UNIVERSIDADE D
COIMBRA

Tânia Daniela Carvalho Gonçalves

Skeleton Based Motion Analysis and Assessment for Telerehabilitation

Master's Dissertation in Electrical and Computer Engineering, supervised by Professor Dr. Paulo José Monteiro Peixoto and presented to the Faculty of Sciences and Technology of the University of Coimbra.

September 2022

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Skeleton Based Motion Analysis and Assessment for Telerehabilitation

Tânia Daniela Carvalho Gonçalves

Coimbra, September 2022

Skeleton Based Motion Analysis and Assessment for Telerehabilitation

Supervisor:

Prof. Dr. Paulo José Monteiro Peixoto

Co-Supervisor:

Dr. João Luís Ruivo Carvalho Paulo

Jury:

Prof. Dr. Paulo Jorge Carvalho Menezes

Prof. Dr. Gabriel Falcão Paiva Fernandes

Prof. Dr. Paulo José Monteiro Peixoto

Dissertation submitted in partial fulfillment for the degree of Master in Electrical and Computer Engineering.

Coimbra, September 2022

Acknowledgements

Throughout the development of this dissertation, there were moments of victories and others of persistence and sacrifice. Fortunately, at this stage, I was always accompanied by extraordinary people.

I would like to thank Professor Paulo Peixoto and João Paulo for guiding and motivating me during this journey. Thank you for your trust, patience and for the follow-up.

I would like to thank ISR and the entire team, especially Dennis Hesenkamp, for helping and encouraging me during the development of this dissertation.

I would also like to express my gratitude to my friends Cristina Pierdevara, Nuno Marques, Daniel Almeida, Nuno Mendes, Inês Santos, Henrique Oliveira, Mónica Rocha, Jandira Mandinga, Josefa Vieira and José Azevedo for believing in me, for accompanying and encouraging me throughout my academic course, especially at this stage.

I would like to express my thankfulness to my parents, brother and sister for understanding and believing in me.

A special thanks to my best friend Tânia Ribeiro and my cousin Susana Andrade for their encouragement and patience and for not letting me give up on the most difficult days.

Resumo

Atualmente, tem sido cada vez mais importante proporcionar uma reabilitação adequada em casa e determinar estratégias para prevenir lesões devido ao aumento da expectativa de vida, à prevalência de doenças crónicas e ao sedentarismo, para ajudar as pessoas a viver de forma independente e melhorar a sua qualidade de vida. Para possibilitar uma recuperação mais rápida, prática e económica, a reabilitação física do paciente em casa precisa de ser monitorizada e avaliada para fornecer *feedback* ao utilizador.

Deste modo, nesta dissertação propõe-se um sistema de avaliação da postura e do desempenho do utilizador a executar exercícios físicos. O pipeline proposto pode ser dividido em duas partes principais: contagem e segmentação de repetições e a avaliação de cada repetição. Inicialmente, a pose humana é estimada com o *MediaPipe* para um vídeo de CrossFit. Os *landmarks* estimados são a entrada do algoritmo *K-Nearest Neighbors (KNN)* para segmentar os exercícios em repetições. Seguidamente, cada repetição é avaliada pela rede *attentive Bidirectional Long Short-Term Memory (BiLSTM)*. Esta dissertação está a ser desenvolvida no âmbito do projeto *Intelligent Platform for Autonomous Collaborative Telerehabilitation (INPACT)*, que pretende produzir um protótipo de sistema de telereabilitação funcional.

O sistema desenvolvido atinge uma *accuracy* média de 79.21% para a segmentação de exercícios de *Push-Ups* (valor mais baixo) e 93.97% para segmentação de exercícios de *Jumping Jacks* (valor mais elevado). Para a avaliação de desempenho foi implementada uma solução binária e multiclasse, alcançando uma *accuracy* de 95.21% e *F1-Score* de 94.18% para o primeiro caso e uma *accuracy* de 89.51% e *F1-Score* de 61.25% para o segundo caso.

Os resultados comprovam que este sistema tem potencial para uma aplicação doméstica automatizada, sendo prático, simples e flexível, uma vez que requer apenas o uso de uma câmara *Red Green Blue (RGB)* comum por parte do paciente.

Palavras Chave: Reabilitação; Estimação de Pose Humana; KNN; Long Short-Term Memory (LSTM); Mecanismo de Auto-Atenção.

Abstract

With increased life expectancy, prevalence of chronic diseases, and sedentary lifestyles, it is becoming increasingly important to provide appropriate home rehabilitation and establish injury prevention strategies to help people live independently and improve their quality of life. In order to provide a faster, more practical and affordable recovery, the patient's physical rehabilitation at home needs to be monitored and assessed to provide feedback to the user to improve his ability and perform exercises correctly.

Thus, in this dissertation, a system for posture and performance evaluation of human locomotion is proposed. The proposed pipeline can be divided into two main parts: Repetition Counting and Segmentation and Performance Evaluation. First, given an input video, the human pose in each frame is estimated using MediaPipe BlazePose. The estimated landmarks are the input to the KNN algorithm for exercise segmentation and counting. Then, each repetition is fed into an attentive BiLSTM to evaluate performance. This dissertation is being developed as part of the INPACT project, which aims to produce a functional telerehabilitation system.

The proposed pipeline manages to achieve a mean accuracy of 79.21% for exercise repetition segmentation of Push-Ups (lowest value) and 93.97% for exercise repetition segmentation of Jumping Jacks (higher value). For performance evaluation we implemented a binary and a multi-class solution, achieving an accuracy of 95.21% and F1-Score of 94.18% for the first case and an accuracy of 89.51% and F1-Score of 61.25% for the second one.

The results show that this system has potential for automated domestic application, being practical, simple, and flexible since it only requires the use of an ordinary RGB camera.

Keywords: Rehabilitation; Human Pose Estimation; KNN; LSTM; Self-Attention Mechanism.

"All progress takes place outside the comfort zone."

— Michael John Bobak

Contents

List of Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
BN	Batch Normalization
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional neural network
FC	Fully Connected Layer
HPE	Human Pose Estimation
INPACT	Intelligent Platform for Autonomous Collaborative Telerehabilitation
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
ML	Machine Learning
RGB	Red Green Blue
RNN	Recurrent Neural Networks
ROI	Region Of Interest
TSM	Temporal Self-similarity Matrix

List of Figures

List of Tables

1 Introduction

In this chapter, we provide a general contextualization of the research that addresses the project's problem, the necessity for its development, and its objectives and contributions.

1.1 Motivation and Context

Physiotherapy as a rehabilitation process is fundamental for the recovery, treatment and prevention of injuries or dysfunctions in the skeletal muscle complex. Therefore, treatment with rehabilitation exercises aims to strengthen and achieve a quality of movement. To accomplish this, it is necessary to monitor the body alignment and the efficiency of the exercises performed. In addition, the duration, intensity and frequency of the exercises must be defined according to the patient's problem and regularly adapted according to the patient's evolution [56].

Rehabilitation of patients is usually carried out in clinics. However, it would be desirable to perform it at home to avoid trips to clinics and thus additional costs. Since a specialist is not always available, automated systems must be developed to monitor the patient at home during exercises [4].

It is crucial to assess whether the exercises are performed correctly, for the motivation and evolution of patients. Therefore, it is proposed a method for evaluating the performance of patients executing rehabilitation exercises using machine and deep learning algorithms [4]. This dissertation is part of the INPACT project whose goal is to develop a telerehabilitation system for autonomous monitoring and evaluation of rehabilitation exercises.

In order to create a model that can analyze human motion, data must be collected using a motion capture system. One possible approach is motion capture with marker-based suits, which are very accurate but expensive, require a complex setup, assume a controlled environment, and are impractical for home use [54].

The proposed solution will rely solely on an ordinary camera to acquire an image of the

user performing the rehabilitation exercises. Machine Learning (ML) algorithms will be used to obtain the skeleton of the user from a regular RGB image. The analysis of the motion of the skeleton joints will enable the assessment of the quality of the exercise by comparing it with examples provided by a professional in the field. To train the algorithms, one needs a dataset with several videos containing repetitions of the exercises to be evaluated. In this dissertation, a dataset with five CrossFit exercises is used. This dataset includes video recordings of 47 participants with temporal and categorical annotations at a frame level. It includes variations of those exercises performed with the most frequent errors users do when performing them. It also contains the ground truth for segmenting each repetition and evaluating the subjects' performance [16].

1.2 Proposed Work Overview

A framework for evaluating human activities was proposed and implemented. The system consists of two core steps: Exercise Counting and Segmentation and Performance Assessment.

In the first step, a KNN algorithm is used to segment the repetitions of the exercise. To do that, initially, the RGB videos are converted to skeleton data using MediaPipe BlazePose. Then, we use the KNN algorithm to segment and count the repetitions.

In the second step, human motion sequences are evaluated using an attentive BiLSTM. To do this, each repetition needs to be pre-processed. The output of this stage will indicate if the exercise was performed well or not, which allows the system to provide feedback to the user.

Figure 1.1 presents a simplified conceptual global overview of the proposed framework. A more detailed diagram and extensive description of the system behavior can be found in chapter 3.

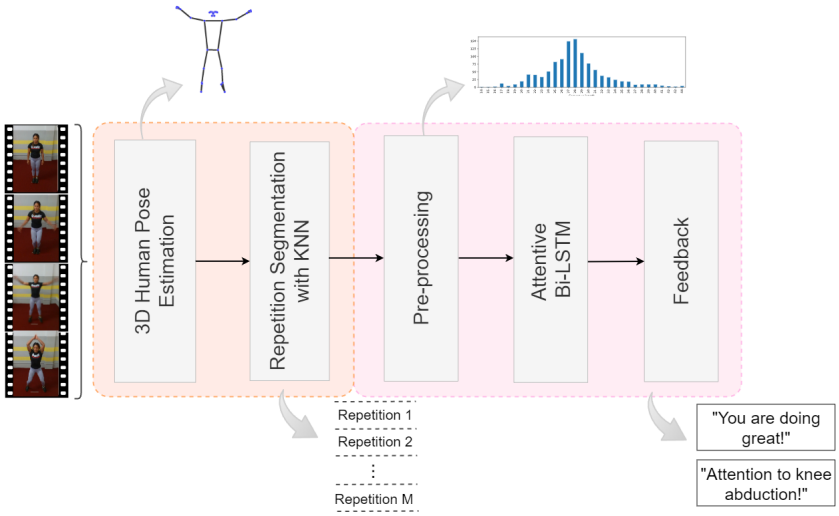


Figure 1.1: Overview of the proposed solution pipeline.

1.3 Objectives

With this work, we intend to propose a system capable of analyzing and giving feedback to the user when performing therapeutic exercises. To achieve this goal, the following specific objectives were outlined:

1. Select and pre-process the dataset that will be used;
2. Estimate the patient's human pose of exercise sequences;
3. Develop a model capable of evaluating the performance of the user executing the exercises;
4. Give feedback to the user on the exercise's performance and suggest improvements if the exercise is performed incorrectly.

1.4 Main Contributions

The present dissertation proposes a method that results in two main contributions, summarized as follows:

1. Select and pre-process a new dataset, covering a few physical exercises with different action speeds, into a set of skeleton landmarks;
2. Implementation of a system combining human pose estimation, exercise repetition counting and segmentation, assessment and feedback generation.

1.5 Outline

This dissertation is organized into six chapters. Chapter 1 presents a general contextualization of the research addressing the project's problem, the justification for the development of the work as well as the objectives. Chapter 2, addresses the state of the art about the different methods available for exercise segmentation and counting and performance analysis of human motion sequences. Chapter 3 describes all the proposed methods, in particular the MediaPipe BlazePose, KNN algorithm and attentive BiLSTM. In chapter 4, all details of the implementation of this project are described. The tests performed and the respective results and analysis of the developed system are described in chapter 5. Finally, chapter 6 presents the main conclusions of the developed system, points out the identified limitations, and proposes new directions for development.

2 State of the Art

Over the last few years, the importance of taking advantage of technological and scientific advances in the maintenance and improvement of the health and physical condition of the population has become increasingly important, specifically for patients with musculoskeletal limitations and injuries. Therefore, analyzing and evaluating patient activity in videos is an increasingly required task, being important in the field of computer vision. This type of technology helps the population to be more independent and have a more peaceful and easier life as it has applications in terms of entertainment, and surveillance and also to facilitate daily tasks [5].

