^k \) model at step \( k \)
4. Apply \( P_{global}^k \) to extract the Skin Map at step \( k \)
5. \( P_{local} = \mathcal{N}(x_{local}, \mu_{local}^i, \Sigma_{local}^i) \)
6. Send \( p_{local} \) to the cloud
7. In the cloud:
8. At step \( k \), aggregate all the local models from available edge devices
9. Update the model with \( \alpha P_{local}(x) + (1 - \alpha) P_{global}(x) \otimes C \)
10. Else
11. Import Global Skin \( P_{global}^K \) model at step \( K \) as the final skin model
12. Apply \( P_{global}^K \) to extract the final Skin Map
13. Apply morphology to extract the hand region

in the image was considered as the hand, assuming that the hand is present in the image. In the recognized region, colored markers were also eliminated from the image. These regions were retrieved using erosion-dilation operations. An open-close morphological operation was also applied to get the cleaned area of the hand. Finally, by multiplying the binary image with the original image, the final colored image of the hand, including the labels, was extracted.

Since the hand region was extracted in the previous step, extracting marker positions in the image is then trivial. Thresholding in the HSV color space was used to extract the marked regions. Marker pixels are defined as those with a significant color difference with the skin color. The coordinates of marker centers were obtained by averaging the spatial coordinate of pixels belonging to a marker. Upon completing this step, several isolated marker regions are detected, and the pixels of each region indicate the presence of a marker. The center of the mass for each marker region was considered as the position of the desired joints. These positions are passed to the exercise recognition and ROM measurement step. The process of hand region segmentation is summarized in Algorithm 1.

4.2.2 Exercise recognition and ROM

Exercise recognition is implemented to determine which rehabilitation exercise is performed and which joints should be considered for ROM measurement. A total of seven exercises were considered in this study. Some exercises were calculated for each finger separately. Exercises and the corresponding normal ROM range are specified in Table 2. Note that, the formal definition of each exercise can be found in [6]. The hand pose for each exercise and the joints

\[ \text{http://mocap.cs.cmu.edu/} \]
responsible for calculating ROM are illustrated in Fig. 5. Exercise recognition can be considered as an activity recognition problem. In conventional action recognition methods, skeletal data are obtained via marker-based methods such as the CMU dataset\(^1\) or marker-less pose reconstruction methods [39]. In markerless methods, the skeleton data is learned jointly, and all joints should be recognized at once. A big disadvantage of these methods is handling occluded joints. In many video frames, some joints are not in the field of view of the camera, and their position is obtained by applying methods such as Inverse Kinematics (IK) and Point Cloud (CP) methods. However, as shown in our experimental results, marker-less methods do not provide the required accuracy for application such as ROM measurement. Given a video stream with marker positions, then the problem of exercise recognition is similar to human action recognition with skeleton data [24, 48].

Motivated by research in the field of human action recognition, the Naive-Bayes Nearest Neighbor (NBNN) classifier was considered for rehabilitation exercise recognition [46]. The main advantage of this method is that it does not require learning parameters. Accordingly, new exercises can be added to the list of exercises, and given a set of labeled exercise samples, the new exercise can also be detected.

To implement the method, for each frame of the input video, an appropriate exercise representation feature vector based on the coordinate of the detected markers should be provided. Accordingly, given the position of markers at frame \(f_i\), a descriptor \(f_v_i\) was calculated, and the training frame, which minimizes the distance to this vector, was searched:

\[
 d_i^{C_k} = \|f_v_i - NN_{C_k}(f_v_i)\|^2
\]

