Journal on Interactive Systems, 2022, 13:1, doi: 10.5753/jis.2022.2409

Supervised Classification of Motor-Rehabilitation Body Movements with RGB Cameras and Pose Tracking Data

Luis Guilherme Silva Rodrigues [**Universidade Estadual Paulista** | ***guilherme.rodrigues@unesp.br***]
Diego Roberto Colombo Dias [**Universidade Federal de São João del-Rei** | ***diegodias@ufsj.edu.br***]
Marcelo de Paiva Guimarães [**Universidade Federal de São Paulo** | ***marcelo.paiva@unifesp.br***]
Alexandre Fonseca Brandão [**Universidade Estadual de Campinas** | ***brandaobiotec@gmail.com***]
Leonardo C. Rocha [**Universidade Federal de São João del-Rei** | ***lcrocha@ufsj.edu.br***]
Rogério Luiz Iope [**Universidade Estadual Paulista** | ***rogerio.iope@unesp.br***]
José Remo Ferreira Brega [**Universidade Estadual Paulista** | ***remo.brega@unesp.br***]

Abstract The technological evolution allowed the use of a single camera for precise and effective body tracking, reducing the cost and increasing the accessibility of applications in places where depth cameras and wearable sensors are not available. This paper describes and implements a supervised machine learning process consisting of a mobile application used as a motion capture device which also transforms the data into an input for a machine learning model that classifies upper and lower limbs movements (24 types of human movements). The user performs movements in front of the camera, and the trained model classifies them. We designed the system to work in a motor-rehabilitation context to assist the professional while the patient does physical exercises. The implementation can summarize the movements made during the rehabilitation sessions by counting the repetitions and classifying them when done completely or reached a specific range of motion.

Keywords: *Classification, Computer Vision, Machine Learning, Pose Tracking, Supervised Learning, Motor-rehabilitation*

1 Introduction

A stroke can be a vessel blockage (ischemic) or a vessel rupture (hemorrhagic) in the brain that damages the tissue around the occurrence. Stroke is not always lethal and usually leaves a sequel in survivors (Mukherjee et al., 2006) that vary depending on the affected region. Visual disturbances, speech problems, loss of coordination, and dizziness are possible sequelae. The treatments also vary and usually require rehabilitation therapies, such as neuromotor rehabilitation and speech therapy. It is one of the diseases that cause most long-term disability worldwide, generates costs for the health system, and reduces patients' quality of life. Five and a half million people die of stroke each year, and there are more than 80 million people living today in the world who have suffered a stroke (World Stroke Organization, 2019).

Some of the cells in neural pathways can die or recover only to some degree due to a stroke. The neuromotor treatment stimulates the rearrangement of motor circuits due to the brain's capacity for plasticity (Borich et al., 2018; Mehta and Keshavan, 2015). The recovery is directly related to the intensity of therapy, so the earlier the rehabilitation process is started, the better the results (Dimyan and Cohen, 2011). Patients must not give up on physical therapy to have the best recovery possible. It necessitates approaches that improve engagement and interest in the process that can take months to complete. Serious games and virtual and augmented reality environments can increase patients' interest in physical therapy (Rego et al., 2010) and provide detailed session data to the therapist.

The interaction with virtual environments within a physical therapy treatment is still limited since most computer applications in this field still rely on specific equipment (i.e., Kinect). Besides requiring extra investments by a clinic, this issue also prevents game-based treatments and virtual reality environments from using for home care. This demand grew during the health restrictions caused by the SARS-CoV-2 pandemic. The use of virtual reality applications to assist physical therapy sessions at home is becoming more accessible to the public because of the hardware evolution and even algorithms that enable the use of an ordinary camera instead of a depth camera for body tracking, which generates a lower cost.

In 2010, Microsoft launched the Kinect body-tracking device as a follow-up to the Xbox gaming console. Applications in other areas, including research, also started to use this device as a human-computer interface, surpassing entertaining purposes.

Kinect device has an additional depth (D) sensor and its RGB-D image-capturing results in a precise skeleton structure. However, the use of RGB-D devices is less accessible than RGB cameras, which include PC (Personal Computer) webcams and smartphone ones. In order to test the feasibility of the RGB alternative, this work combines the Pose Tracking technique with applications that generally use the Kinect device.

Applications that support gesture interaction are becoming increasingly popular due to the access of body tracking from devices with standard cameras, mainly through smartphones. It brings the opportunity for body tracking data to be part of data-driven studies such as data science and machine learning. Deep learning has shown to be a good solution in computer vision tasks, including Pose Estimation from standard RGB (red, green, and blue color channels) images. When Pose Estimation happens on sequenced images

such as videos, the task can be called Pose Tracking.

In a previous study, Rodrigues et al. (2021) presented a novel application that recognizes eight movements performed by the users. The results presented that the data generated by the tracking algorithm through RGB devices are suitable for input in classification models and suggest that the process can be extended to more types of movements, keeping the same preprocessing. This paper extended the developed system created by Rodrigues et al. (2021) to analyze more complex movements and recognize both sides of the body since the previous study had only right arm movements. In this paper, we define movement classes that comprise exercises for the shoulders, elbows, and legs, involving most body articulations. They vary in direction and range of motion. We based them on the possible exercises in motor rehabilitation therapy, where the patient must perform the movements involving the affected limb multiple times. The trained classification model presented in this paper can be inserted into any motor rehabilitation system to identify the exercises automatically.

The challenge of classifying movements using machine learning models becomes possible through the implementation of supervised learning. The learned function will detect the intrinsic movement patterns and generalize variation of values in the body articulations of the existing movements in the training set.

One example of real-world applications for machine learning in motion capture data is in a motor-rehabilitation scenario, where the camera captures movements while the patient exercises. Suppose these exercises are being stored in a compatible format to fit into a machine learning model. In that case, some approaches may cluster similar exercises, give a score to an exercise, and even predict a treatment based on historical data.

This work focuses on presenting a data structure representing eight articulations and 24 movements of the body. We also verify its usability by applying a supervised learning technique using a labeling application that connects to the motion tracking system.

The remainder of the paper is structured as follows. In Section 2, we discuss related work. Section 3 explains the details of the materials and methods used to capture human articulations, process, store, and infer the movements. Section 4 presents the results obtained and a discussion of these. Finally, Section 5 contains the conclusions and suggestions for future work.

2 Related Work

This section presents some of the main works related to the proposal presented in this paper and separates them according to the two main themes related to our implementation, motion capture using RGB cameras and machine learning classification with motion capture data.