In recent years, the importance of using technological and scientific advances to maintain and improve the health and physical condition of the population has become increasingly important, especially for patients with musculoskeletal limitations and injuries. However, many times, the patient does not have the possibility to attend rehabilitation clinics, giving up or performing the exercises incorrectly and inconsistently, hindering their recovery and well-being. The use of an automated system to assess the patients' rehabilitation at home can remedy this situation by providing more motivation and adapting the exercises to the patient's needs [45] [56].

However, these tasks present some challenges due to the high dimensionality of video data and changes in appearance characteristics (scene, context, and point of view variation) [5]. Machine learning can be extremely useful in this context, as it can be used to learn complex models from data observation. However, in order to train these models, a large dataset must be provided. This dataset must be sufficiently diversified to create a model capable of identifying and evaluating patient performance given the above mentioned factors. A widely used system for capturing human motion from video sequences requires the use of special suits and markers. However, this type of motion capture system is expensive and requires calibration, which is not suitable for a domestic application. It is essential to create a more practical and simple system, that uses an ordinary camera. To overcome some of

these restrictions and limitations, (3D) Human Pose Estimation (HPE) using BlazePose with MediaPipe is used to describe human motion [7]. Thereby, the HPE is used to identify and classify the joints of the human body. In fact, HPE from videos has several applications, such as sign language recognition, augmented reality, motion tracking for videogames and human fall detection [42] [39] [55] [24] [51].

In order to evaluate the exercise sequences and provide the respective feedback, it is necessary to compare the patient's performance executing an exercise with the performance of an expert in the field [33].

Therefore, it is necessary to combine HPE, repetition counting and movement assessment fields to assist people in physical therapy exercise [60].

2.1 Human Pose Estimation and Dataset Selection

Human Pose Estimation - MediaPipe BlazePose

HPE consists of a computer vision technique capable of detecting and locating the key points of the human body from images or RGB frames. This technique is highly used due to its simplicity and stability and has several applications, such as body posture analysis, identification and evaluation of fitness and yoga exercises, gesture recognition and character animation [60] [7].

Several methods can be used for pose detection, such as OpenPose [26], PoseNet [11], Dense Pose [21], BlazePose [7] ... The chosen one was MediaPipe BlazePose. This method is a ML solution, more specifically a lightweight Convolutional neural network (CNN) architecture, that provides high-fidelity human pose tracking, capable of inferring 33 3D landmarks (as in Figure 2.2) from RGB images. Each landmark includes the x, y and z coordinates and visibility. Unlike models that are based on the standard COCO topology [1] [34] (with only 17 key points, as shown in Figure 2.1), BlazePose accurately locates more key points, making it more convenient for fitness and exercise applications. For that reason, the use of MediaPipe BlazePose stands out for being fast and computationally efficient, which allows its use in real-time and on low edge devices, such as mobile or laptop, and supports Android and iOS platforms [7] [47].

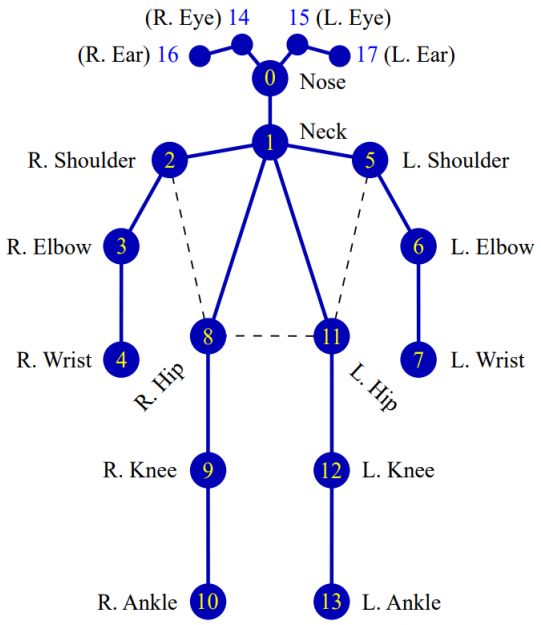


Figure 2.1: COCO Landmarks [34].

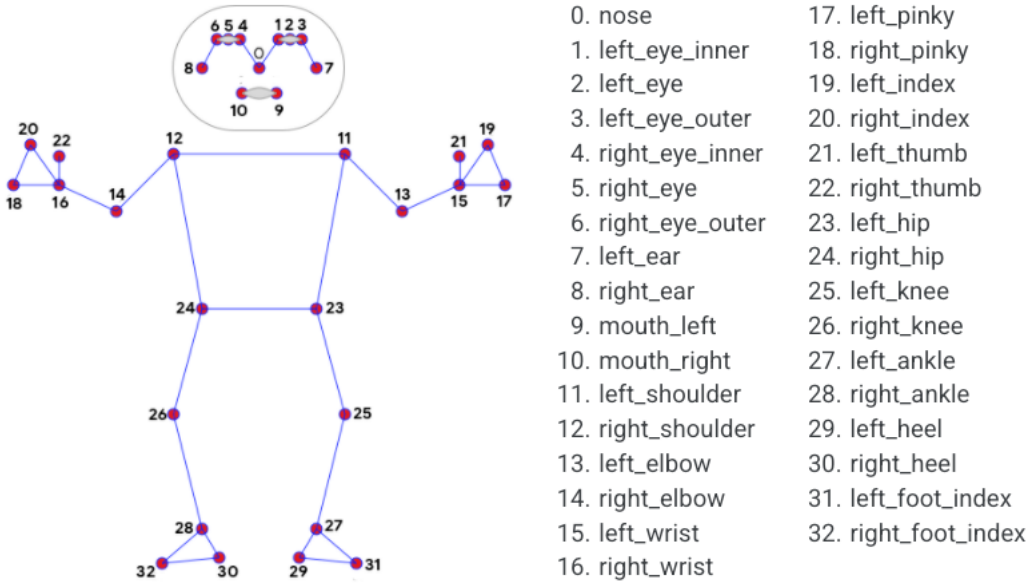


Figure 2.2: MediaPipe Landmarks [2].

Another advantage of using BlazePose is the possibility of obtaining 3D coordinates, this means depth information, using only a Two-Dimensional (2D) camera. Each landmark represents the locations of the body joints of the person in the image, composed of 3D coordinates with the origin at the center of the hips [2].

It is worth noting that pose detection with BlazePose only allows the detection of one person in an image, which means that if there is more than one person in the image, the model will assign key points to the person detected with the highest confidence [7].

Dataset Selection

The segmentation and evaluation methods used in this dissertation require a set of videos that include annotated repetitions of rehabilitation exercises. Since, to our knowledge and research, there are no datasets with these characteristics, some datasets of fitness and CrossFit videos were analyzed, and the most appropriate one, described below, was used in this work.

Ferreira et al., 2021 [16]: A dataset that contains Crossfit videos of 130 subjects performing 5 exercises, and for each exercise there are more than 500 well-performed repetitions. To capture the samples, 4 depth cameras were used, with a resolution of 30 frames per second. All videos were manually annotated in the temporal and categorical domains by an expert physiologist, with each video frame showing the respective pose and, if it is the case, associated errors. The acquisition conditions are diversified, varying, namely, lighting, background disorder, subjects' clothing and changes in the speed of repeated actions.

2.2 Exercise Repetition Counting and Segmentation

In many cases, human activities involve repetitive actions and the performance of rehabilitation exercises is one of them. This means that counting and segmenting temporal repetitions is a crucial step in understanding and analyzing human movements and aims to count the number of repetitive actions in a video [41] [13] [28] [48] [61] [49].

The analysis and segmentation of action repetitions thus offers several possible applications, such as pedestrian detection [46], gesture-based computer interaction, computer gaming [58], cardiac and respiratory signal recovery [30], 3D reconstruction [31] [59] and camera calibration [22].

In the context of this work, it is essential to identify the temporal limits of each repetition of an exercise to assess the quality of movement when performing therapeutic exercises for each session [58]. The segmentation of a video in repetition intervals has been increasingly studied, and there are already some methods proposed in the literature for this purpose.

Segmentation of Class Agnostic Repetitive Actions

For the segmentation of class agnostic repetitive actions, the most used approach is generally divided into two phases: periodicity detection (determining if a frame is part of a repeating action) and repetition counting (predicting the count number of actions in a video) [14] [17].

The robust estimation of periodicity in temporal series is an alternative that allows to delimit the repetition [5]. The detection of periodicity in video sequences can be performed by examining the correlations of spatio-temporal features, creating a Temporal Self-similarity Matrix (TSM) to denote the similarity between frames [25] [14] [60].

TSM is a representation used for recognition and analysis of human action [14]. This matrix is efficient as it incorporates the combination of features with enough descriptive information to characterize the movement [27].

Identifying the periodicity in a video sequence using deep learning models allows counting the repetitions of the action using period duration predictions [17]. This process also allows locating and segmenting each repeating sequence.

The advantage of this approach is that it is suitable for segmenting any type of video (regardless of the particular exercise). The main disadvantage is that it can hardly be adapted to real-time, since it is computationally intensive and in most cases requires the whole video to properly analyze and segment the repetitions.

Segmentation of Specific Physical Exercises

In addition to the methods described above, there are other approaches to perform the segmentation of each video repetition that use skeleton features and don't use any prediction of the periodicity.

A simple alternative to count and segment exercises, even though prone to errors, can be to apply heuristics to the skeleton data. By using this method, it is avoided to create a deep learning model and a dataset, it is only necessary to analyse the coordinates or calculate the angles for the segments that connect the landmarks of interest in the exercise and define the respective conditions [2].

Some methods examine the coordinates of the joints [36] and other approaches measure the angles of the active joints for repetition counting [37] [57]. Figure 2.3 shows in a simplified way some characteristic angles of the squat exercise that needs to be monitored so that it is possible to identify and segment each repetition [62]. The angles indicated in Figure 2.3 can be easily calculated from 2D pose landmarks.

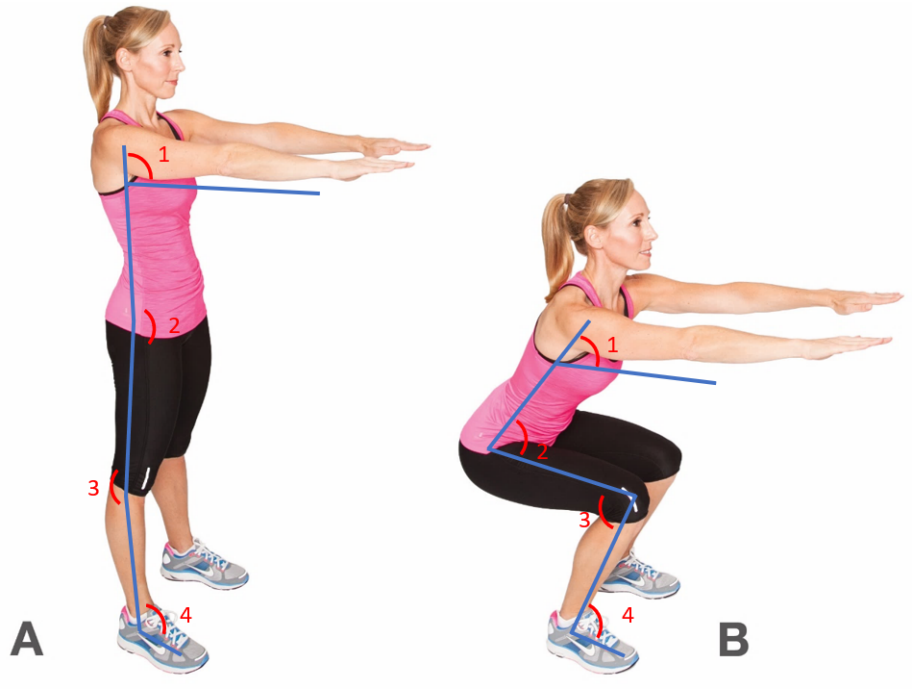


Figure 2.3: Representation of the Squat exercise angle parameters. Terminal states: A - Key Pose Up; B - Key Pose Down. Angles: 1 - Trunk; 2 - Hip; 3 - Knee; 4 - Ankle. [62]

The terminal states (key poses Up and Down) of Figure 2.3 represent the most important poses that the subject must perform during a repetition of the squat exercise. [16]

To recognize the characteristic poses of each exercise, KNN can be used as a classifier because it is simple and easy to use and implement. This algorithm determines the class of the object based on the closest samples from the training set [2] [3] [8]. This algorithm was chosen since it is easy to add new data to the algorithm and it can be adapted to real-time counting. This method is not as powerful as a deep learning model for classification, but it works very well if we are working with a limited number of poses.