\(NN_{C_k}(f_v_i)\) is the NN-descriptor of \(f_v_i\) in the training dataset that belongs to class \(C_k\). In this equation, \(C_k\) is the exercise class with exercise \(K = 1, 2 \cdots, 7\). The searching process is repeated for each frame \(f_i\) in the input video. Each element of sequence \(K\) is assigned with an exercise

| Rehabilitation Exercise                             | Abbreviation | ROM (in degrees) |
|-----------------------------------------------------|--------------|-----------------|
| Thumb Carpometacarpal Radial Abduction              | CMC          | 0–50            |
| Metacarpophalangeal Abduction                       | MA           | 0–25            |
| Wrist Ulnar Deviation                              | WUD          | 0–30            |
| Wrist Radial Deviation                             | WRD          | 0–20            |
| Wrist Extension                                    | WE           | 0–70            |
| Wrist Flexion                                      | WF           | 0–80            |
| Metacarpophalangeal Flexion                        | MP           | 0–90            |

Fig. 5 ROM measurement for wrist and finger exercises
label. After processing all the frames of training videos, \( K \) sequence is produced. Having calculated \( d_k^i \) for all frames, the class probability of each class was calculated as:

\[
G_{C_k} = \frac{1}{F} \sum_{i=1}^{F} \exp \left( \frac{d_k^i}{\sigma} \right)
\]

(5)

where \( F \) is the number of frames in the input video. Note that the descriptor used in this study is defined as:

\[
f v_i = S \bigg[ \| p_1^{y_{\text{min}}}-p_1^{y_{\text{max}}} \|, \| p_1^{x_{\text{max}}}-p_1^{x_{\text{min}}} \|, \| p_1^{y_{\text{min}}}-p_1^{y_{\text{max}}} \|, \| p_1^{y_{\text{min}}}-p_1^{y_{\text{max}}} \| \bigg],
\]

(6)

where \( p \) is the coordinate of the marker in the \( i^{th} \) frame, and \( y_{\text{min}}, y_{\text{max}}, x_{\text{max}}, x_{\text{min}} \) represent markers with the maximum and minimum coordinate values across all the detected markers. In addition, \( S \) is a scalar and it is equal to the number of markers detected in the input image. This helps to distinguish between the feature vector of the hand in the horizontal and perpendicular poses with respect to the camera. It should be noted the number of markers detected in the two poses is significantly different. Thus, \( S \) scales the \( f v \), thus the exercises in each pose could be easily detected. Experimental results and comparison with other approaches for describing an input frame showed that this setting has superior performance for exercise recognition.

In the case of continuous exercise recognition, our approach for upper body ROM exercises explained in [29] can be applied repeated. Continuous exercices can be considered as a set of ROM motions performed in a sequence. The set is denoted as \( E_c, c \in \{1, 2, \ldots, C\} \), where \( c \) is the detected exercise. Each ROM activity targets a specific joint, and its value is either increasing or decreasing. So, the change in rotation sequence of moving joint can indicate start/end points of an exercise. To detect these changes, let \( E^i_c \) represent the \( i^{th} \) frame of exercise \( c \), then if \( E^i_c = E^{i+1}_c \), exercise \( c \) is still active. However, if \( E^i_c \neq E^{i+1}_c \) and \( E^i_c + E^{i+1}_c \), then a start/end point is detected, and the exercise class is updated with \( E^i_c \).

After identification of an exercise, the ROM value can be obtained via the angle between the inner product of the rotating joint and the proper adjacent joints as \( \cos^{-1} \left( \frac{v_1 \cdot v_2}{|v_1||v_2|} \right) \). \( v_1 \) and \( v_2 \) are the vectors connecting the revolving joint with two other joints. If the exercise is left uncompleted, the smaller ROM value will indicate impairment, and the medical experts can devise treatment plans to improve the patient’s ROM.

**5 Experimental results**

In this section, we explain the results of evaluating the proposed method. The results are explained in four subsections: details of the dataset created for this study, the validity of ROM measurements using the proposed method, the performance of Federated skin learning, and the performance of exercise recognition. Note that in the second subsection, the validity of ROM measurements using the proposed method has been compared to the results of ROM measurement obtained from the standard Goniometry approach and the marker-less solution presented by the Leap Motion device. Leap Motion is a commercial depth-sensing camera released in 2013. Leap Motion has two cameras and
three infrared LEDs. The dual camera creates a gray stereo image using infrared light. Leap Motion extracts hand data from raw sensor data using computer vision-based algorithms. Leap Motion can detect hand joints from 2.5 to 80 cm away from the camera plane. The field of view of the camera is 150°. The camera, along with its algorithms, has been previously studied for ROM measurement in [42] and a recent study conducted by authors [20].