2.1 Motion capture approaches

In literature, there are many implementations of systems for rehabilitation using Kinect devices (Breedon et al., 2016), which were introduced in 2010 by Microsoft, and other devices with depth sensors, such as Intel RealSense and Leap Motion for hand tracking. Cameras with depth sensors can be more precise, but their costs and availability also increase.

Ordinary cameras, with only the RGB color channels, can be used as an alternative due to their availability in smartphones and desktops/laptops. In literature, other solutions involving motor rehabilitation track the movements of the patient based on a determined color present in the camera (Aung and Al-Jumaily, 2012; Jaffe, 2003). When a subject is wearing a green glove in front of the camera, the system needs to find all green pixels in the camera frame and set the glove's position in the center of the green pixels' area. That is also a viable solution for building an augmented reality system for rehabilitation (Phan et al., 2022) but gives the information of only the joints with the colorful equipment.

Some works use fiducial markers as a means of interaction with the applications, which also allows finding the point where the marker is in the scene (Toh et al., 2011; Dinevan et al., 2011). However, the mentioned applications with ordinary cameras do not have information about the body's skeleton as cameras with depth sensors do.

The markerless capture solution from ordinary camera images is known as 3D Human Pose Estimation, which requires devices with high computational performance. This option may become standard in new implementations in motion capture as technology advances and more efficient algorithms arise. Thus, our work uses this technique, with a framework already available, with an ordinary smartphone. This solution gives complete body information, including articulations of the upper and lower limbs, without needing a depth sensor camera or fiducial markers on the body.

2.2 Movements classification

Choubik and Mahmoudi (2016) have successfully classified human poses using a feature vector calculated from the Kinect skeleton structure. The vocabulary of the classifier had 18 poses associated with both arms. The raw data are values related to the image's origin, and the feature vector for the machine learning models is not the raw X and Y information from the device for each body joint. Instead, they use the coordinate of the joint relative to other joints. It results in an array of 20 subtractions between the points.

Ijjina and Mohan (2014) also classify body movements and have distances between joints as the feature vector. The difference is that it adds a time variable. The samples have a different duration that can take 2 to 15 seconds to perform. In order to have a consistent feature vector to input into the machine learning model, it permanently reduces the frames to equal size of temporal samples. Their feature vector visualized in 2D shows different patterns for different actions (e.g., jumping, waving, and clapping).

Also, Farooq and Won (2015) reviewed many datasets of skeleton data made with the Kinect device. Some works include publicly available datasets with the data generated by the RGB-D camera device. For example, Bloom et al. (2016) have 20 gaming actions (e.g., punch right, punch left, kick right, kick left). Also, Leightley et al. (2015) created a dataset related to healthcare movements and discussed potential ap-

plications for machine learning models applied in motion capture data. Their healthcare dataset consists of 10 seconds of balance with open and closed eyes and feet close together, balance with one leg, sit-to-stand movement in a chair for 60 seconds, jumps, and walking movements.

The data generated by RGB-D camera devices are more detailed than the mentioned approaches using an RGB camera only. However, the Pose Tracking technique can also generate the skeleton structure in real-time, such as the Pose Tracking model in the work of Bazarevsky et al. (2020). The similarity of body structures between Kinect data and the outputs RGB Pose Tracking brings the opportunity of porting the mentioned solutions to be used in RGB-only devices as already implemented by Rodrigues et al. (2021).

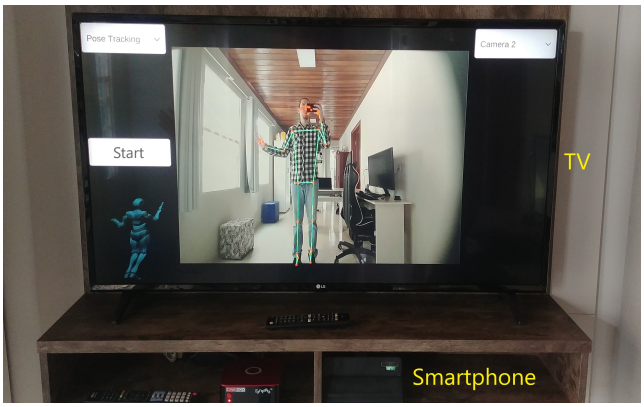

Figure 1. A smartphone is placed below the TV with its camera pointed to the user while its screen is projected to the TV.

3 Materials and Methods

We did the first step of the process, which is capturing the human skeleton, using the framework Mediapipe (Lugaresi et al., 2019), which enables body pose tracking in real-time on smartphones and desktop/laptop computers with a connected webcam. Although we use a specific solution, the framework also supports object detection and segmentation, face landmarks tracking, and hand tracking.

Mediapipe framework decreases the development time of multimedia-related applications that requires computer vision models. It also enables the solution to run effectively with native processing on the hardware for both smartphones and computers.

Mediapipe Pose Tracking algorithm outputs full-body tracking that includes upper and lower limbs. It predicts the location of 33 pose landmarks. It also predicts a full-body segmentation mask represented as a two-class segmentation (human or background) (Bazarevsky et al., 2020). The user should stay about two meters from the front of the camera so the whole body can appear in the camera frame. The distance also depends on the angular view of the camera. Some smartphone cameras have a wider field of view, and a smaller distance is necessary to detect the user's whole body.

We also used Unity[1] game engine to create the interface of the system.

For desktop computers, the webcam tracks the body, and the user can see himself through the monitor. When using a smartphone, its camera should point at the user, and a TV is used for visualization to see the real-time tracking and interact with the system when messages appear on the screen (**Figure 1**).

3.1 Movement definition and data labeling

Real-time body tracking happens when the system tracks the user's silhouette in the camera frame. However, the user will not necessarily start making the defined exercises right away. Also, the user may want to perform movements repeatedly but with a time for resting between movements. Because of this, we used a button to indicate the start and end of the movement.

[1] https://unity.com/

The movements only last a few seconds, and once the user returns to the starting position, it is considered a complete movement.

During the dataset's creation, we connected a second phone to the tracker application through the local network. It synchronizes the tracker application's record button and has a text input to type down the movement's name. When the user finishes a movement, the recording goes to a remote database with the joint data with the respective label. The user stayed with the second phone in the less used hand during the movement so it could press the buttons without walking towards the primary smartphone and causing interference in the recording. The application has a list of accepted names in the settings menu to avoid typos in the text input and waits for the internet connection to send the recording successfully. We also have a manager application where we can see a replay of the recording, check the label, and delete the recording if necessary.