2.3 Performance Evaluation of Human Motion

Human motion consists of a combination of translation and rotating movements of each joint in the body. Thus, when comparing human motion between two individuals, it is possible to obtain the similarity of movement, which can be used to analyze and evaluate the performance of a subject to perform a certain task [43]. The most challenging part is comparing two movements with different speeds and variations [29].

Many real-world applications, such as speech recognition and activity recognition, involve the collection of data over time, constituting a Time-Series [19] [40]. Thus, human motion

analysis can be viewed as a time series problem, since both involve an ordered series of observables. Time series analysis examines how the data are structurally dependent [50]. According to Li et al. [32], many consider RNN to be one the most effective technique for time series prediction. In particular, LSTMs are often used in this regard because they are able to capture long- and short-term temporal dependencies [9].

Sliding the time window is generally used to acquire time series prediction features. However, the LSTM capacity to focus on subwindow features across multiple time steps is limited. To overcome this issue, Li et al. [32] and Coskun et al. [12] proposed an architecture of deep learning based on attentive LSTM for time series prediction and human motion analysis, respectively. The attention mechanism allows it to create an embedding that selectively focuses on the semantically descriptive parts of the input sequence.

3 Proposed Method

This dissertation proposes a solution that monitors and provides feedback to the user's performance in real time. It relies on the visual perception of the user's body based on the motion of skeleton joints, without using any markers. It has the ability to analyze the user's performance recurring to deep learning techniques.

3.1 Approach Formulation

The steps for implementation are as follows:

1. Dataset Selection and Pre-processing;
2. 3D Human Pose Estimation with BlazePose;
3. Exercise Repetition Segmentation and Counting with KNN;
4. Performance Assessment using Attentive BiLSTM;
5. Validation and Testing of Implemented Algorithms;
6. Providing Feedback.

3.2 System Overview

The proposed automated system for exercise assessment is represented in Figure 3.1 and takes as input a video of a subject performing physical exercises, converts them to skeleton landmarks, evaluates them and outputs feedback. To achieve this, two main stages were performed as follows:

The objective of the first stage is to segment and count the repetitions. Initially, we input the frames I_t from an exercise video sequence, where I_t is the image input of the I^{th} frame with $t = 1, 2, ..., N$, and use MediaPipe BlazePose for 3D human pose estimation. We obtain

a set of skeleton joints ($P = P_1, P_2, ..., P_N$), each one with 33 landmarks. After that, we fed the keypoints P to KNN model that compares the recurrent pose with a set of samples of the target pose and segment and count the repetitions.

The second stage is responsible to evaluate the subject's performance. First of all, we need to pre-process each repetition of the skeleton joints P. Using Z-score, we did an outlier removal according to the distribution of the length of the repetitions. Then, we add zero padding in order to ensure each repetition has the same length and normalized the data. We need to split the data into training (75%), validation (15%) and testing (15%). After that the data $X = (x_1, x_2, ..., x_n)$ are fed into an Attentive BiLSTM. Batch normalization is added between layers to avoid overfitting and to speed up model convergence, the binary cross entropy is used as the loss function and the AdaGrad algorithm is used as the optimizer. The output of this model will tell us if the exercise is well-performed or not, specifying the error, and according to that we give feedback to the user.

More details about the system operation and the parameters chosen will be given in chapter 4.

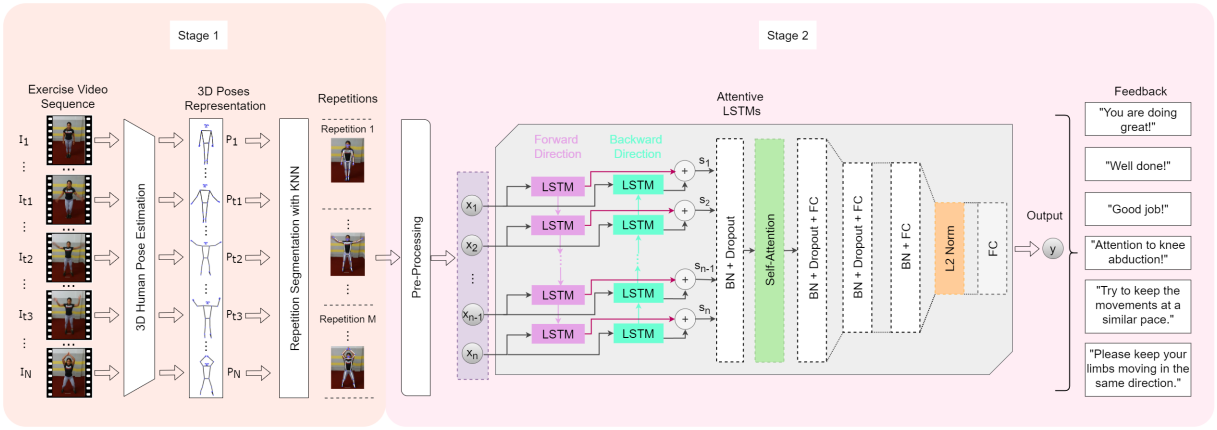


Figure 3.1: Overview of the proposed system architecture.

3.3 Dataset

The selection of informative samples is an important factor for the success and the training speed of the network in deep learning [38]. Thus, the selected dataset was part of Ferreira et al. [16], including 47 participants, performing five exercises (Squats, Burpees, Push-Ups, Sit-Ups and Jumping Jacks). This dataset includes correctly and incorrectly executed movements. A frontal camera perspective is used, with the shooting angle chosen so that most of the joints of the whole body are visible, making the performance of the pose estimator

more reliable. The exercise area only covers one scenario, which means that the acquisition conditions are similar, such as lighting, background, and clutter. It is noteworthy that these factors are not significant for the results as long as they do not affect skeleton detection.

Since the selected dataset only presents one camera perspective, it is necessary to ensure that the subject is under the same perspective so that the system works properly, as shown in Figure 3.2.

Figure 3.2: An illustrative example of an image from the dataset for the expected point of view of the subject who will use the system [16].

All videos have a label in the temporal and categorical domains by an expert physiologist, which means that it is possible to have a ground truth in the segmentation of repetitions. In addition, it also presents ground truth to evaluate user performance as it presents a set of categorical errors associated with a frame range.

Table 3.1 shows the distribution of participants for each exercise, with the squat exercise having the highest number of participants.

Exercise	Participants Number
Squats	45
Jumping Jacks	42
Burpees	37
Push-Ups	32
Sit-Ups	26

Table 3.1: Subjects number per exercise.

3.4 Human Pose Estimation - MediaPipe BlazePose

The developed models for the segmentation and evaluation of the exercises present skeleton keypoints as input instead of RGB video frames. Therefore, for the human pose estimation from CrossFit videos, BlazePose was implemented through MediaPipe Pose framework since it is fast, light, accurate, simple to implement and achieves performance in real-time. Given a video or a sequence of images of a selected exercise, for each video frame the BlazePose system generates 3D coordinates of 33 joints of the human body, as shown before in Figure 2.2.

HPE using MediaPipe BlazePose includes a two-step detector-tracker. With the detector, it initially locates the subject's Region Of Interest (ROI) in the frame, while the tracker predicts the subject's pose landmarks. Since the input data are video sequences, the detector is invoked only on the first frame or when the tracker can no longer identify the pose in the frame. This process is fast and efficient since the detection is computationally expensive but compensated by the speed of tracking. Therefore, this model is specifically designed for real-time pose estimation [2].

3.5 Exercise Repetition Segmentation and Counting

3.5.1 KNN algorithm

KNN is a simple, non-parametric supervised learning algorithm that is primarily used for classification and regression. This algorithm assumes that similar samples are close to each other and therefore belong to the same class [15] [44].

To implement a classifier according to this algorithm we need a sample dataset with

respective labels and K value. The classification process starts by computing the distance between the current data to be tested and the training sample data. Then it is necessary to select the K nearest samples in the training dataset. To achieve this, the distance between the data input and the sample data is computed. The K samples that present the smallest distance value are the selected neighbors. In this way, to determine the class of the current data to be tested, it is verified which is the majority class in the K nearest samples [15] [44]. This entire classifier procedure is outlined in Figure 3.3.

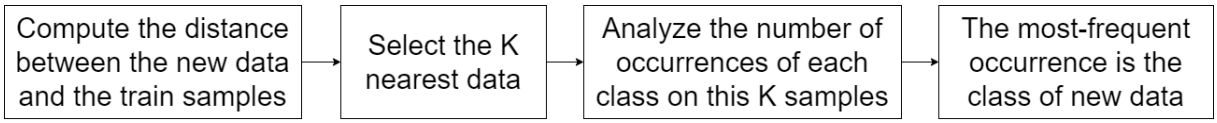


Figure 3.3: Workflow of KNN algorithm. Adapted from [15].

There are several techniques for calculating the distance between samples, however the most commonly used is the Euclidean distance, which represents the shortest distance between two points, according to Equation (3.1).

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.1}$$

3.6 Performance Evaluation

3.6.1 RNN, LSTM and BiLSTM

RNN

Recurrent Neural Networks (RNN) is a neural network that allows information to remain throughout the network. This happens because this network presents recurrent connections that create a memory or state to the network so that it learns and takes advantage of the ordering of observations in the input sequences. Thus, RNN are applied to lists and sequential data, including time series [10].

The basic principle in RNN is that the input data and some information from the previous step are used to calculate the output and to select the information for the next step. This means that RNN allows us to relate recent information from previous steps to the current task. However, the range of contextual information that standard RNN can access is quite limited, and sometimes more context is needed where the relevant information is at a more

distant point. In fact, RNN has a problem in updating weights, which influences a given input in the hidden layer and, therefore, in the output of the network, which can lead to decays or blows up exponentially (vanishing and exploding gradients). Therefore, LSTMs are used to combat this drawback, as their architecture is specifically designed to avoid long-term dependencies [10] [20].

LSTM

Long Short-Term Memory (LSTM) is a highly used RNN with better performance than the standard version and has many applications, such as speech recognition, language modeling and translation [10].

An LSTM layer consists of a set of recurrently connected blocks, called the memory cells that include weights and gates [10] [20] [52].

A generic structure of a LSTM is presented in Figure 3.4 [52].

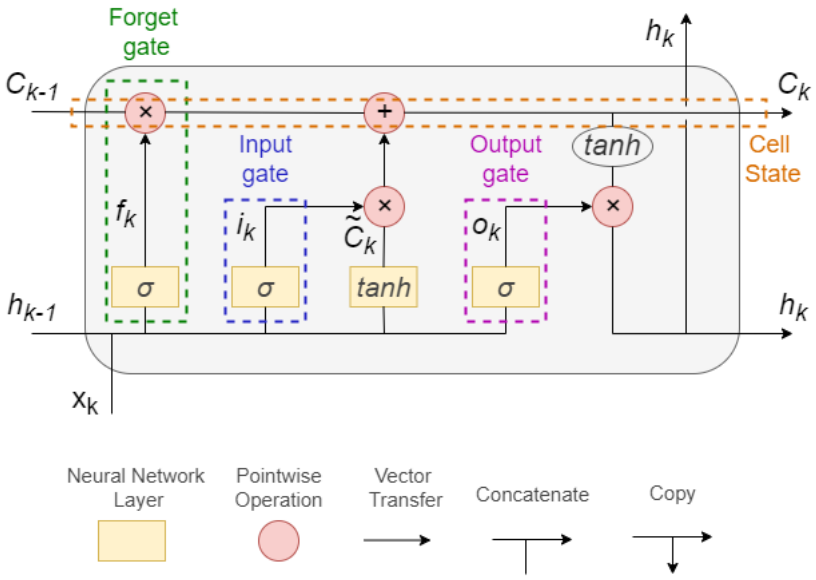


Figure 3.4: Generic architecture of the LSTM memory cell. Adapted from [52].

As can be seen from Figure 3.4, the cell state is an essential component of LSTM because it runs the entire chain, can have information removed or added, carefully regulated by gates. Gates consist of weighted functions that further control the information flow in the memory cell. An LSTM has three gates [10] [52] [53]:

- Forget Gate: decides how much information the current memory cell will receive and discard from the cell state;

- Update Input Gate: decides how much information from the input to update in the memory cell;
- Output Gate: decides what information to output based on the memory cell.