5.1 Dataset

The exercises chosen in this study were chosen in accordance with Pendleton [34]. Based on Pendleton, ROM exercises for the hand include 4 exercises for the wrist and 6 exercises for the fingers, which can be performed by the patient. These can be considered as task-oriented training exercises, defined as single-joint and single plane movement exercises [41]. Note that different types of exercises were addressed in Section 2. Other exercises can be a sequential combination of the mentioned exercises. In this study, we included 4 wrist and 3 fingers exercises. The ROM measurement for Metacarpophalangeal Joint Extension and Proximal Interphalangeal Flexion are measured similar to exercise CMC Abduction. There is no difference between them. Furthermore, the exercises such as Metacarpophalangeal Abduction (MA) are measured for each finger separately. Thus, MA and similar finger exercises are considered for each finger accordingly. It should be considered that in order to measure the ROM and assess the performance of the patient, only target-oriented exercises, i.e., single-joint exercises, are required. However, if the assessment of a sequence of motions is required, the exercise can then be broken down into simpler motions, and each motion should be processed individually as explained in the Section 4. If one wishes other motions that are not seen in the proposed approach, the desired motion can be defined based on the joints involved in a new exercise.

To obtained records of the exercises, five healthy volunteers were recruited. Healthy volunteers were considered so they can cover the full ROM during the exercise and make records available for all normal ROM ranges. The process of sticking markers and guidelines on how to perform each exercise was explained to them by an occupational therapy expert. Each exercise was recorded with a commercial RGB camera and the Leap Motion sensor. Each volunteer repeated each exercise 6 times, making a total of 210 records of the exercises. For the sole purpose of validating ROM measurement accuracy with the marker-based solution presented here, separate experiments were conducted. One of the volunteers was asked to repeat each exercise while the exercise was recorded with the normal RGB camera, a Leap Motion sensor. Also, the occupational therapist provided Goniometry reading for each record as reference values. In order to record the Leap Motion data, a UI was created to control the capturing process. In all cases, a similar procedure to the proposed vision-based approach was followed. For a specific exercise, using the definition of an exercise, the appropriate joints were extracted from the recorded data. It should be considered that in this situation, the joints names were extracted using the Leap Motion SDK. In order to calculate the ROM, for a detected exercise, the difference between the smallest and the largest angles in all the frames was considered as the ROM value for the desired joint.

5.2 Validation of ROM measurement

In order to examine the accuracy of the marker-based method, we compared our results with Goniometry and the Leap Motion capture system. Figure 6 shows the Mean Squared Error
(MSE) for each exercise. The values obtained from Goniometer were used as a reference. As the results indicate, the marker-based method shows small error values, whereas the readings obtained from Leap Motion show more than several degrees of error. The reason was found to be an unstable marker localization of the Leap Motion during motion capture. The Standard Deviations (SDs) of reading from the three devices are also shown in Fig. 7. The numbers show that the readings obtained from the marker-based method are more stable compared to the other two methods. The cause of higher SD in Goniometer reading redirects to human error, while the significant instability of the reading obtained from the Leap Motion are results of the inaccurate joint localization algorithm. According to these results, it can be implied that markerless joint localization, despite being very easy to use, has room for improvement, and the current methods do not meet the accuracy requirements of applications such as ROM measurement.