Creating the supervised learning dataset works as follows: the user chooses the movement to be performed and stays in a rest position with both arms down (also called anatomical position in the motor rehabilitation context). When he presses the start button, the button text changes from *Start* to *Recording* on the main screen, and he starts to perform the chosen movement. Once he finishes by pressing the *Finish* button in the second device, the sample is stored.

3.2 Types of movements

The first defined movements for classification involved the articulation of the shoulder and the elbow. The names of the four movements are *Coronal*, *Sagittal*, *Transverse*, and *Elbow*. Three of them represent the upper limb moving in one of the anatomical planes, and the *Elbow* movement comprises a flexion of the elbow followed by an extension.

Anatomical planes transect the body in two portions, front and back, left and right, and top and bottom. *Coronal*, *Sagittal*, and *Transverse* represent movements in which the arm moves mainly along one of the planes (**Figure 2**).

Movements related to anatomical planes vary in two ways: it can be a *Full* movement when the user reaches the maximum of the range of motion of these movements, or when the range of movement is only 90º, it is called a *Half* movement.

Movements named *Right Coronal Full* start with the arm

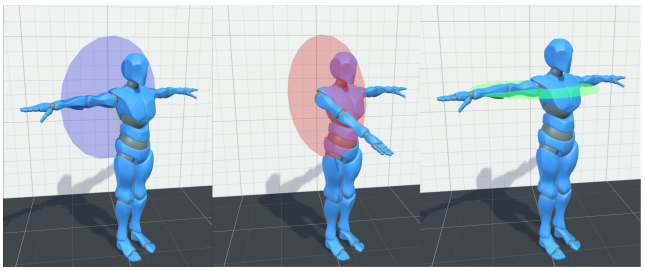

Figure 2. Coronal (left), Sagittal (center) and Transverse (right) anatomical planes. The arm moves along the highlighted area and create distinct movements.

pointing down; then, the arm moves along the coronal plane until it points up; and goes back to the initial position. While movements named *Right Coronal Half* also have the same initial and end position, but the arm only reach the middle of its range in the coronal plane (**Figure 3**).

The user not used to the anatomical planes terminology can still memorize *Coronal* movements as raising the arm to the side and *Sagittal* as raising the arm forward (**Figure 3**). The *Transverse* movements require an arm raise in the Coronal plane until 90º at the beginning of the movement. Then the arm should move horizontally in the Transverse plane for approximately 180º (with the arm close to the chest) in the *Transverse Full* or 90º in the *Transverse Half* (**Figure 3**).

The *Elbow* movements also have two classes related to the range of motion, separated by the angle at which the movement reached. They are named *Elbow Full* and *Elbow Half* (**Figure 3**).

Some movements draw a *Circle* in the air with the upper limb. It is more complex from an anatomical plane's point of view but intuitive from a user's point of view. In an interactive system controlled with arms, for example, the gesture can be used as a forward command when the *Circle* has a *Clockwise* direction and a backward command when it is a *Counterclockwise* one (**Figure 3**).

We also added lower limbs to the dataset to test all body articulations in the model input. The defined movements were the *Kick* and *Knee Raise*, similar to one of the Kinect datasets of the related work in subsection 2.2. We choose the *Kick* and *Knee Raise* (**Figure 3**) since there are soccer applications where the user simulates the kick of the ball, and the research group is developing a system where the user walks in a virtual environment at the same time he performs a gait in the real world. One step in the walk is equivalent to the movement with the name *Knee Raise*.

The sum of all categories (six anatomical planes, two elbows, two circles, and two lower limbs) results in 12 movements. Movements use either the right (**Figure 3**) or the left limb. The same category made with a different side of the body is considered a different movement, doubling the numbers to 24 classes in total, 12 movements with the right side of the body and 12 with the left side.

3.3 Skeleton data and preprocessing

A movement has an intrinsic dimension – time. It is not easy to imagine a movement given only an image, with just a point in time. Therefore, multiple moments are necessary to have sufficient information to infer which movement the user is performing. This task will require a video (multiple frames) instead of a picture (one frame), and the skeleton data will have the same amount of frames.

The devices used in this study to extract information about the body have a fixed frequency of 30 frames per second. So, for example, a movement that took 10 seconds contains 300 frames. Unlike video information, which contains an image for each frame, the information of each frame of the tracking device is only 33 body key points, which is much lighter than images.

The initial Pose Tracking data returns 33 points with the origin relative to the camera frame. However, raw data is not sent directly to the machine learning model. The system transforms the data to have relative values by using rotations of the articulation calculated by the combination of two joints described below:

- Elbow rotation: Calculated using the point *elbow* and *wrist*.
- Shoulder rotation: Calculated using the points *shoulder* and *elbow*.
- Hip rotation: Calculated using the points *hip* and *knee*.
- Knee rotation: Calculated using the points *knee* and *ankle*.

In the database, each articulation of a frame has the information represented as a data structure of three floating-point values, e.g. [12.50, 91.00, 360.00], known as Euler angles.

The rotation format keeps relative value between the points, similar to the difference between points seen in related work (Choubik and Mahmoudi, 2016; Ijjina and Mohan, 2014). Choubik and Mahmoudi (2016) use the difference between the joints to classify the person's pose. For example, values calculated for variable d that means the distance between shoulder and wrist would result in $d = 100px$ if the person is on the left or on the right side of the frame (**Figure 4**), which is correct since the position relative to the camera does not influence which pose the person is doing. The study discusses why the original values of x and y (relative to the camera frame) should not be the model input. Their implementation converts raw values to values relative to the body, with the difference in pixels from joint A to joint B. It can classify poses of any user, whatever his size and position in the scene (Choubik and Mahmoudi, 2016).

In order to improve the consistency already obtained by differences between joints, this work uses the rotation angles because the values will also be the same independently of the person's position in the scene. It has one more benefit since the distance gets smaller when the user is far from the camera, but the angle value remains the same. For example, when the person stays in the center but far away, the system computes the same angle compared to the same pose in closer locations, while the distance d in pixels decreases (**Figure 4**).