According to Figure 3.4 and considering $X = (x_1, x_2, x_3, ..., x_n)$ the input sequence of length n, the behavior of all gates is described by Equations (3.2) to (3.7).

$$f_k = \sigma(W_{f_x}[x_k], W_{f_h}[h_{k-1}], b_f) \tag{3.2}$$

$$i_k = \sigma(W_{i_x}[x_k], W_{i_h}[h_{k-1}], b_i) \tag{3.3}$$

$$\widetilde{C}_k = tanh(W_{c_x}[x_k], W_{c_h}[h_{k-1}], b_c) \tag{3.4}$$

$$C_k = f_k * C_{k-1} + i_k * \widetilde{C}_k \tag{3.5}$$

$$o_k = \sigma(W_{o_x}[x_k], W_{o_h}[h_{k-1}], b_o) \tag{3.6}$$

$$h_k = tanh(C_k) * o_k \tag{3.7}$$

where k is the current iteration, h_{k-1} is the previous hidden state and h_k is the current hidden state. The indices i, f, o and c are related to the input, forget and output gates and the memory, respectively, W_{f_x}, W_{f_h}, W_{i_x}, W_{i_h}, W_{c_x}, W_{c_h}, W_{o_x} and W_{o_h} are weight matrices, b_f, b_i, b_c and b_o are the bias values and f_k, i_k, o_k are the forget, update input and output gates. $\widetilde{C_k}$ is a vector of the candidate values that will be added into the memory cell and C_k is the current vector of the memory cell [52] [53].

BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM) aims to extract as much useful information as is available from the input sequence by traversing the input time steps in both the forward and backward directions. Therefore, Figure 3.5 shows that BiLSTM adds one more LSTM layer, so that now there are two layers side by side, giving the original input sequence as input to the first layer and reverses the direction of information flow to the second layer [10].

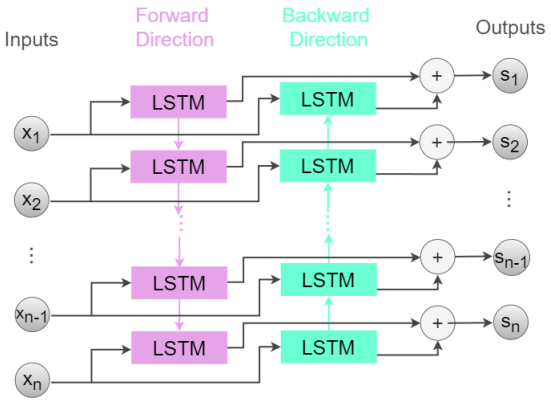


Figure 3.5: BiLSTM. Adapted from [12].

3.6.2 Self-attention Mechanism

As mentioned before, the human pose can be represented by skeleton landmarks. However, human motion sequences include several poses that do not contribute to describing and evaluating human motion [12] [35].

Therefore, the Self-attention mechanism [35] [12] [18] can be used to address this issue because it relates different positions of a sequence to compute a representation of it. Specifically, the proposed self-attention mechanism allows it to extract the most informative aspects of the human motion.

The objective is to assign a score to each pose in a movement sequence to know which poses are more informative [35] [18].

This mechanism takes the LSTM hidden states S as input and outputs a vector of weights A (annotation matrix), according to Equation (3.8).

$$A = -log(W_{s2}tanh(W_{s1}S^T)) \tag{3.8}$$

where $S = s_1, s_2, ..., s_n$ are the hidden states with size n-by-$2u$, u is the hidden unit number for each LSTM, W_{s1} is the weight matrix with shape d-by-$2u$, d is a hyperparameter that determines the weight matrices size, W_{s2} is a vector of parameters with size r-by-d and r is the time steps to pay attention to from the sequence.

After that, we compute the sum up the LSTM hidden states S according to the weight provided by A to get a matrix representation M of the input sequence, with size r-by-$2u$:

$$M = AS \tag{3.9}$$

4 Implementation

This chapter describes all details of the implementation of the developed system, including KNN algorithm for exercise repetition segmentation and counting and Attentive BiLSTM for posture and performance evaluation.

4.1 Exercise Repetition Segmentation and Counting

Each video sequence has a set of repetitions, so we use a KNN algorithm to segment them to be able to assess the subject's posture and performance.

4.1.1 KNN algorithm

The KNN algorithm is used to create a functional pose classifier and recognize the poses to subsequently segment the repetitions. This algorithm is effective as it determines the object class based on the closest samples from the training set. In this method, the input is a set of skeleton landmarks estimated with MediaPipe.

The Key Poses of the exercises are defined according to Table 4.1 and one of them is selected to be the target pose, the pose that defines the limits of the repetition. The outputs of this first step consist of a vector indicating the frames that limit each repetition, a graph that demonstrates the confidence per frame and a video with the skeleton joints and connections and repetitions count.

To count and segment the repetitions, the algorithm monitors the confidence of a target pose class. Confidence corresponds to the number of samples from the target pose, from the K nearest samples.

To implement the KNN classifier, the following steps had to be performed.

1. **Select Key Pose images samples and estimate their pose:**

   Image samples of each exercise were selected corresponding to the poses defined according to Table 4.1. It is important to sample different subjects and exercise variations. Later, the performance of the method using images of only one subject (instructor) is compared with the use of images of several subjects. For each sample, pose detection is performed with MediaPipe BlazePose and saved in a CSV file.

2. **Perform the classification:**

   In the first place, the euclidean distance is used as a distance metric to sort the samples. After that, we find the K pose samples with the lowest distance. The classification of the pose then determines how many neighbors of each key pose are present in these samples. That allows computing the confidence, which corresponds to the number of neighbors of the target pose.

3. **Repetition segmentation and counting:**

   To illustrate the behavior of the algorithm, consider the Squat exercise (Figure 2.3), with the Key poses Up and Down, with Up being the target pose, with two threshold values (HighThreshold and LowThreshold).

   To count a new repetition, the subject must go through both Key Poses (Up and Down). Considering these two thresholds (HighThreshold and LowThreshold), it is determined that the subject must pass through the two key positions to complete an exercise cycle. It is necessary to go below the LowThreshold value to reach the Down pose and then exceed the HighThreshold value to reach the Up pose.The repetition counter increases and the limit of the exercise is saved when the confidence immediately exceeds the HighThreshold.

Exercise	Key Poses
Squats	1: Standing upright position; 2: Squat position: bent knees and hips aligned below them.
Burpees	1: Standing upright position; 2: Squat with hands at the floor; 3: Push-up position: down; 4: Stand and jump with hands above the head
Push-Ups	1: Plank position (full elbow extension); 2: Down position: chest touching the floor with bent elbows.
Sit-Ups	1: Sitting position with feet soles together; 2: Lay down with the hands touching the floor above the head.
Jumping Jacks	1: Standing upright position; 2: Feet apart (shoulder width) and hands touch above the head.

Table 4.1: Key Poses per exercise [16].

Equation (4.1) shows how accuracy was calculated to evaluate the performance of the repeat segmentation algorithm.

$$Acc = \frac{T_P}{T_P + F_P + F_N} \tag{4.1}$$

Where T_P are true positives, which includes the limits predicted that correspond to a real limit, F_P are false positives, which includes the limits predicted that do not correspond to a real limit or a limit repeated and F_N are false negatives, which includes the real limits that were not predicted.

It is worth noting that when one pose is not detected, we simply move on to the next. For a real-world application, a variable would be needed to control the number of consecutive frames in which the pose was not detected so that a warning message would appear for the subject.

4.2 Performance Evaluation with Attentive LSTM

To evaluate the users' posture and performance, it is necessary to pre-process each repetition.

4.2.1 Dataset Pre-processing and Distribution

Repetitions per Exercise

The input of the Attentive BiLSTM, in the case of binary classification, includes the skeleton landmarks of each repetition, with the respective binary target (0 for a well-performed repetition and 1 for an ill-performed repetition). The number of repetitions per exercise is shown in Table 4.2, with the Jumping Jacks exercise having the most samples and the Burpees exercise having the fewest samples. Figure 4.1 shows the percentage of repetitions of each exercise, divided by well and ill-performed.

Exercise	Repetitions number
Jumping Jacks	1180
Squats	897
Push-Ups	516
Sit-Ups	337
Burpees	293

Table 4.2: Repetitions per exercise.

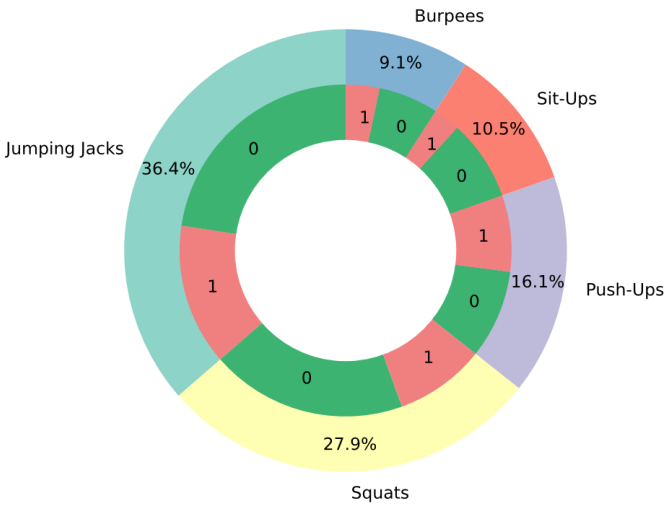


Figure 4.1: Pie chart that shows in the outer circle the percentage of repetitions per exercise and in the inner circle shows the ill-performed (red) and well-performed (green).

For multi-class classification, the most common errors made by participants performing CrossFit exercises are defined in Table 4.3, according to an expert physiologist [16].

Exercise	Errors List
Squats	1. Excessive feet rotation; 2. Incomplete hip flexion; 3. Knee abduction; 4. Incomplete hip extension; 5. Excessively flexed trunk.
Burpees	1. Jumping with uncoordinated feet; 2. Chest not touching the floor; 3. Not jumping at all; 4. Jumping without hip extension; 5. Hands below head.
Push-Ups	1. Chest not touching the floor; 2. No plank position; 3. Resting with chest touching the floor; 4. Touching with knees at the floor; 5. Elbows not fully extended
Sit-Ups	1. Not laying down completely; 2. Hands not touching ground above head; 3. Not touching tiptoe or ground; 4. Closing lower limbs.
Jumping Jacks	1. Movements with different rhythms; 2. Not fully closing the legs or not touching the hands; 3. Asynchronous motion: limbs movement in opposite directions. 4. Knee abduction.

Table 4.3: Proposed exercises and corresponding most common errors [16].

Pre-Processing

The input data were pre-processed to improve the training and efficiency of the deep learning model and to certify that all repetitions have the same size.

The length distribution of each repetition of Jumping Jacks of the original dataset is shown in Figure 4.2. Through this figure is visible the big difference between the longest repetition (110 frames) and the shortest one (11 frames).

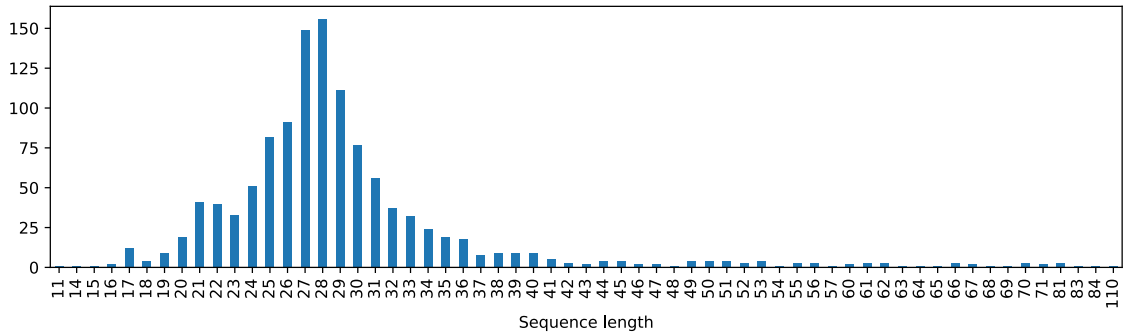


Figure 4.2: Length distribution of each repetition of Jumping Jacks of the original dataset.

Assuming that extremely short or long sequences are low-quality samples, we did an outlier removal using Z-score, with Z = 2.5. This means that the repetitions that are more than two and a half standard deviations from the mean are discarded which results in the distribution shown in Figure 4.3.