5.3 Performance of federated skin learning

An important step of the proposed method is the federated skin learning step. This step should both be evaluated in terms of the proposed MGMM model and the Federated Learning step. The MGMM model is sensitive to several parameters, including the number of Gaussian in the mixture model and the choice of color components for skin modeling. Figure 8 shows the
results of the True Positive Rate (TPR) of detecting skin pixels versus False Positive Rate (FPR) of detecting non-skin pixels based on Recursive Operating Characteristics (ROC) curves. In the experiments, the number of Gaussians in the mixture model was changed from GMM = 1 to 6. The results showed that considering 4 Gaussians, GMM = 4, to model the skin color distribution has better performance. Also, based on experimenting with different color channels, it was found that the combination of Hue component from the HSV color space and Cb, Cr components from the YCbCr space are the best choices of color component for skin detection. From the results, it can be implied that the combination of GMM and using Multivariate Gaussian distributions can improve the results of skin detection significantly.

For selecting parameter $\alpha$, three settings were tested: combining the local and global model with equal weights, i.e., $\alpha = 0.5$, making more weight on the global model and vice versa. We found when the local model is more weighted, the local model tends to better recognize the skin pixels. We found that alpha = 0.6 works well with our data. In practice, this parameter can be adjusted for better performance based on the available data. However, the choice to put more weights on the local model was confirmed with our results.

Federated skin learning was applied to update the global skin color model. This step was proposed to obtain an updated skin model in the domain of images being used in the ecosystem. The model should be updated every time a new video is being processed. The global model was updated repeatedly until the changes to the model were negligible. Figure 9 shows the accuracy of skin detection on skin test pixels. Three different thresholds were
applied to Eq. (3). As the results show, after 5, or 6 iterations, the skin model starts to converge, and skin pixels were detected accurately. In the test process, non-skin pixels were obtained from the background of the skin images.

In order to evaluate the proposed method, 6 parties were involved in the study. The FL part of the solution is designed for the Privacy-Preserving of the users/clients. Our Federated Learning is designed to detect the hand region by recognizing the skin pixel in the image. Hence, the accuracy of the skin detection in FL and Traditional ML (None-FL) can be compared together. In order two compare these two approaches, we setup FL and None-FL setups. In the none-FL setup, the data for all parties were aggregated, and a single dataset was created for training and testing the MGMM model. In the FL setup, data from each party was considered individually and isolated from other parties. The steps for FL were applied according to Algorithm 1. The initial model was sent to each client, and the local model was learned. All the local models were aggregated and sent back to the cloud for the model update. To evaluate the model, Recursive Operating Curve (ROC) based on True Positive Rate (TPR) and False Positive Rate (FPR) can be used. Also, the model can be evaluated based on the accuracy using the classification decision threshold $\theta$ as explained in Eq. (2). Accuracy can be defined as $\frac{TPR + TNR}{P + N}$, where TNR is the True Negative Rate, and P and N represent positive and negative samples, accordingly. Figure 10 shows the accuracy of the method. The figure shows the accuracy of the method for different choices of $\theta$. The accuracy values highlighted in the blue region have high TPR and small FPR values, and hence, this region of thresholds and the accuracy values can be regarded as an acceptable range of accuracy. For each client, the accuracy of the trained model is also shown in the figure. As it can be seen, based on the distribution of the test pixel on each local server, different accuracy was obtained. The average accuracy of test pixels is shown with the thick blue line. As can be seen, a reduction in the performance of the FL approach compared to the None-FL can be observed. For example, for $\theta = 0.12$, where, TPR for all clients is over>90%, the average accuracy of the FL approach is 4% less, compared to Traditional FL. This decrease in performance is of course, different for each client when considered individually as shown with gray lines in the figure. It should be noted that in both FL and None-FL setups, the HGR dataset was also considered for model initialization. In the None-FL setup, the collected and HGR data were
used in the training process. In the FL setup, the HGR dataset was used for learning the initialized model. In Both setups, equal train-test splits was considered. The results are shown for test splits.