Euler angles represent a determined position or rotation of the articulation. It is formed by three angles in different directions (**Figure 5**), and our system stores the data in this format. It contains all the necessary information to know if the body part is pointing forward or not, pointing up or down at a determined point in time. The order of the axis in rotation may differ in other systems. There is more than one way to

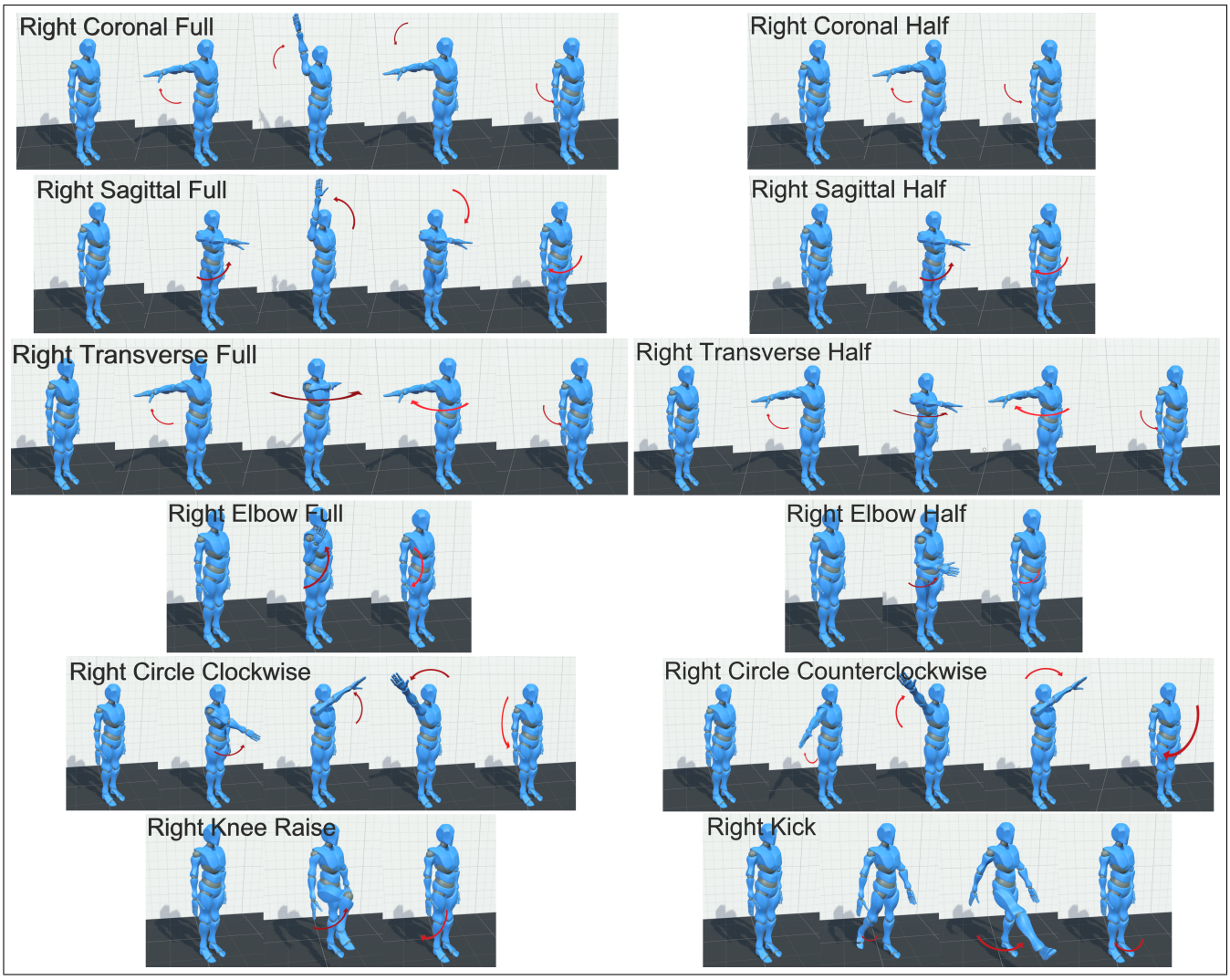

Figure 3. Sequence of 12 movements related of right side of the body (both sides have 24 movements in total).

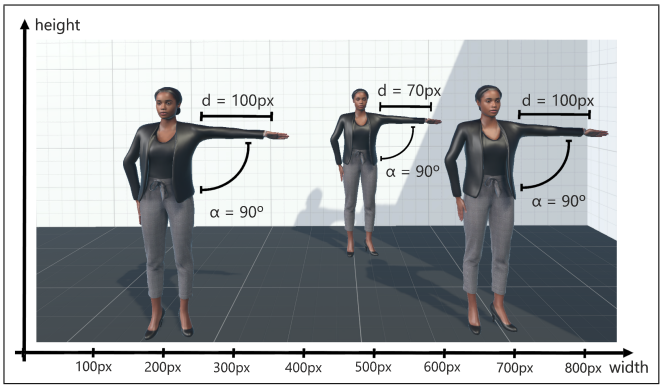

Figure 4. The same pose in different positions of a frame is represented by different values in pixels but equal values in angles, keeping the representation consistent.

reach the same position if the order of the axes is changed. This ambiguity does not happen in our case since the data come from the same system.

We store the database angles in the 0 to 360 range. Also, values were normalized to make the machine learning model train faster. That means that the model's input has its range transformed to a minimum of 0 and a maximum of 1. The model could not converge with the same set of hyperparameters during training without the normalization. We also reduced movement samples that have more than 90 frames before going to the input layer without losing information since it maintains the characteristics of the movement.

The change in the number of frames happens due to a fixed input size requirement for the current model. We transformed the recorded samples to have 90 frames, resulting in an equivalent sample of three seconds of recording. If a movement is less than three seconds, it is "stretched" to fit in 90 frames; If the recording has more than three seconds, we combine the frames until the frames total is 90. For example, if an angle list has 180 frames, a new list of 90 will be created by taking the average of values two by two. This average step depends on the ratio between the length of the recorded movement and the fixed frame number (90).

The transformation still keeps the same movement pattern as we can see in one of the training samples where a *Right Coronal Half* movement lasted four seconds, or 120 frames, in **Figure 6**, and then was reduced to 90 frames. **Figure 7** with 90 frames shows similar curves to the original sample after the average process, and the x-axis changed from 120 to 90. The y-axis also changes due to the already mentioned

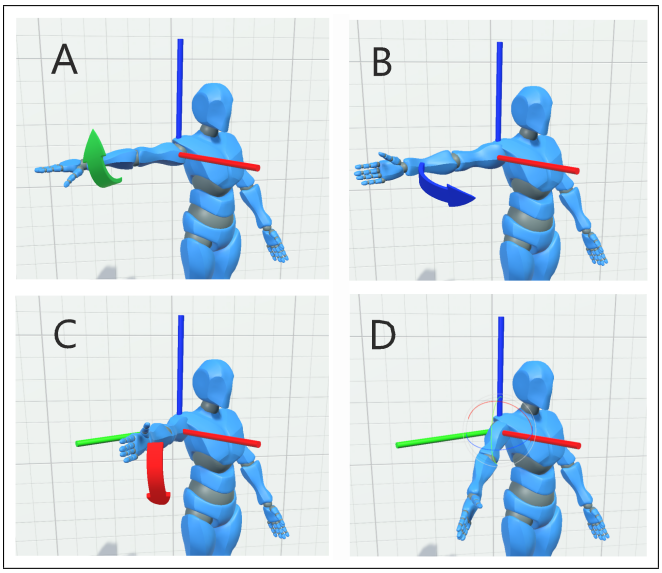

Figure 5. The combination of three angular rotations (A,B,C) can produce any position for the Right Shoulder articulation (D).

normalization.