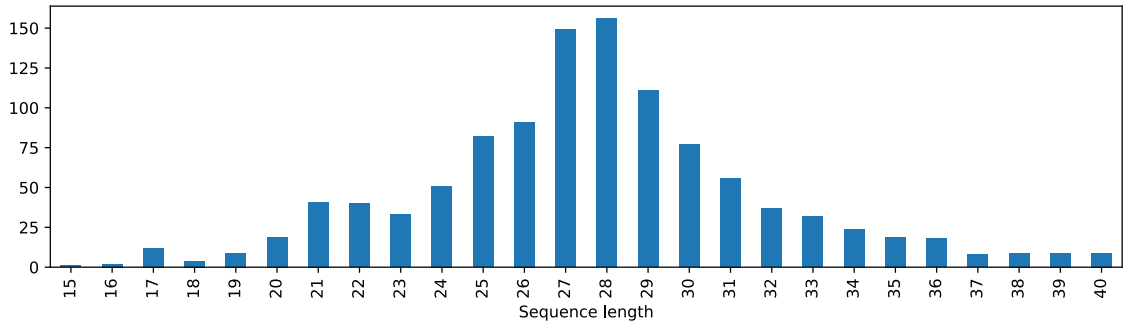


Figure 4.3: Distribution of lengths per repetition of Jumping Jacks after outlier removal with Z-Score.

After that, zero padding was used to certify that each repetition has the same length (length of the longest repetition) and then the values were normalized in range of 0 and 1.

Train, Validation and Test Split

To avoid the model overfitting and to accurately evaluate the model, it is essential to split the data into training, validation, and testing samples. So, the partition of the dataset has been 70%, 15%, 15% for the training, validation and testing, respectively.

4.2.2 BiLSTM with Self-Attentive Mechanism

In the second stage of the pipeline, the pre-processed repetitions were fed into the BiLSTM. The attentive BiLSTM method, developed with the TensorFlow framework, is derived from

the implementation of Coskun et al. [12] and was implemented by another member of the team.

So, given n time steps of a motion sequence X=x_1, x_2, ..., x_n, the output $S = (s_1, s_2, ..., s_n)$ of the BiLSTM is computed. The BiLSTM is followed by the Batch Normalization (BN) and the dropout. The output of this layer is forwarded to the attention layer, which is followed by the structure: Fully Connected Layer (FC)(320,), dropout, BN, FC(320), BN, FC(final_embedding_size), BN, l2 Norm, FC(1), where $FC(m)$ is a fully connected layer with m hidden units. We use leaky ReLU as activation function for every FC.

About the self-attention mechanism, W_{s1} and W_{s2} are the weights with shapes d-by-$2u$ and r-by-d, respectively, where u is the hidden unit number for each LSTM, d is a hyperparameter that determines the weight matrices size and r is the time steps to pay attention to from the sequence. These weights are initialized with the Glorot Uniform distribution.

The loss function used was the cross entropy loss which is the most commonly used for classification problems. The default dropout value (p=0.5) was used to avoid overfitting [6]. BN is added between layers also to prevent overfitting and to speed up the model convergence [23]. The optimizer used was AdaGrad with the default initial learning rate equal to 0.001. The activation function used was the Sigmoid for binary classification and Softmax for multiclass classification.

Best Combination of Hyper-Parameters

In order to determining the best combination of hyperparameters that maximizes the model performance, the following hyperparameters were used and tested:

For *hidden_units* we tested with 32, 64 and 128, for r we tested with 3, 5 and 10 (can't be much higher than this, because some repetitions length are really small), d was tested for 5, 10 and 15, $final\_embedding\_size$ was tested with 16, 32, 64, 128, *epochs* was tested with 300, 500, 700 and 1000, *batch_size* for 32 and 64, and *optimizer* for SGD (LR = 0.01), Adam (LR = 0.001), AdaGrad (LR = 0.001), AdaGrad (LR = 0.002) and RMSProp (LR = 0.001), where LR is learning rate.

Evaluation metrics:

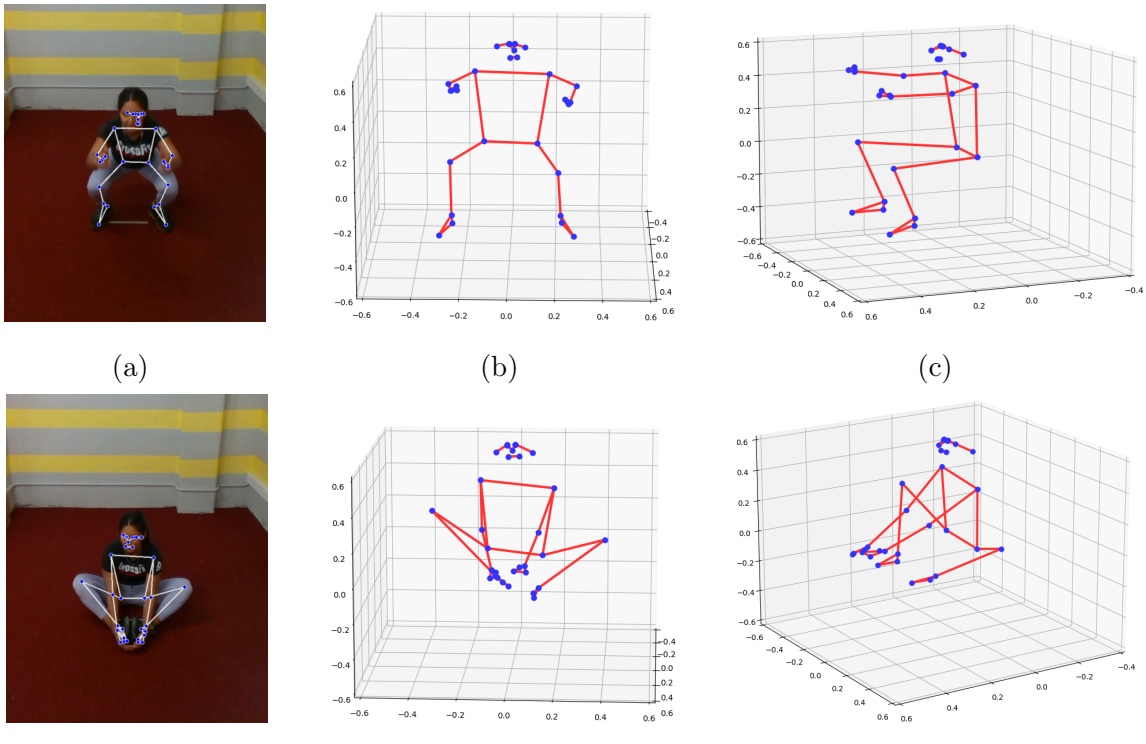
Accuracy, cross entropy loss and F1-score were the evaluation metrics used. The most relevant for this case is the F1-score because the main goal is the recovery and improvement of patients' health and false positives make this task harder.

5 Results and Discussion

To evaluate the proposed solution, a set of experiments were performed. The present chapter is dedicated to the evaluation of these experiments and the discussion of the results.

5.1 Human Pose Estimation

MediaPipe with BlazePose was used to perform the human pose estimation, obtaining 3D landmarks, as we can see in Figure 5.1. This estimation was crucial to acquire the best representation of the exercises in order to segment and evaluate them properly.



(a) (b) (c)

(d) (e) (f)

Figure 5.1: Landmarks estimated with MediaPipe for Squats and Sit-Ups. (a) anb (d) are 3D landmarks normalized by frame width and height. (b), (c), (e) and (f) are 3D landmarks with pose center as point between hips.

From Figure 5.1 we can see that this method is effective, but like other estimators of human posture, it has a drawback in that it sometimes does not predict all landmarks because some parts are occluded, which means that we "lose" some information. Through the mean visibility of landmarks per exercise sequence, these occlusions were verified mostly on Burpees and Push-Ups as we can see from Table 5.1.

Exercise	Visibility (%)
Jumping Jacks	98.69
Squats	96.90
Sit-Ups	91.32
Burpees	82.51
Push-Ups	80.92

Table 5.1: Visibility per exercise performed by subject 2.

It is worth noting that by using skeletal data instead of image sequences, we had a great benefit in terms of data compression, obtaining a skeletal dataset of 242 MB instead of 31.5 GB RGB corresponding to the original image sequences.

5.2 Exercise Repetition Segmentation and Counting

For segmentation, the KNN algorithm was used, testing different values of K, HighThreshold and LowThreshold, being aware that the K value cannot exceed the number of samples of each key pose for each exercise.

This method was tested initially with samples of key poses of only one subject in order to find out if it is a feasible method with data from only one certified trainee and then with a set of images of more subjects. To evaluate the results, we save a video showing the repetitions number and the skeleton joints and connections, plot the confidence per frame and calculate the accuracy according to Equation (4.1). A repetition is considered correct if the value predicted is in a given time window of 15 past frames.

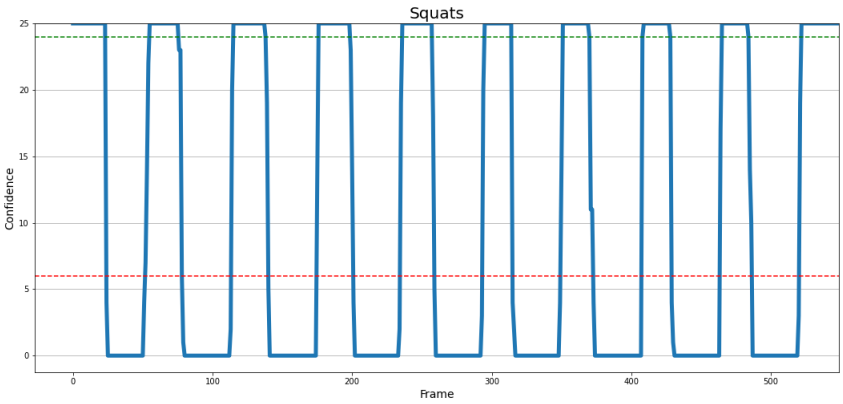


Figure 5.2: Plot of the confidence per frame of subject 45 performing Squats. Blue line - Confidence; Green line - High Threshold (= 24); Red Line - Low Threshold (= 6).

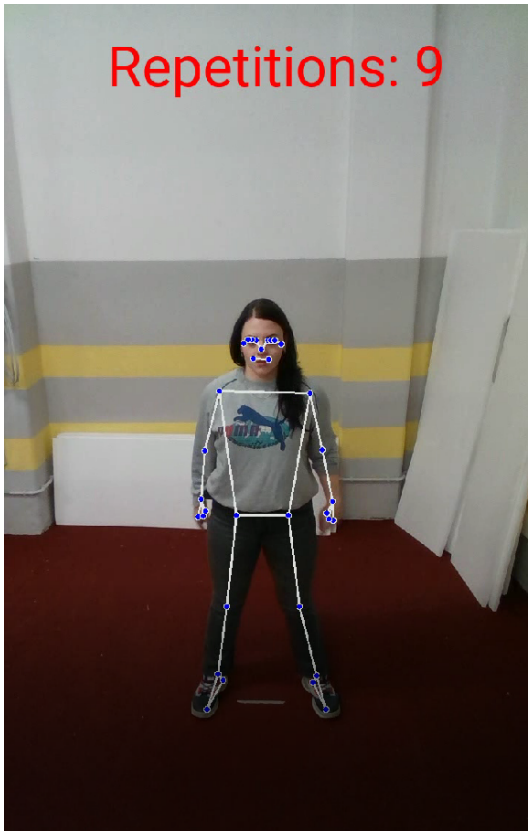


Figure 5.3: Frame of the output video showing the skeleton joints and connections and the repetitions number of subject 45 performing Squats.

For the Sit-Ups the KNN classifier was tested with a set of 5 subjects individually (because it has the least participants number), whereas for the other exercises, a set of 10 subjects was used. The results for the Squats can be found in Table 5.2, where it can be seen that using samples from different subjects, the results can vary between them by about 5 percentage points (pp).

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Accuracy	74.02	70.90	74.54	73.84	75.43	75.55	73.59	72.59	72.76	73.00

Table 5.2: Segmentation accuracy for Squats using samples of 10 subjects individually, with K = 20 and Thresholds of 19 and 13.

As previously mentioned, for the five exercises, the use of samples from different subjects was tested individually and the best results are recorded in Tables 5.3 to 5.7. After an empirical study, it was found that the K and Threshold values selected, among the others that were tested, proved to have the best results.