The ROC curve during training each party reveals two important points (Fig. 11). Each ROC curve in each figure shows TPR and FPR for one round of model updates. Two points can be observed from these figures. First, the ROC curves show that after a certain number of iteration, each client starts to stabilize, and further training does not impact the detection rate in terms of TPR and FPR. These figures also that it takes $\theta > .1$ for TPR values in all figures to reach a value greater than 90%. It should be noticed that TPR and FPR values are drawn for changing $\theta \in [0 1]$. Here we changed $\theta$ in 0.01 steps. These figures also reveal how the performance of FL in terms of the runtime can be evaluated. Runtime is affected by several factors, including the number of samples in each client, variability of data with respect to the initial model, and the number of iteration that is required for the model to update. Figure 11 shows the ROC curves for at each iteration. Some models on the client side require more steps to converge while others converge with fewer updates. The decisive parameter on runtime then depends on the number of iterations that is required to update all models. The more iteration it is required, the more time it is required to train the mode. In our dataset, after six or seven iterations, all the local models converged.

5.4 Performance of exercise recognition

One of the challenges in ROM measurement was to identify the detected markers. This step was required to specify which markers should be selected for ROM measurement. To tackle the problem, two approaches were considered: a learning-based approach and a data-driven approach. In the learning approach, sticker identification was considered as a classification problem with the number of markers as the number of classes. The main problem here was that based on the posture of the hand, the number of detected stickers was different and did not

Fig. 11 TPR and ROC curves for each local model. ROC curve in each figure illustrates TPR and FPR values for changing $\theta \in [0 1]$ after consecutive model updates
meet the requirement of this paper. In the classification scheme, we considered two possible feature representations. Both presentations used the spatial position of the detected markers. First, we considered the Euclidian distance between the detected markers and their coordinates in the image as features. In the second presentation scheme, all the possible permutations of the detected markers were considered, and the Euclidian distance between markers was used as the feature vector of the input video frame. A KNN classifier was considered to choose the nearest combination and assign a class label to each of the markers.

Table 3 summarizes the results of marker classification for both presentations. As the results indicate, none of the presentations shown significant accuracy for marker recognition. In both cases, high variance in the number of detected markers on the hand was the cause of insignificant results. Figure 12 shows an example of the feature vector containing both distance and relative position of markers for a detected marker A. Based on observations, it was found that using a data-driven approach, and outlying markers were always detectable and recognizable. Using four markers that contain the min-max x and y coordinates among the detected markers, all the other markers could be identified. The data-driven approach was able to correctly identify all the detected markers. The reason for the significant improvement in the accuracy, compared to learning approaches, could be the biasing the data-driven approach by providing information about the class of the four outlying markers.

For exercise recognition, several conventional classifiers were considered and tested. The result of exercise classification is provided in Table 4. The classifiers used in the study included: Decision Trees (DT), Support Vector Machines (SVM), and Long Short Term Memory (LSTM). Once the marker position we obtained, distinguishing between different exercises was trivial because, in each exercise, a distinct set of joints were rotating. However, we tested several classifiers to evaluate the robustness of the method. In our case, the NBNN is a very suitable choice because it is easily scalable of the number of exercises change. NBNN works on the basis of nearest neighbors. In any case that the number of exercises is increased, there is no need to retrain the model. All is required is to provide examples of the new exercise and add them to the previously saved data. At inference time, frames are classified based on their Euclidian distance to labeled data. The Bayesian scheme is then added to the number of

### Table 3 Marker recognition using spatial features

| Feature representation                  | Precision (%) | Accuracy (%) |
|----------------------------------------|--------------|--------------|
| Distance                               | 55.9         | 56.6         |
| Distance + coordinates                 | 57.6         | 58.2         |
| The relative position of markers       | 59.7         | 58.4         |
| Data-driven marker recognition         | 100          | 100          |

Fig. 12 Feature vector for describing a marker using euclidian distance and marker relative positions. Elements of the feature vector include, $d_i$ which stands for distance. For relative position, given marker A as an entry, the relative position feature with regard to other markers are encoded as $-1$ if they fall below or to the left of the reference marker and vice versa for $+1$.
detected frames to classify the exercise. Hence, the method can be easily scaled to other situations with new exercises.