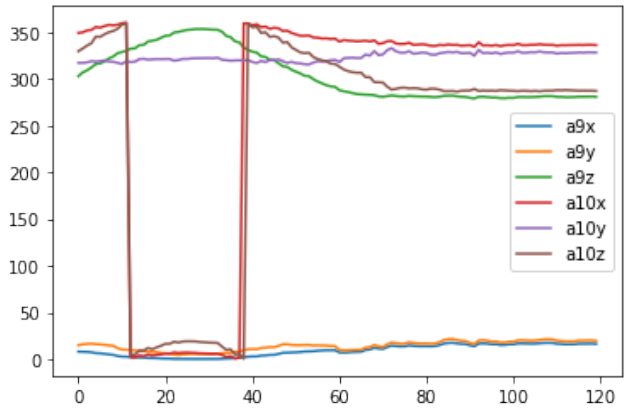

Figure 6. The change of 6 rotation angles of a movement over time, 4 seconds resulted in 120 frames and range from 0 to 360 degrees.

3.4 Machine learning model

The input vector was already introduced partially in **Figure 7** because *a9x*, *a9y*, *a9z*, *a10x*, *a10y*, and *a10z* are 6 of the 24 feature vectors of the model input. As described in subsection 3.3, the system transforms points of the pose tracking algorithm into rotations, specifically in Euler angles (**Figure 5**). We named the articulations of elbows, shoulders, hips, and knees as *a5,a6,a9,10,a13,a14,a17, and a18* according to the id of the body point (**Figure 8**).

Furthermore, the Euler representation for each articulation creates three values for each articulation. The representation decomposes the final rotation into three rotations along each axis, x, y, and z, in the Unity 3D space. **Figure 8** shows all articulations present in the model matrix input with a purple arrow showing the resulted pose and 3-axis origin colored in red, green, and blue that represents the origin of angles, equivalent to **Figure 5**, but now, for all used articulations.

The first calculated angle *a9x* is an array of size 90 due to the time dimension, and it turns out to be the first row of

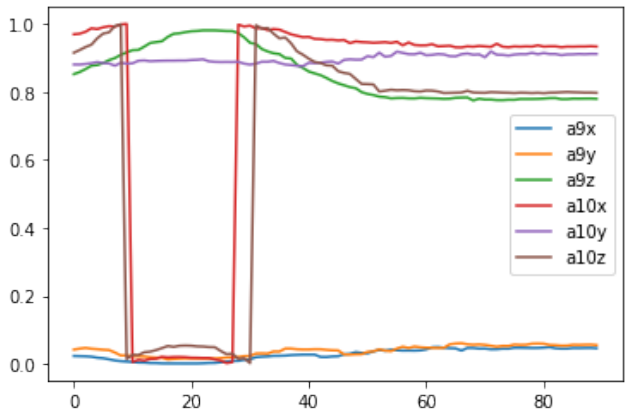

Figure 7. Rotation angles transformed to a fixed number of frames (reduced to 90 frames and normalized from 0 to 1).

the input matrix. The model's input is a matrix where each row represents one Euler angle, and the 90 columns represent the reduced frames. There are 24 rows since each of the eight articulations has three Euler angles, which are named as follows:

- a9x: Right Shoulder - Rotation around the Roll Axis
- a9y: Right Shoulder - Rotation around the Pitch Axis
- a9z: Right Shoulder - Rotation around the Yaw Axis
- a10x: Right Elbow - Rotation around the Roll Axis
- a10y: Right Elbow - Rotation around the Pitch Axis
- a10z: Right Elbow - Rotation around the Yaw Axis
- a5x: Left Shoulder - Rotation around the Roll Axis
- a5y: Left Shoulder - Rotation around the Pitch Axis
- a5z: Left Shoulder - Rotation around the Yaw Axis
- a6x: Left Elbow - Rotation around the Roll Axis
- a6y: Left Elbow - Rotation around the Pitch Axis
- a6z: Left Elbow - Rotation around the Yaw Axis
- a17x: Right Hip - Rotation around the Roll Axis
- a17y: Right Hip - Rotation around the Pitch Axis
- a17z: Right Hip - Rotation around the Yaw Axis
- a18x: Right Knee - Rotation around the Roll Axis
- a18y: Right Knee - Rotation around the Pitch Axis
- a18z: Right Knee - Rotation around the Yaw Axis
- a13x: Left Hip - Rotation around the Roll Axis
- a13y: Left Hip - Rotation around the Pitch Axis
- a13z: Left Hip - Rotation around the Yaw Axis
- a14x: Left Knee - Rotation around the Roll Axis
- a14y: Left Knee - Rotation around the Pitch Axis
- a14z: Left Knee - Rotation around the Yaw Axis

The two-dimensional input is connected to the first convolutional layer of the model, the result passes to a second convolutional layer, and the subsequent three layers, including the output, are fully connected. The sequence of the layers in the model with convolutional blocks followed by dense layers (**Figure 9**) also appears in deeper model architectures, such as AlexNet (Krizhevsky et al., 2017) and VGG (Simonyan and Zisserman, 2014).