- For Squats, the best values were obtained using samples from subject 6, K = 25 and Thresholds 24 and 10, with a mean accuracy of 84.99%.
- For Burpees, the best values were obtained using samples from subject 2, K = 25 and Thresholds 24 and 6, with a mean accuracy of 82.61%. (key poses 1 and 3)
- For Push-Ups, the best values were obtained using samples from subject 5, K = 25 and Thresholds 24 and 6, with a mean accuracy of 80.61%.
- For Sit-Ups, the best values were obtained using samples from subject 1, K = 25 and Thresholds 24 and 15, with a mean accuracy of 86.74%.
- For Jumping Jacks, the best values were obtained using samples from subject 4, K = 25 and Thresholds 24 and 6, with a mean accuracy of 93.97%.

Burpees and Push-Ups are the exercises with more occlusions and have the lower mean accuracy, while Jumping Jacks have the highest mean accuracy and higher landmarks visibility.

For the five exercises, we also tested the use of samples from different subjects and the best results are recorded in Tables 5.8 to 5.12, using K = 25 and Thresholds 24 and 6.

- For Squats, the best values were obtained using samples from subjects 3, 4 and 6 with a mean accuracy of 81.89%.
- For Burpees, the best values were obtained using samples from subjects 2, 12 and 14 with a mean accuracy of 84.19% (key poses 1 and 3).

- For Push-Ups, the best values were obtained using samples from subjects 5, 6 and 10 with a mean accuracy of 79.29%.
- For Sit-Ups, the best values were obtained using samples from subjects 1 and 5 with a mean accuracy of 87.34%.
- For Jumping Jacks, the best values were obtained using samples from subjects 2, 4 and 5 with a mean accuracy of 93.67%.

Tables 5.3 through 5.12 highlight critical results that will be discussed individually. It is possible to denote that for both cases where samples from only one subject and different subjects are used, the lower results are from the same subjects.

The algorithm had difficulty predicting the repetitions for subject 17 for Push-Ups and subject 27 for Sit-Ups due to the defined LowThreshold and HighThreshold, respectively. This can be seen in Figures 5.4 and 5.5, where the trouble areas are circled in orange.

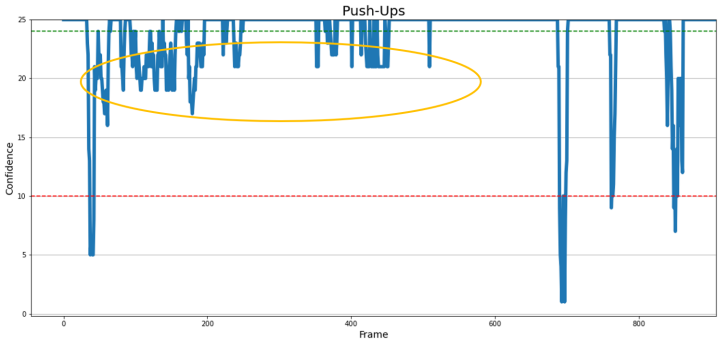


Figure 5.4: Plot of the confidence per frame of subject 17 performing Push-Ups. Blue line - Confidence; Green line - HighThreshold; Red Line - LowThreshold

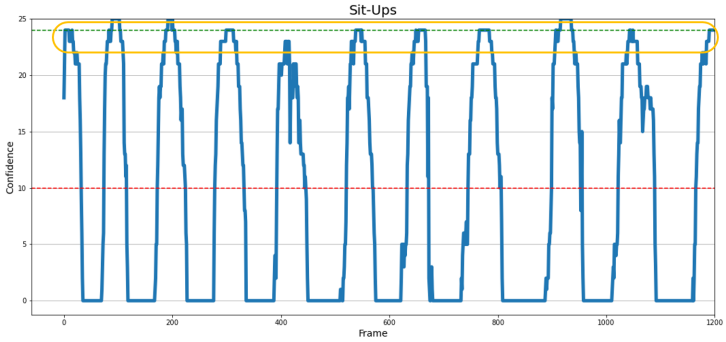


Figure 5.5: Plot of the confidence per frame of subject 27 performing Sit-Ups. Blue line - Confidence; Green line - HighThreshold; Red Line - LowThreshold

For the remaining cases, i.e., subject 27 for Squats, subjects 11 and 18 for Burpees and

subject 28 for Push-Ups, it was determined from the predicted limits and the output video that the counts are performed too early, i.e., the upper limit is reached earlier than expected.

It is also verified that the use of samples from different subjects did not significantly improve the results, so it is preferable to use samples from only one subject to perform the segmentation.

The use of more key poses was investigated to find out if it would influence the results. After comparing the use of two and four key poses for Burpees, with K = 25 and Thresholds 24 and 10, we obtained 79.46% and 74.48%, respectively, concluding that the addition of intermediate key poses is not justified.

Since the examples used for training and evaluation of the KNN algorithm were all from the same perspective, we performed an additional experiment using an ordinary RGB camera with a perspective variation of about 45^{o} to test the effectiveness of the algorithm in counting. It was found that the algorithm correctly counts Jumping Jacks in this perspective.

	S1	S2	S3	S4	S5	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16
S6	100.0	100.0	100.0	100.0	52.63	100.0	100.0	100.0	75.00	88.89	100.0	100.0	72.22	69.23	92.86

	S17	S18	S19	S20	S21	S22	S23	S24	S25	S26	S27	S29	S30	S31	S32
S6	72.22	48.00	80.95	81.25	100.0	84.62	100.0	66.67	100.0	62.96	21.74	96.15	100.0	50.00	93.33

	S33	S34	S35	S36	S37	S38	S39	S40	S41	S44	S45	S46	S49	S50
S6	90.91	100.0	94.44	90.48	66.67	100.0	75.00	100.0	100.0	75.00	100.0	83.33	61.11	93.75

Table 5.3: Segmentation accuracy for Squats using samples of subject 6 and thresholds of 24 and 10, K = 25.

	S1	S3	S4	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21
S2	94.44	94.44	88.89	87.75	14.29	71.43	40.00	40.00	100.0	66.67	100.0	28.57	100.0	89.47	66.67

	S22	S24	S25	S27	S28	S29	S31	S32	S33	S34	S35	S36	S37	S38	S39
S2	54.55	92.31	69.23	100.0	100.0	100.0	88.89	100.0	100.0	100.0	100.0	100.0	100.0	100.0	80.00

	S40	S41	S44	S45	S48	S49
S2	100.0	80.00	100.0	66.67	100.0	60.00

Table 5.4: Segmentation accuracy for Burpees using samples of subject 2 and thresholds of 24 and 6, K = 25.

	S2	S3	S6	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21
S5	42.31	96.55	95.65	100.0	84.62	100.0	75.00	100.0	100.0	50.00	11.11	45.45	100.0	86.62	91.30

	S22	S24	S25	S27	S28	S29	S31	S32	S33	S34	S35	S36	S38	S40	S41
S5	68.75	93.75	93.10	50.00	7.69	100.0	64.29	100.0	100.0	50.00	100.0	100.0	100.0	100.0	94.74

	S48
S5	100.0

Table 5.5: Segmentation accuracy for Push-Ups using samples of subject 5 and thresholds of 24 and 6, K = 25.

	S2	S3	S5	S15	S16	S17	S19	S20	S21	S22	S24	S25	S27	S28	S29
S1	88.46	90.91	100.0	90.00	93.33	80.00	92.31	100.0	84.62	71.43	72.22	93.33	18.18	50.00	100.0

	S31	S32	S33	S34	S35	S36	S38	S40	S41	S48
S1	100.0	100.0	100.0	100.0	100.0	100.0	60.00	93.75	100.0	90.00

Table 5.6: Segmentation accuracy for Sit-Ups using samples of subject 1 and thresholds of 24 and 23, K = 25.

	S1	S2	S3	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16
S4	100.0	100.0	96.49	100.0	100.0	100.0	100.0	100.0	76.19	100.0	100.0	86.96	100.0	100.0	96.55

	S17	S18	S19	S20	S21	S22	S24	S25	S27	S28	S29	S31	S32	S33	S34
S4	100.0	60.98	100.0	100.0	91.18	50.09	90.91	97.78	95.65	100.0	100.0	100.0	100.0	57.14	100.0

	S35	S36	S37	S38	S39	S40	S41	S44	S45	S48	S49
S4	95.83	100.0	78.95	100.0	92.86	100.0	100.0	94.44	95.00	100.0	86.96

Table 5.7: Segmentation accuracy for Jumping Jacks using samples of subject 4 and thresholds of 24 and 6, K = 25.

	S1	S2	S5	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18
S3, S4, S6	100.0	100.0	52.63	100.0	100.0	100.0	75.00	88.89	100.0	80.00	72.22	64.29	92.86	65.00	42.86

	S19	S20	S21	S22	S23	S24	S25	S26	S27	S29	S30	S31	S32	S33	S34
S3, S4, S6	80.95	78.79	100.0	78.57	100.0	64.71	100.0	62.96	22.27	92.59	100.0	75.00	82.35	90.91	47.06

	S35	S36	S37	S38	S39	S40	S41	S44	S45	S46	S49	S50
S3, S4, S6	94.44	100.0	50.00	91.67	75.00	100.0	100.0	75.00	100.0	100.0	45.00	93.75

Table 5.8: Segmentation accuracy for Squats using samples of subjects 3, 4 and 6 and thresholds of 24 and 6, K = 25.

	S1	S3	S4	S10	S11	S13	S15	S16	S17	S18	S19	S20	S21	S22	S24
S2, S12, S14	94.44	94.44	88.89	87.50	14.29	40.00	100.0	66.67	100.0	28.57	100.0	89.47	66.67	54.55	92.31

	S25	S27	S28	S29	S31	S32	S33	S34	S35	S36	S37	S38	S39	S40	S41
S2, S12, S14	69.23	100.0	100.0	100.0	88.89	100.0	100.0	100.0	100.0	100.0	100.0	100.0	80.00	100.0	80.00

	S44	S45	S48	S49
S2, S12, S14	100.0	66.67	100.0	60.00

Table 5.9: Segmentation accuracy for Burpees using samples of subjects 2, 12 and 14 and thresholds of 24 and 6, K = 25.

	S2	S3	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S24
S5, S6, S10	42.31	96.55	84.62	100.0	75.00	100.0	100.0	50.00	11.11	41.67	100.0	84.62	91.30	68.75	93.75

	S25	S27	S28	S29	S31	S32	S33	S34	S35	S36	S38	S40	S41	S48
S5, S6, S10	93.10	50.00	7.69	100.0	64.29	100.0	100.0	50.00	100.0	100.0	100.0	100.0	94.74	100.0

Table 5.10: Segmentation accuracy for Push-Ups using samples of subjects 5, 6 and 10 and thresholds of 24 and 6, K = 25.

	S2	S3	S15	S16	S17	S19	S20	S21	S22	S24	S25	S27	S28	S29	S31
S1, S5	92.00	90.91	90.00	93.33	80.00	92.31	100.0	84.62	71.43	81.25	93.33	18.18	50.00	100.0	100.0

	S32	S33	S34	S35	S36	S38	S40	S41	S48
S1, S5	100.0	100.0	100.0	100.0	100.0	75.00	93.75	100.0	90.00

Table 5.11: Segmentation accuracy for Sit-Ups using samples of subjects 1 and 5 and thresholds of 24 and 6, K = 25.

	S1	S3	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18
S2, S4, S5	100.0	96.49	100.0	100.0	100.0	100.0	76.19	100.0	100.0	86.96	100.0	100.0	96.55	100.0	60.98

	S19	S20	S21	S22	S24	S25	S27	S28	S29	S31	S32	S33	S34	S35	S36
S2, S4, S5	100.0	100.0	91.18	59.09	90.91	97.78	95.65	100.0	100.0	100.0	100.0	57.14	100.0	95.83	100.0

	S37	S38	S39	S40	S41	S44	S45	S48	S49
S2, S4, S5	78.95	100.0	92.86	100.0	100.0	94.44	95.00	100.0	86.96

Table 5.12: Segmentation accuracy for Jumping Jacks using samples of subjects 2, 4 and 5 and thresholds of 24 and 6, K = 25.

5.3 Performance Evaluation

The performance evaluation of the quality of the exercises performed by the user was tested as a binary and multiclass classification problem with skeletal joints obtained with MediaPipe using repetition segmentation with Ground Truth and with KNN. For all models, we use learning curves to monitor learning performance. For the following results, the best hyperparameters are selected to maximize the model's performance.