In the NBNN approach, parameter $\sigma$ in Eq. (5) was set to 0.1. It was found that this parameter did not significantly contribute to the accuracy of exercise recognition. Since the variance between similar actions was much less compared to actions from different classes, i.e., significant intra-class variance compared to the between-class variance, small values of $\sigma$ suffices.

In the LSTM scheme, we handled exercise recognition as a time series classification problem. Each exercise was represented with 15 frames, and 10 features form the marker coordinates. Accordingly, the input vector was a tuple $(\text{batch size}, F, fv)$. In the neural network architecture, we stacked to LSTM layer followed by a dense layer. The cross binary entropy loss was used for the cost function. Several batch sizes were considered to find a better parameter. However, as a hyper parameter, this could change based on the amount of data at hand.

### 6 Conclusions

The revolution of IoT has created new opportunities in the realm of smart and personalized healthcare. IoT healthcare enables a transition toward patient-centric care offering a diverse array of opportunities such as increased accessibility, availability, customization, cost efficiency, and precision. Combining the capabilities of edge, fog, and cloud creates a hierarchical IoT paradigm (i.e., Edge-Fog-Cloud) to improve IoT healthcare functioning. Within the healthcare industry, a massive deluge of data such as images from the patient performing rehabilitation exercises is generated by IoT cameras. Transmitting the patient’s private data through non-secure media, as well as the processing and storing this data in centralized servers in the cloud, generates many privacy concerns that will need to be addressed. Therefore, analyzing health data while simultaneously safeguarding personal privacy is a complicated and urgent challenge. This paper proposed a holistic hierarchical and Privacy-Preserving IoT architecture for hand rehabilitation assessment based on machine vision techniques. We presented a novel ML technique to accurately recognize the Range of Motion as a critical indicator for analyzing the progress of patients during the rehabilitation phase. We demonstrated how the proposed ML techniques could be mapped to the edge, fog, and cloud layers. Furthermore, we illustrated how Federated Learning, differential privacy, and secret sharing techniques could be incorporated to be able to preserve the privacy of data owners.

**Acknowledgments** We would like to thank Mr. Ramin Banimehdi for providing us with Goniometry measurements.

---

### Table 4  Results of exercise recognition

|       | DT  | NBNN | SVM | LSTM |
|-------|-----|------|-----|------|
| Wrist | 100 | 100  | 100 | 100  |
| Fingers | 84.27 | 84.87 | 84.2 | 84.30 |
| Total | 99.04 | 99.43 | 99.03 | 99.00 |