The input layer has a 24×90 shape. Moreover, the next layers of the model, which contain the model parameters, are configured as follows:

- The first 1D convolutional block has 64 filters and a

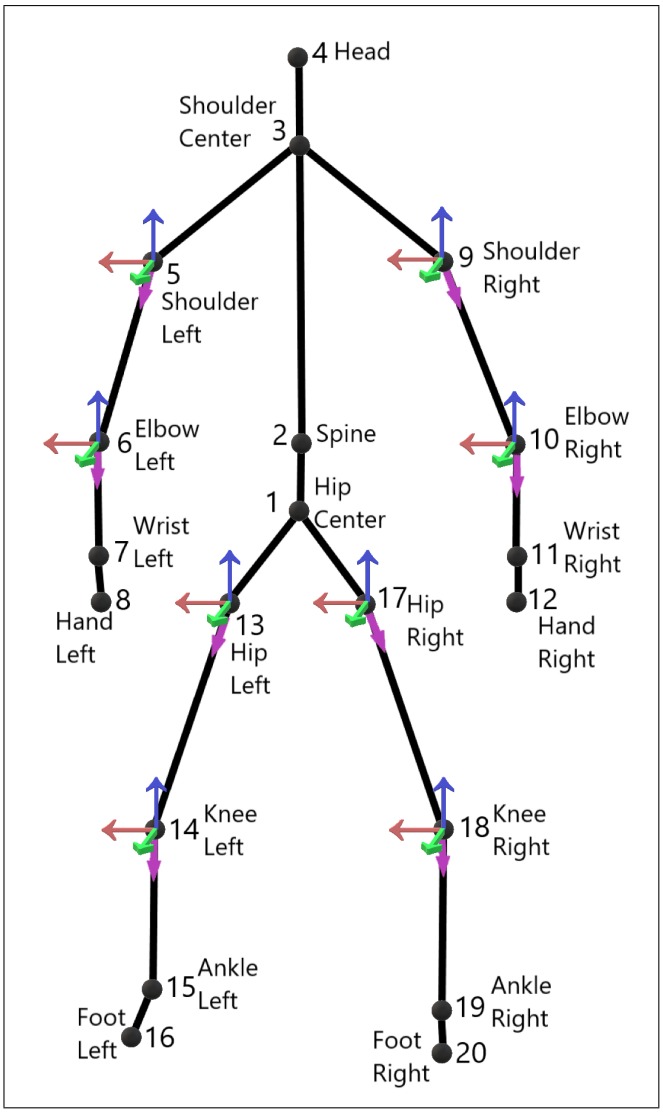

Figure 8. Origin of the rotation of all used articulations in red, green and blue. Purple represents the current position of the articulation.

kernel size of 5, an average pooling of size 2, and ReLu as the activation function;
- The second 1D convolutional block has 128 filters and a kernel size of 5, an average pooling of size 2, and ReLu as the activation function;
- In the third layer, the two-dimensional shape is flattened into one-dimensional to fit into a fully connected layer with 64 nodes and ReLu as the activation function;
- The fourth layer is also a fully connected layer with 64 nodes and ReLu as the activation function;
- Lastly, the output layer is fully connected with the previous one. It has 24 nodes, and the chosen activation function is Softmax, which results in an output of probabilities that add up to 100% distributed among the classes.

3.5 Dataset

We instantiated a database to store the skeleton data tracked in the main system. It also includes movement labels and metadata about the recordings. The samples were stored in the database with the dataset name in the description field to query all movements later for the model training.

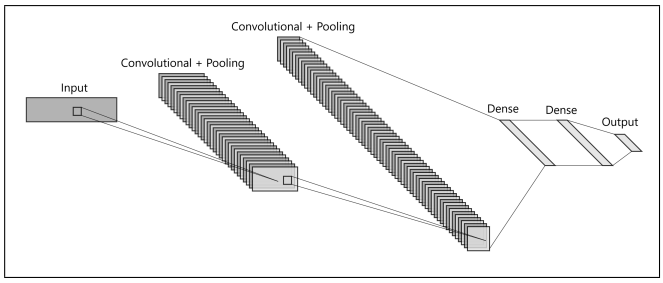

Figure 9. Illustration of the machine learning model with a matrix as input and 5 layers.

One of the authors recorded 240 movements a day on average. The dataset has 2400 movements, with 100 recordings of each movement type. We used 100 samples of each class to keep the dataset balanced.

An essential practice for developing supervised machine learning models is to divide the dataset into train and test samples. The training group is responsible for learning the model parameters and the testing group for computing the loss and accuracy of the model. Therefore, we divided the recorded movements of the dataset into:

- Train set: 1440 movements (60%);
- Test set: 960 movements (40%);

3.6 Inference scenario

The main difference between the training and the inference is the labeling. After the convergence of the model, with parameters adequately fitted, the model automatically names new samples. Thus, the text input in the secondary smartphone is not necessary anymore.

Since the text input is used only for the labeling and training process, we set it to “Automatic” (In Portuguese, *Automático*). **Figure 10** shows the current implementation of the main system when using the trained model to classify movements. A person does any of the determined movements after the recording has started. While the user performs the movement, he can see the elapsed time in the secondary phone, the text message “Recording” (In Portuguese, *Gravando*), and the button “Finish” (In Portuguese, *Finalizar*) to press when the movement ends.

At the end of the movement, the remote server receives the recorded skeleton data through an API that waits for the new movement. The server stores and opens the trained model with a deep learning framework. The endpoint receives a list of the Euler angles, the script written in Python preprocesses the list, and the inference of the deep learning framework happens. The index of the highest probability in the 24 softmax result matches the index of a list of all movement names, and this text is the API response.

Meanwhile, the primary system presents a message in the center of the screen that says “Processing...” that lasts while the response does not arrive. Then, the text is changes to “Movement: ” (In Portuguese, *Movimento:*) concatenated with the API result.

When the movement ends, the current implementation sends the recording with no label to the API of a remote server that returns the predicted name, and the system shows

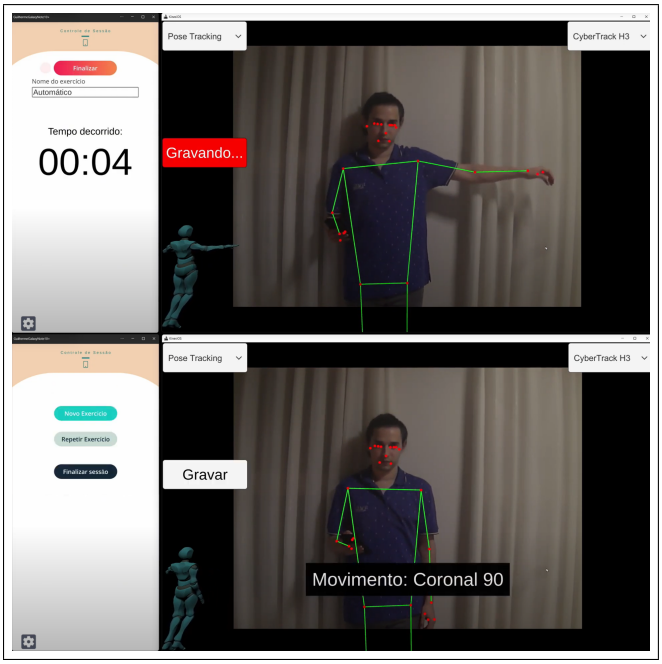

Figure 10. Screenshots of a second phone for remote control (in the left) and the main system classifying a movement (in the right).

the result on the main screen. That means the application requires internet access.