Binary Classification - Segmentation with ground truth

Considering both training time and the F1-Score, the model with best results was implemented with $epochs = 500$, $hidden\_units = 128$, $r = 5$, $d = 10$, $final\_embedding\_size = 64$, $batch\_size = 32$ and $optimizer = AdaGrad$ and $initial\_learning\_rate = 0.001$. In the test, we obtained a cross entropy of 0.1425, binary accuracy of 0.9521 and F1-Score of 0.9418. The training and validation learning curves are represented in Figure 5.6. The training took about 48 minutes.

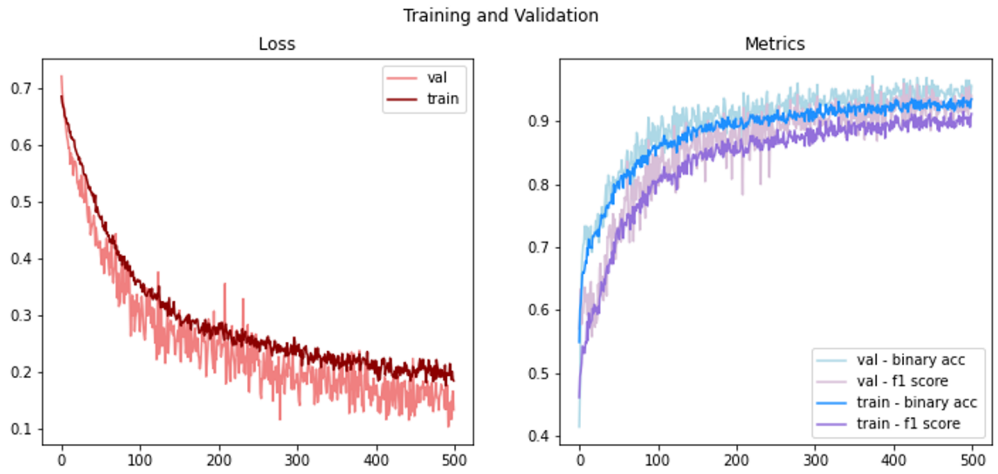


Figure 5.6: Training and Validation curve of Loss, Accuracy and F1-score for binary classification with segmentation with ground truth.

Using the same parameters, but without the self-attention mechanism, we reached a cross entropy of 0.1897, binary accuracy of 0.9177 and F1-Score of 0.8933. The training took about 1h15 and the training and validation learning curves are represented in Figure 5.7. Thus, the attention mechanism improves the results slightly and helps to reduce the training time.

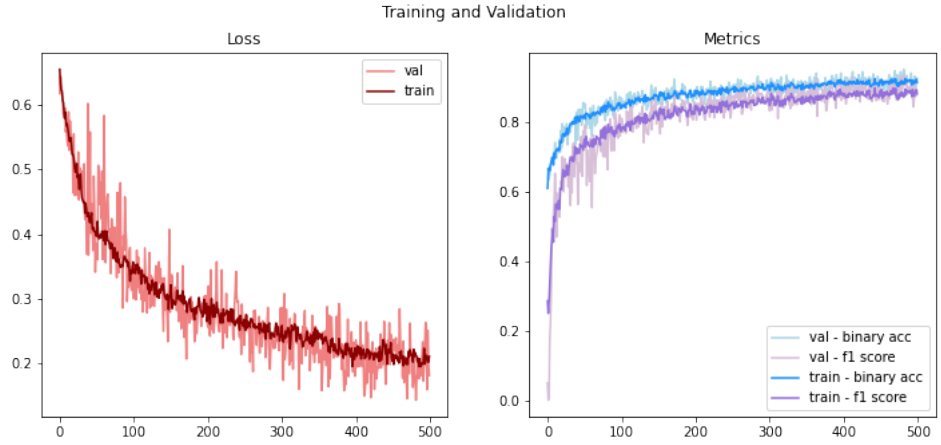


Figure 5.7: Training and Validation curve of Loss, Accuracy and F1-score for binary classification with segmentation with ground truth. without attention mechanism.

The previous results show that adding the self-attention mechanism reduces the training time and improves the performance of the model.

Binary Classification - Segmentation with KNN

For binary classification, with repetitions segmented with KNN, we use $epochs = 700$, $hidden\_units = 64$, $r = 5$, $d = 10$, $final\_embedding\_size = 64$, $batch\_size = 32$ and $optimizer = AdaGrad$ and $initial\_learning\_rate = 0.001$. In the test, we obtained a cross entropy of 0.1795, binary accuracy of 0.9347 and F1-Score of 0.9043. The training and validation learning curves are represented in Figure 5.8. The training took about 1h10.

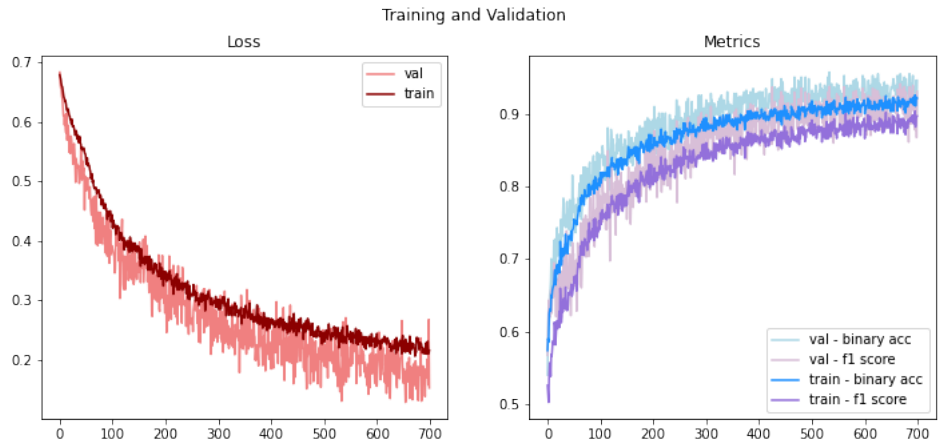


Figure 5.8: Training and Validation curve of Loss, Accuracy and F1-score for binary classification with Segmentation with KNN.

As expected, the results with repetitions segmented with KNN are slightly lower than

those from segmentation using ground truth.

Binary Classification - Training with Segmentation with ground truth and Testing with Segmentation with KNN

In order to simulate a real-world behavior, we train the model with segmentation with ground truth and test with segmentation with KNN. We use $epochs = 600$, $hidden\_units = 128$, $r = 5$, $d = 10$, $final\_embedding\_size = 64$, $batch\_size = 32$ and $optimizer = AdaGrad$ and $initial\_learning\_rate = 0.001$. In the test, we achieved a cross-entropy loss of 0.8907, accuracy of 0.7309 and F1-Score of 0.6629. The training and validation learning curves are represented in Figure 5.10.

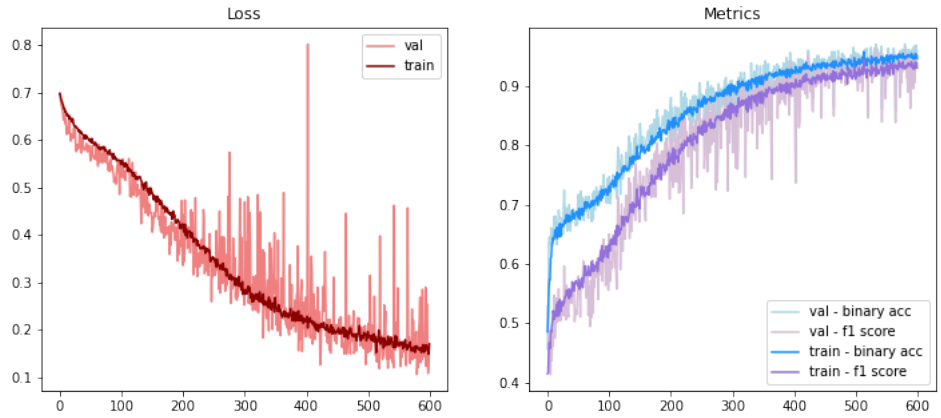


Figure 5.9: Training curve of Loss, Accuracy and F1-score for binary classification, trained with ground truth segmentation and tested with KNN segmentation.

Comparing the F1-Score obtained (0.6629) with the best F1-Score (0.9418) we can see a variation of 25 percentage points.

Multi-class Classification

For multi-class classification, with repetitions segmented with ground truth, we use $epochs = 1000$, $hidden\_units = 64$, $r = 5$, $d = 10$, $final\_embedding\_size = 64$, $batch\_size = 32$ and $optimizer = AdaGrad$ and $initial\_learning\_rate = 0.001$. The training took about 31 minutes. In the test, we achieved a cross-entropy loss of 0.3293 and accuracy of 0.8951. The training and validation learning curves are represented in Figure 5.10.

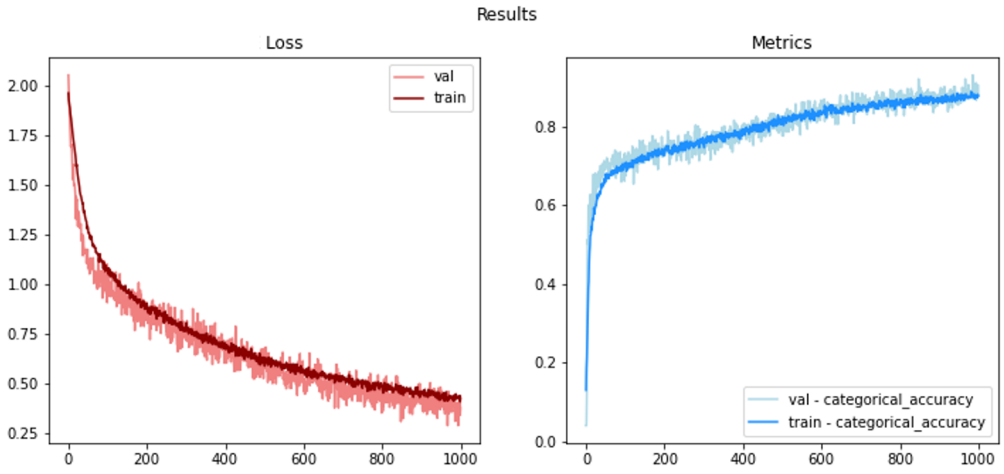


Figure 5.10: Training and Validation curves of Loss and Accuracy for multi-class classification.

The results with multi-class classification are slightly lower than in binary classification because in this case, it's more notorious the unbalanced data. This model is enabled to predict only one error per exercise.

Overall, the plots obtained (Figure 5.6 to 5.10) show models well-fitted because the training and validation optimization learning curves (loss curves) decrease and performance learning curves (accuracy and F1-Score curves) increase to a point of stability.

6 Conclusions and Future Perspectives

In this chapter, some conclusions are drawn about the work done, the merits of the proposed methods are pointed out, and future research directions are suggested.

6.1 Work Accomplished and Conclusions

The aim of this dissertation was to develop a system capable of evaluating and providing feedback to the user when performing certain exercises, in order to motivate the user and achieve a fast and effective recovery. This system can bring benefits at environmental and economic sustainability levels if applied to rehabilitation patients, reducing the trips to rehabilitation clinics and health costs.

The proposed system estimates the pose of the user with MediaPipe, segment and count the repetitions with KNN, evaluate each repetition with Attentive BiLSTM allowing to understand if the user is performing the exercises properly.

Using MediaPipe to estimate human pose allows for a more practical, affordable, and simple system, as it only requires the use of an ordinary RGB camera, rather than more complex and intrusive setups, such as using markers over some key joints on the user's body.

About the KNN algorithm, it has advantages since it evaluates each frame immediately and it is easy to implement. The addition of new data is a simple task and the only parameters necessary to monitor are K and Thresholds.

For the evaluation of each repetition, we use an Attentive BiLSTM algorithm capable of binary and multi-class classification. This algorithm, trained within a short time period, was efficient in performance evaluation.

6.2 Future Work

After the conclusion of this dissertation, some future work is identified, highlighting some improvements:

1. Create a wider dataset with more perspectives and exercise variations and apply other kinds of exercises to the system (specifically rehabilitation exercises), including both repetition exercises and isometric exercises, and according to each exercise count or measure the time of each repetition;

2. The approach developed assumes that the exercise performed by the user is the one indicated by the system. However, it would also be safer to create a detection process to confirm that the user is performing the indicated exercise prior to segmentation and evaluation;

3. Since the attention mechanism of the BiLSTM can extract key poses for performance assessment, it can potentially be used for repetition segmentation.