---
References

1. Acar A, Aksu H, Uluguc AS, Conti M (2018) A survey on homomorphic encryption schemes: theory and implementation. ACM Comput Surv 51:1–35
2. Al-Rubaie M, Chang JM (2019) Privacy-preserving machine learning: threats and solutions. IEEE Secur Priv 17:49–58
3. Aono Y, Hayashi T, Wang L, Moriai S (2017) Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans Inf Forensics Secur 13:1333–1345
4. Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V et al (2019) Towards federated learning at scale: System design. https://arxiv.org/abs/1902.01046v2
5. Boone DC, Azen SP, Lin C-M, Spence C, Baron C, Lee L (1978) Reliability of goniometric measurements. Phys Ther 58:1355–1360
6. Bronzino JD (2000) Biomedical Engineering Handbook. CRC Press, Boca Raton
7. Carter TI, Pansy B, Wolff AL, Hillstrom HJ, Backus SI, Lenhoff M, Wolfe SW (2009) Accuracy and reliability of three different methods for manual goniometry for wrist motion: a cadaveric study. J Hand Surg 34:1422–1428
8. Dwork C (2008) Differential privacy: a survey of results. In: International conference on theory and applications of models of computation. https://doi.org/10.1007/978-3-540-79228-4_1
9. Eini DS, Ratzon N, Rizzo A, Yeh S, Lange B, Yaffe B et al (2010) A simple camera tracking virtual reality system for evaluation of wrist range of motion. In: Proc. of the 8th Intl Conf. on Disability, Virtual Reality & Associated Technologies, pp 123–130
10. Farahani B, Firoouzi F, Chang V, Badaroglu M, Constant N, Mankodiya K (2018) Towards fog-driven IoT eHealth: promises and challenges of IoT in medicine and healthcare. Futur Gener Comput Syst 78:659–676
11. Ferri'ndez-Baena A, Susin A, Ligadas X (2012) Biomechanical validation of upper-body and lower-body joint movements of kinect motion capture data for rehabilitation treatments. In: Fourth international conference on intelligent networking and collaborative systems. https://doi.org/10.1109/INCoS.2012.66
12. Firoouzi F, Farahani B, Ibrahim M, Chakrabarty K (2018) Keynote paper: from EDA to IoT eHealth: promises, challenges, and solutions. IEEE Trans Comput-Aided Design Integr Circuits Syst 37:2965–2978
13. Gajdosik RL, Bohannon RWJPT (1987) Clinical measurement of range of motion: review of goniometry emphasizing reliability and validity. Phys Ther. https://doi.org/10.1093/ptj/67.12.1867
14. Georgeu G, Mayfield S, Logan AM (2002) Lateral digital photography with computer aided goniometry versus standard goniometry for recording finger joint angles. J Hand Surg Br. https://doi.org/10.1054/jhsb.2001.0692
15. Gibert S, Carozzo C, Cachon T, Fau D, Genevois J, Viguier E (2012) Evaluation of the effectiveness of bandages on restricting carpus range of motion in healthy dogs at a walk using electrogoniometry. Computer Methods in Biomechanics and Biomedical Engineering. https://doi.org/10.1080/10255842.2012.713730
16. Grzejsczak T, Kauwolok M, Galuszka A (2016) Hand landmarks detection and localization in color images. Multimed Tools Appl 75:16363–16387
17. Ji Z, Lipton ZC, Elkan C (2014) Differential privacy and machine learning: a survey and review. https://arxiv.org/abs/1412.7584v1
18. Jones MJ, Rehg JM (2002) Statistical color models with application to skin detection. Int J Comput Vis 46:81–96
19. Kakumanu P, Makrogiannis S, Bourbakis N (2007) A survey of skin-color modeling and detection methods. Pattern Recogn. https://doi.org/10.1016/j.patcog.2006.06.010
20. Kavian M, Nadian-Ghomsheh A (2020) Monitoring wrist and fingers range of motion using leap motion camera for physical rehabilitation. In: International Conference on Machine Vision and Image Processing. https://doi.org/10.1109/MVIP49855.2020.9116876
21. Knepley KD, Mao JZ, Wieczorek P, Okoye FO, Jain AP, Harel NY (2020) Impact of Telerehabilitation for stroke-related deficits. Telemed J E Health. https://doi.org/10.1089/tmj.2020.0019
22. Li P, Li J, Huang Z, Li T, Gao C-Z, Yiu S-M, Chen K (2017) Multi-key privacy-preserving deep learning in cloud computing. Futur Gener Comput Syst 74:76–85
23. Li T, Sahu AK, Talwalkar A, Smith V (2020) Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag 37:50–60
24. Liu B, Cai H, Ju Z, Liu H (2019) RGB-D sensing based human action and interaction analysis: a survey. Pattern Recogn 94:1–12