The trained model works in the Tensorflow platform. However, it is still possible to use it directly in the application for offline inference through a conversion to the ONNX[2] (Open Neural Network Exchange) format, which is compatible with Unity and keeps the architecture and the weighed parameters of the original model.

One feature we will implement in the future is to remove the start button and detect automatically when the user is resting in an anatomical position by using simple rules. When the user leaves this position, the recording starts until the user returns to the anatomical position.

4 Results

The trained model achieved good results in the test set, with similar precision between the upper and lower body classes. We also noticed that the relatively more complex circular movement, which passes through the three anatomical planes simultaneously, is also capable of being differentiated by the classifier.

The classifier system receives the eight articulations of the body by capturing and preprocessing the data even when the movement involves only one or two articulations. That shows that the model can handle noise from articulations that are not being used at the moment and are activated only during related movements. When the user performs a movement related to the right arm, the leg articulations do not interfere with the result, even if the person makes subtle movements to balance while standing.

The test set with 960 samples contains 40 movements of each class (lines of **Table 1**). The model predicts the majority of the samples as the corrected class, this inferences are considered true positives, and the detailed values are present in the diagonal line in confusion matrix (**Table 1**).

On average, the model is corrected 90% of the time, two classes reached 100% in the precision metric, and one class reached 100% in the recall metric (**Table 2**).

The most wrongly classified samples had the real class *Sagittal Half*, but the model predicted as *Elbow Half* for both arms (**Table 1**). In *Sagittal Half* movements, the user points the hand to the camera and occludes both the shoulder and elbow articulations. Although the arm is straight and pointing forward, the pose tracking algorithm wrongly tracked as if the hidden elbow articulation was a little lower than the actual position, possibly creating confusion with the *Elbow* classes.

Occlusion of the elbow articulation can affect the *Sagittal* classes negatively. In order to mitigate the error, the dataset could include new samples with the camera slightly moved, so the elbow stays visible when the user moves the arm forward.

The four classes related to the lower limb reached precision values close to or above average precision (**Table 2**). The false positives and false negatives of these classes still belong to the lower limb classes, e.g., four false negative samples in the *Left Knee Raise* classified as *Left Kick* (**Table 1**).

Also, new classes related to Circular movements have reached precision close to or above average (**Table 2**), but with seven false positives each in *Right Circle Clockwise* and *Right Circle Counterclockwise* (**Table 1**) resulting in a recall below average in the right side of the body.

The model was able to generalize the movements of people outside the training set in the previous work (Rodrigues et al., 2021), while this work does not focus on this test. The humanoid shown on the left side of the screen (**Figure 1**) has feedback on the tracking and storing because it receives the same rotation used in the model. When a different person uses the system with a different camera and device, the system still moves the humanoid on the screen and runs inferences. Despite that, adding more subjects to the dataset and other devices is still relevant for future work.

5 Conclusions

This extended version of the work done by Rodrigues et al. (2021) presents a broader implementation of body movements classification using the same system. We described how to use the skeleton data for a supervised classification when multiple articulations are involved. The mobile application supported by the Pose Tracking technology allows developers and users to access this data for motor-rehabilitation applications and take more machine learning approaches. The work of Rodrigues et al. (2021) classifies movements of the right shoulder articulation only, while this extended version classifies body movements from 8 articulations:

Developing a new model input with all articulations was essential to creating the current 24-classes dataset. Now, the extension of the movement types is possible with no modification since eight articulations can cover movements with any arm or leg, or even new movements that use more than

[2]https://onnx.ai/index.html

	Left Coronal Full	Left Coronal Half	Left Sagittal Full	Left Sagittal Half	Left Transverse Full	Left Transverse Half	Left Elbow Full	Left Elbow Half	Left Circle Clockwise	Left Circle Counterclockwise	Left Kick	Left Knee Raise	Right Coronal Full	Right Coronal Half	Right Sagittal Full	Right Sagittal Half	Right Transverse Full	Right Transverse Half	Right Elbow Full	Right Elbow Half	Right Circle Clockwise	Right Circle Counterclockwise	Right Kick	Right Knee Raise
Left Coronal Full	38	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Left Coronal Half	1	38	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Left Sagittal Full	0	0	38	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Left Sagittal Half	0	0	1	28	0	0	1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Left Transverse Full	0	0	0	2	34	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Left Transverse Half	1	1	0	2	2	33	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Left Elbow Full	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Left Elbow Half	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Left Circle Clockwise	0	0	3	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Left Circle Counterclockwise	0	0	3	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Left Kick	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	1	0	0	0	0	0	0
Left Knee Raise	0	0	0	0	0	0	0	0	0	0	4	36	0	0	0	0	0	0	0	0	0	0	0	0
Right Coronal Full	0	0	0	0	0	0	0	1	0	0	0	0	37	1	1	0	0	0	0	0	0	0	0	0
Right Coronal Half	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	2	1	0	0	0
Right Sagittal Full	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	1	0	0	0
Right Sagittal Half	0	0	0	0	0	0	0	0	0	0	1	0	0	2	3	26	0	0	1	5	2	0	0	0
Right Transverse Full	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	33	3	1	0	0	0	0	0
Right Transverse Half	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	39	0	0	0	0	0	0
Right Elbow Full	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	3	0	0	0	0
Right Elbow Half	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	35	0	0	0	0
Right Circle Clockwise	0	0	0	0	0	0	0	0	0	0	0	0	2	3	2	0	0	0	0	0	33	0	0	0
Right Circle Counterclockwise	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	1	0	0	0	0	0	33	0	0
Right Kick	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	2
Right Knee Raise	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	39

Table 1. Confusion matrix of the test set

one limb simultaneously, e.g., clapping sit-to-stand, jumping-jacks.

The presented dataset's size with an RGB camera was similar to related studies that created a dataset with Kinect. The processing approaches and model architectures of the Kinect literature can be studied and compared with the RGB Pose Tracking approach since the skeleton data is very similar.

Implementing the data preprocessing algorithms has shown to be a viable solution for reducing motion capture data. In order to shape the data into the machine learning model input, we transformed the raw data into 24×90 nodes, and it did not require high computational power.

The occlusion problem affected the detection of the *Sagittal* class However, a possible solution is to place the camera where it can see the occluded articulation and have better tracking results. Optical tracking will the affected by the occlusion, but it still can be minimized with the development of more precise inference models.