7 Bibliography

[1] COCO - Common Objects in Context. `https://cocodataset.org/`. Accessed: 14-03-2022.

[2] MediaPipe Pose. `https://google.github.io/mediapipe/solutions/pose/`. Accessed: 14-03-2022.

[3] ML Kit - Pose Classification Options. `https://developers.google.com/ml-kit/vision/pose-detection/classifying-poses`. Accessed: 24-03-2022.

[4] ILHAN AYTUTULDU. Performance assessment of physiotherapy and rehabilitation exercises with deep learning. 2019.

[5] Konstantinos Bacharidis and Antonis Argyros. Exploiting the Nature of Repetitive Actions for Their Effective and Efficient Recognition. *Frontiers in Computer Science*, 4(March), 2022.

[6] Pierre Baldi and Peter J Sadowski. Understanding dropout. *Advances in neural information processing systems*, 26, 2013.

[7] Valentin; Grishchenko Ivan Bazarevsky. On-device, Real-time Body Pose Tracking with MediaPipe BlazePose. 2020.

[8] Daniel Delgado Bellamy and Praminda Caleb-Solly. *Collaborative HRI and Machine Learning for Constructing Personalised Physical Exercise Databases*, volume 11649 LNAI. Springer International Publishing, 2019.

[9] Merve Bozo, Erchan Aptoula, and Zehra Cataltepe. A Discriminative Long Short Term Memory Network with Metric Learning Applied to Multispectral Time Series Classification. *Journal of Imaging*, 6(7):7–9, 2020.

[10] Jason Brownlee. Long Short-Term Memory Networks With Python. *Machine Learning Mastery With Python*, 1(1):228, 2017.

[11] Yu Chen, Chunhua Shen, Xiu Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial PoseNet: A Structure-Aware Convolutional Network for Human Pose Estimation. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:1221–1230, 2017.

[12] Huseyin Coskun, David Joseph Tan, Sailesh Conjeti, Nassir Navab, and Federico Tombari. Human motion analysis with deep metric learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11218 LNCS:693–710, 2018.

[13] Ross Cutler and Larry S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, 2000.

[14] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 10384–10393, 2020.

[15] Zhu Fan, Jia Kun Xie, Zhong Yu Wang, Pei Chen Liu, Shu Jun Qu, and Lei Huo. Image Classification Method Based on Improved KNN Algorithm. *Journal of Physics: Conference Series*, 1930(1), 2021.

[16] Bruno Ferreira, Pedro M. Ferreira, Gil Pinheiro, Nelson Figueiredo, Filipe Carvalho, Paulo Menezes, and Jorge Batista. Deep learning approaches for workout repetition counting and validation. *Pattern Recognition Letters*, 151:259–266, 2021.

[17] Mihai Fieraru, Mihai Zanfir, Silviu Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. AIFit: Automatic 3D Human-Interpretable Feedback Models for Fitness Training. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 9914–9923, 2021.

[18] F. A. Furfari(tony). Attention Is All You Need. *IEEE Industry Applications Magazine*, 8(1):8–15, 2002.

[19] John Cristian Borges Gamboa. Deep Learning for Time-Series Analysis. *CoRR*, abs/1701.0, 2017.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[21] Rıza Alp Guler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense Human Pose Estimation In TheWild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2016.

[22] Shiyao Huang, Xianghua Ying, Jiangpeng Rong, Zeyu Shang, Hongbin Zha, and Computer Science. Camera Calibration from Periodic Motion of a Pedestrian Shiyao Huang , Xianghua Ying *, Jiangpeng Rong , Zeyu Shang and Hongbin Zha Key Laboratory of Machine Perception (Ministry of Education) School of Electronic Engineering and Computer Science , Cent. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3025–3033, 2016.

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[24] Hyun Kang, Chang Woo Lee, and Keechul Jung. Recognition-based gesture spotting in video games. *Pattern Recognition Letters*, 25(15):1701–1714, 2004.

[25] Giorgos Karvounas, Iason Oikonomidis, and Antonis Argyros. ReActNet: Temporal Localization of Repetitive Activities in Real-World Videos. 2019.

[26] Woojoo Kim, Jaeho Sung, Daniel Saakes, Chunxi Huang, and Shuping Xiong. Ergonomic postural assessment using a new open-source human pose estimation technology (OpenPose). *International Journal of Industrial Ergonomics*, 84(February), 2021.

[27] Daniel Leightley, Jamie S. McPhee, and Moi Hoon Yap. Automated Analysis and Quantification of Human Mobility Using a Depth Sensor. *IEEE Journal of Biomedical and Health Informatics*, 21(4):939–948, 2017.

[28] Ofir Levy and Lior Wolf. Live repetition counting. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:3020–3028, 2015.

[29] Jianwei Li, Hainan Cui, Tianxiao Guo, Qingrui Hu, and Yanfei Shen. SPATIO-TEMPORAL FEATURE ENCODING School of Sports Engineering , Beijing Sports University , Beijing 100084 , China NLPR , Institute of Automation , Chinese Academy of Sciences , Beijing 100190 , China. 2020.

[30] Xiaoxiao Li, Vivek Singh, Yifan Wu, Klaus Kirchberg, James Duncan, and Ankur Kapoor. Repetitive Motion Estimation Network: Recover cardiac and respiratory signal from thoracic imaging. (Nips):1–4, 2018.

[31] Xiu Li, Hongdong Li, Hanbyul Joo, Yebin Liu, and Yaser Sheikh. Structure from Recurrent Motion: From Rigidity to Recurrency. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3032–3040, 2018.

[32] Youru Li, Zhenfeng Zhu, Deqiang Kong, Hua Han, and Yao Zhao. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowledge-Based Systems*, 181:104785, 2019.

[33] Yalin Liao, Aleksandar Vakanski, and Min Xian. A Deep Learning Framework for Assessing Physical Rehabilitation Exercises. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(2):468–477, 2020.

[34] Feng-Cheng Lin, Huu-Huy Ngo, Chyi-Ren Dow, Ka-Hou Lam, and Hung Le. Student Behavior Recognition System for the Classroom Environment Based on Skeleton Pose Estimation and Person Detection. *Sensors*, 21:5314, 2021.

[35] Zhouhan Lin, Minwei Feng, Cicero Nogueira Dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–15, 2017.

[36] An Lun Liu and Wei Ta Chu. A posture evaluation system for fitness videos based on recurrent neural network. *Proceedings - 2020 International Symposium on Computer, Consumer and Control, IS3C 2020*, pages 185–188, 2020.

[37] P. S. Madanayake, W. A.D.K. Wickramasinghe, H. P. Liyanarachchi, H. M.D.M. Herath, A. Karunasena, and T. D. Perera. Fitness Mate: Intelligent workout assistant using motion detection. *2016 IEEE International Conference on Information and Automation for Sustainability: Interoperable Sustainable Smart Systems for Next Generation, ICIAfS 2016*, 2016.

[38] Mahmut Kaya and Hasan Sakir Bilge. Deep Metric Learning : A Survey. *Symmetry*, 11.9:1066, 2019.

[39] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose Estimation for Augmented Reality: A Hands-On Survey. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2633–2651, 2016.

[40] Shinnosuke Matsuo, Xiaomeng Wu, Gantugs Atarsaikhan, Akisato Kimura, Kunio Kashino, Brian Kenji Iwana, and Seiichi Uchida. Attention to Warp: Deep Metric Learning for Multivariate Time Series. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12823 LNCS:350–365, 2021.

[41] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3 SPEC. ISS.):90–126, 2006.

[42] Amit Moryossef, Ioannis Tsochantaridis, Roee Aharoni, Sarah Ebling, and Srini Narayanan. Real-Time Sign Language Detection Using Human Pose Estimation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12536 LNCS:237–248, 2020.

[43] Jonghyuk Park, Sukhyun Cho, Dongwoo Kim, Oleksandr Bailo, Heewoong Park, Sanghoon Hong, and Jonghun Park. A Body Part Embedding Model with Datasets for Measuring 2D Human Motion Similarity. *IEEE Access*, 9:36547–36558, 2021.

[44] Martin Persson. Automatic gait recognition – Using Deep Learning (Automatisk Gångstilsigenkänning). 2020.

[45] Patrick Philipp, Nicole Merkle, Kai Gand, and Carola Gißke. Continuous support for rehabilitation using machine learning. *IT - Information Technology*, 61(5-6):273–284, 2019.

[46] Yang Ran, Isaac Weiss, Qinfen Zheng, and Larry S. Davis. Pedestrian detection via periodic motion analysis. *International Journal of Computer Vision*, 71(2):143–160, 2007.

[47] Luis Guilherme Silva Rodrigues, Diego Dias, Marcelo De Paiva Guimaraes, Alexandre Fonseca Brandao, Leonardo Rocha, Rogerio L. Iope, and José Remo Ferreira Brega. Classification of Human Movements with Motion Capture Data in a Motor Rehabilitation Context. *ACM International Conference Proceeding Series*, pages 56–63, 2021.

[48] Tom F.H. Runia, Cees G.M. Snoek, and Arnold W.M. Smeulders. Repetition Estimation. *International Journal of Computer Vision*, 127(9):1361–1383, 2019.

[49] Romeo Šajina and Marina Ivašic Kos. Pose estimation, tracking and comparison.

[50] R Salaheldin, M ElHelw, and N El Gayar. A Time Series Classification Approach for Motion Analysis Using Ensembles in Ubiquitous Healthcare. *InIAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 277–288, 2014.

[51] Mohammadamin Salimi, José J. M. Machado, and João Manuel R. S. Tavares. Using Deep Neural Networks for Human Fall Detection Based on Pose Estimation. *Sensors*, 22(12):4544, 2022.

[52] Rodrigo Salles, Jérôme Mendes, Rui Araújo, Carlos Melo, and Pedro Moura. Prediction of Key Variables in Wastewater Treatment Plants Using Machine Learning Models. In *Proc. 2022 IEEE International Joint Conference on Neural Networks (IJCNN 2022), at the 2022 World Congress on Coomputational Intelligence (WCCI 2022)*, pages 1–9, Padova, Italy, 2022. IEEE.

[53] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. The Performance of LSTM and BiLSTM in Forecasting Time Series. *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, pages 3285–3292, 2019.

[54] Rajiv Singh and Swati Nigam. Deep neural networks for human behavior understanding. *Handbook of Multimedia Information Security: Techniques and Applications*, pages 667–679, 2019.

[55] Liangchen Song, Gang Yu, Junsong Yuan, and Zicheng Liu. Human pose estimation and its application to action recognition: A survey. *Journal of Visual Communication and Image Representation*, 76(February):103055, 2021.

[56] Dapeng Tang. Hybridized Hierarchical Deep Convolutional Neural Network for Sports Rehabilitation Exercises. *IEEE Access*, 8:118969–118977, 2020.

[57] Eduardo Velloso, Andreas Bulling, Hans Gellersen, Wallace Ugulino, and Hugo Fuks. Qualitative activity recognition of weight lifting exercises. *ACM International Conference Proceeding Series*, pages 116–123, 2013.

[58] Qifei Wang, Gregorij Kurillo, Ferda Ofli, and Ruzena Bajcsy. Unsupervised Temporal Segmentation of Repetitive Human Actions Based on Kinematic Modeling and Frequency Analysis. *Proceedings - 2015 International Conference on 3D Vision, 3DV 2015*, (1111965):562–570, 2015.

[59] Bin Wu, Craig Herb, and Mariya Khiterer. Technical Report 20090515 Technical Report 20090515. pages 1–18, 2009.

[60] Qingtian Yu, Haopeng Wang, Fedwa Laamarti, and Abdulmotaleb El Saddik. Deep learning-enabled multitask system for exercise recognition and counting. *Multimodal Technologies and Interaction*, 5(9), 2021.

[61] Huaidong Zhang, Xuemiao Xu, Guoqiang Han, and Shengfeng He. Context-aware and scale-insensitive temporal repetition counting. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:667–675, 2020.

[62] Mohd Asyraf Zulkifley, Nur Ayuni Mohamed, and Nuraisyah Hani Zulkifley. Squat Angle Assessment Through Tracking Body Movements. *IEEE Access*, 7:48635–48644, 2019.