25. Malitas DB, Ferland C, Perron M, Roy J-S (2019) Reliability of inclinometer-derived passive range of motion measures in youth with cerebral palsy 39:655–668
26. Mastos M, Miller K, Eliasson A-C, Imms C (2007) Goal-directed training: linking theories of treatment to clinical practice for improved functional activities in daily life. Clin Rehabil 21:47–55
27. McVeigh KH, Murray PM, Heckman MG, Rawal B, Peterson JJ (2016) Accuracy and validity of goniometer and visual assessments of angular joint positions of the hand and wrist. J Hand Surg 41:e21–e35
28. Mortazavi F, Nadian-Ghomsheh A (2018) Stability of Kinect for range of motion analysis in static stretching exercises. PLoS One 13:e0200992
29. Mortazavi F, Nadian-Ghomsheh A (2019) Continues online exercise monitoring and assessment system with visual guidance feedback for stroke rehabilitation. Multimed Tools Appl 78:32055–32085
30. Mündermann L, Corazza S, Andriacchi TP (2006) The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications. J Neuroeng Rehabil 3:1–11
31. Nadian-Ghomsheh A (2017) Hybrid color-texture multivariate Gaussian model for skin detection. In: 10th Iranian Conference on Machine Vision and Image Processing. https://doi.org/10.1109/IranianMVIP.2017.8342336
32. Naji S, Jalab HA, Kareem SA (2019) A survey on skin detection in colored images. Artif Intell Rev 52: 1041–1087
33. Paracchini M, Marcon M, Villa F, Tubaro S (2020) Deep skin detection on low resolution grayscale images. Pattern Recogn Lett 131:322–328
34. Pendleton HM, Schultz-Krohn W (2017) Pedretti’s Occupational therapy: Practice skills for physical dysfunction. Elsevier, St. Louis
35. Reither LR, Foreman MH, Migotsky N, Haddix C, Engsberg JR (2018) Upper extremity movement reliability and validity of the Kinect version 2. Disabil Rehabil Assist Technol 13:54–59
36. Shamir A (1979) How to share a secret. Commun ACM 22:612–613
37. Shamsi M, Mirzaei M, Khabiri SS (2019) Universal goniometer and electro-goniometer intra-examiner reliability in measuring the knee angle during active knee extension test in patients with chronic low back pain with short hamstring muscle. Med Rehab. https://doi.org/10.1186/s13102-019-0116-x
38. Shifa A, Imtiaz MB, Asghar MN, Fleury M (2020) Skin detection and lightweight encryption for privacy protection in real-time surveillance applications. Image Vis Comput 94:103859
39. Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R et al (2011) Real-time human pose recognition in parts from single depth images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2011.5995316
40. Thomas CK, Zaidner E, Calancie B, Broton J, Bigland-Ritchie BJEN (1997) Muscle weakness, paralysis, and atrophy after human cervical spinal cord injury. Exp Neurol. https://doi.org/10.1006/exnr.1997.6690
41. Timmermans AA, Spooren AI, Kingma H, Seelen HA (2010) Influence of task-oriented training content on skilled arm-hand performance in stroke: a systematic review. Neurorhabil Neural Repair 24:858–870
42. Trojo RL, Vázquez JPG, Ramirez MLG, Corral LEV, Marquez IR (2017) Hand goniometric measurements using leap motion. In 14th IEEE Annual Consumer Communications & Networking Conference. https://doi.org/10.1109/CCNC.2017.7983095
43. Veerbeek JM, van Wegen E, van Peppen R, van der Wees PJ, Hendriks E, Rietberg M, Kwakkel G (2014) What is the evidence for physical therapy poststroke? A systematic review and meta-analysis. PLoS One 9:e87987
44. Voigt P, Von dem Bussche A (2017) The eu general data protection regulation (gdpr) A Practical Guide. Springer, Cham. https://doi.org/10.1007/978-3-319-57959-7
45. Wang Y, Neff M (2013) Data-driven glove calibration for hand motion capture. In Proceedings of the 12th ACM SIGGRAPH. https://doi.org/10.1145/2485895.2485901
46. Weng J, Weng C, Yuan J (2017) Spatio-temporal naive-bayes nearest-neighbor (st-nbnn) for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2017.55
47. Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: concept and applications. ACM Trans Intell Syst Technol 10:1–19
48. Zhang H-B, Zhang Y-X, Zhong B, Lei Q, Yang L, Du, et al J-X (2019) A comprehensive survey of vision-based human action recognition methods. Sensors. https://doi.org/10.3390/s19051005

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Ali Nadian-Ghomsheh1 · Bahar Farahani1 · Mohammad Kavian1

1 Cyberspace Research Institute, Shahid Beheshti University, Tehran, Iran

Springer