The resulting trained model has predetermined movements that already fit in healthcare applications, such as a repetitions counter for physical exercises and detecting commands using the body as the interface. For example, a serious game can give 10 seconds for the player to perform the movement indicated on the screen. The player scores if the system classifies the performed movement as the same one shown on the screen.

This work can be extended to receive hands and fingers data with RGB camera pose tracking, and facilitates the study of gestures recognition for human-computer interfaces and sign language recognition.

In future work, we intend to use our tool to facilitate the development of motor-rehabilitation applications to assist the professional while the patient does physical exercises.

Metric	Precision	Recall
Formula	TP/(TP+FP)	TP/(TP+FN)
Left Coronal Full	95%	95%
Left Coronal Half	97%	95%
Left Sagittal Full	81%	95%
Left Sagittal Half	85%	70%
Left Transverse Full	94%	85%
Left Transverse Half	92%	83%
Left Elbow Full	95%	100%
Left Elbow Half	78%	98%
Left Circle Clockwise	97%	93%
Left Circle Counter Clockwise	95%	93%
Left Kick	89%	98%
Left Knee Raise	100%	90%
Right Coronal Full	95%	93%
Right Coronal Half	86%	93%
Right Sagittal Full	75%	98%
Right Sagittal Half	84%	65%
Right Transverse Full	97%	83%
Right Transverse Half	89%	98%
Right Elbow Full	90%	93%
Right Elbow Half	76%	88%
Right Circle Clockwise	89%	83%
Right Circle Counter Clockwise	100%	83%
Right Kick	97%	95%
Right Knee Raise	95%	98%
Average:	**90.50%**	**89.90%**

Table 2. Precision and Recall of the Classes

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoas de Nível Superior (CAPES-Brazil)

This project has been partially supported by Huawei do Brasil Telecomunicações Ltda (Fundunesp Process # 3123/2020).

References

Aung, Y. M. and Al-Jumaily, A. (2012). Shoulder rehabilitation with biofeedback simulation. In *2012 IEEE International Conference on Mechatronics and Automation*, pages 974–979, Chengdu, China. IEEE.

Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., and Grundmann, M. (2020). Blazepose: On-device real-time body pose tracking. In *CVPR Workshop on Computer Vision for Augmented and Virtual Reality*, Seattle, WA, USA. arXiv.

Bloom, V., Argyriou, V., and Makris, D. (2016). Hierarchical transfer learning for online recognition of compound actions. *Computer Vision and Image Understanding*, 144(C):62–72.

Borich, M. R., Wolf, S. L., Tan, A. Q., and Palmer, J. A. (2018). Targeted Neuromodulation of Abnormal Interhemispheric Connectivity to Promote Neural Plasticity and Recovery of Arm Function after Stroke: A Randomized Crossover Clinical Trial Study Protocol. *Neural Plasticity*, 2018.

Breedon, P., Byrom, B., Siena, L., and Muehlhausen, W. (2016). Enhancing the measurement of clinical outcomes using microsoft kinect. In *2016 International Conference on Interactive Technologies and Games (ITAG)*, pages 61–69, Nottingham, UK. IEEE.

Choubik, Y. and Mahmoudi, A. (2016). Machine learning for real time poses classification using kinect skeleton data. In *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, pages 307–311, Beni Mellal, Morocco. IEEE.

Dimyan, M. A. and Cohen, L. G. (2011). Neuroplasticity in the context of motor rehabilitation after stroke. *Nature Reviews Neurology*, 7(2):76–85.

Dinevan, A., Aung, Y. M., and Al-Jumaily, A. (2011). Human computer interactive system for fast recovery based stroke rehabilitation. In *2011 11th International Conference on Hybrid Intelligent Systems (HIS)*, pages 647–652, Melacca, Malaysia. IEEE.

Farooq, A. and Won, C. S. (2015). A survey of human action recognition approaches that use an rgb-d sensor. *IEIE Transactions on Smart Processing and Computing*, 4(4):281–290.

Ijjina, E. P. and Mohan, C. K. (2014). Human action recognition based on mocap information using convolution neural networks. In *Proceedings of the 2014 13th International Conference on Machine Learning and Applications*, ICMLA '14, page 159–164, USA. IEEE Computer Society.

Jaffe, D. L. (2003). Using augmented reality to improve walking in stroke survivors. In *The 12th IEEE International Workshop on Robot and Human Interactive Communication (ROMAN)*, pages 79–83, Millbrae, CA, USA. IEEE.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

Leightley, D., Yap, M. H., Coulson, J., Barnouin, Y., and McPhee, J. S. (2015). Benchmarking human motion analysis using kinect one: An open source dataset. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–7.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., and Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172.

Mehta, U. M. and Keshavan, M. S. (2015). Cognitive Rehabilitation and Modulating Neuroplasticity with Brain Stimulation: Promises and Challenges. *Journal of Psychosocial Rehabilitation and Mental Health*, 2(1):5–7.

Mukherjee, D., Levin, R. L., and Heller, W. (2006). The cognitive, emotional, and social sequelae of stroke: psychological and ethical concerns in post-stroke adaptation. *Topics in stroke rehabilitation*, 13(4):26–35.

Phan, H. L., Le, T. H., Lim, J. M., Hwang, C. H., and Koo, K.-i. (2022). Effectiveness of augmented reality in stroke rehabilitation: A meta-analysis. *Applied Sciences*, 12(4).

Rego, P., Moreira, P. M., and Reis, L. P. (2010). Serious games for rehabilitation: A survey and a classification towards a taxonomy. In *5th Iberian Conference on Information Systems and Technologies*, pages 1–6, Santiago de Compostela, Spain. IEEE.

Rodrigues, L. G. S., Dias, D., Guimaraes, M. d. P., Brandao, A. F., Rocha, L., Iope, R. L., and Brega, J. R. F. (2021). Classification of human movements with motion capture data in a motor rehabilitation context. In *Symposium on Virtual and Augmented Reality*, SVR'21, page 56–63, New York, NY, USA. Association for Computing Machinery.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Toh, A., Jiang, L., and Lua, E. K. (2011). Augmented reality gaming for rehabhome. In *Proceedings of the 5th International Conference on Rehabilitation Engineering & Assistive Technology*, i-CREATe '11, Midview City, SGP. Singapore Therapeutic, Assistive & Rehabilitative Technologies (START) Centre.

World Stroke Organization (2019). Global stroke fact sheet. International Journal of Stroke.