Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 251 (2024) 446–453

Procedia Computer Science

www.elsevier.com/locate/procedia

The 14th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2024)
October 28-30, 2024, Leuven, Belgium

Accuracy Assessment of 2D Pose Estimation with MediaPipe for Physiotherapy Exercises

SIMOES, W.*, REIS, L., ARAUJO, C., MAIA JR., J.

UEA, Av. Darcy Vargas, 1200, Manaus-AM, Postcode: 69050-020, Brazil

Abstract

Currently, it is becoming increasingly important to provide adequate rehabilitation at home and determine strategies to prevent injuries, chronic diseases caused by lack of movement and a sedentary lifestyle. The goal is to help people and improve their quality of life. To enable faster, more practical and cost-effective recovery, monitoring and evaluating the patient's physical rehabilitation at home is crucial to providing feedback to the user. Therefore, this article proposes a system for evaluating the user's posture and performance during physical therapy exercises. The proposed model initially estimates the human pose using MediaPipe in real time. The estimated landmarks serve as input to the K-Nearest Neighbors (KNN) algorithm, which segments exercises into repetitions. Naïve Bayes organizes and classifies movements, considering the correct center point and angular tolerance margins. The developed system achieved an average accuracy of 99.22% for movements performed by the upper limbs.

Keywords: Posture estimation, Health, RGB Scene, Naïve Bayes, Computer Vision.

* Corresponding author.
E-mail address: waltersimoes@gmail.com

10.1016/j.procs.2024.11.132

1. Introduction

The demand for physiotherapy professionals and services increased by approximately 725% after 2019, primarily due to the demands arising from Covid [1]. Regular and correct physiotherapy exercises play a crucial role in patient recovery, joint health, and respiratory maintenance. They help keep patients active and maintain motor capacity as diagnosed by healthcare professionals.

It is not inexpensive or accessible for everyone to train with a trainer, visit physiotherapy clinics, or visit gyms [2]. Another choice is self-training, which entails prerecording the physical therapy routine's steps but does not provide feedback. Injuries can and will cause more harm than good if we don't receive enough guidance on how we should stand, which is precisely why this research is important. So the question that needs to be answered is: how can feedback be given to physical therapy self-coaching to avoid incorrect movements that lead to injury or worsen the problem you are trying to solve?

Using pose estimation techniques, the position of a person at key points can be determined. This will allow us to estimate or evaluate the pose of the person and provide feedback. In this research proposal, we compare the input image from the camera with reference points. If both are the same, we can determine whether the pose of the input image is correct or provide appropriate feedback for correction.

Since the advent of computer vision research, human pose estimation has always been one of the difficult problems to solve [3, 10]. With the increase in computing power, deep learning models have improved the results and are currently the most widely used approach for body pose estimation [4, 5, 11].

Due to the commute in large cities and the difficulty in finding physiotherapists to supervise exercises in patients' free time, many people end up abandoning the treatment established by the physiotherapist [2]. This study aims to fill the gap in monitoring the performance of on-demand physiotherapy routines through real-time feedback.

This work is organized into the following sections: Section 2 handles the selection of related works. In Section 3, we present the architecture designed to solve the proposed problem. Following: The proof of concept is demonstrated in Section 4, which is established by constructing a prototype. Section 5 shows the results and evaluations. Section 6 discusses the results and suggestions for the next stages of this research.

2. Related Works

The development of a research paper requires observing the discussion of related literature and creating models for the development of methodology and the application of testing and evaluation schemes. Thus, the researchers decided that systematic reviews should be used as a strategy to find the state of the art in body motion tracking by computer vision [6]. We searched journals and papers to gain insights into the current state of pose estimation. Other important areas of image processing and semantic segmentation were also considered as relevant to our project, as well as the different approaches adopted to achieve the goal of real-time articulated pose estimation.

Bazarevsky and his co-authors [7] built an architecture based on a reduced convolutional neural network. The goal was to enable its use on mobile devices to track human poses. In this model, 33 keypoints were mapped and grouped so that their processing was light enough to be used in real time. The application presented a performance of 30 frames per second (FPS).

Hao et al. [8] proposed a system that can detect people with the help of multiple uncalibrated cameras. In other words, when a person is detected on a camera, the system can identify which cameras that same person is present on. The research presents a method to automatically recover the movement lines of people in the environment from image analysis. Furthermore, once these lines are initialized, homographs between views can also be recovered.

Bora et al. [5] proposed the development of a tool for non-invasive human motion capture. The approach used a neural network to adjust the musculoskeletal model to provide a kinetic estimate. The authors used a capture with 2D devices and, from the combination of points, provided a 3D estimate. In terms of joint angle errors, the model presented a range of 3.54 degrees to 5.44 degrees, with an accuracy of around 96%. In certain contexts and movements, the model obtained a score of 75% due to occlusions of some joints.

Bittner et al. [9] proposed a system that uses simple kinetic reasoning for forward or backward motion so that it can accelerate within its bound group of minima, called kinematic jump process, for 3D monocular human tracking. This solution can be used for jump diffusion style research. This model uses multiple coordinates to represent joints.

This system tracks the individual using kinematic motion even if the background is blurred. This system also supports long sequence media for motion identification.

Arrowsmith et al. [10] proposed to detect physical therapy postures in home environments using deep learning on a convolutional neural network (CNN), an implementation of the Mediapipe library, and computer vision. The proposed model mapped 12 landmarks targeting the lower back and shoulders and was embedded in a smartphone to use its camera (monocular vision). The authors indicated an accuracy level of around 98.75%.

Zhe Cao et al. [11] proposed OpenPose real-time multi-person 2D pose estimation using Parity Affinity Fields (PAF). This system achieves high-level accuracy and real-time performance. It uses a nonparametric representation method to identify the body parts in an image. It describes that a refinement of only PAF instead of the refinement of both PAF and body part localization results in higher runtime performance and accuracy.

Alex Kendall and his co-authors [12] proposed Posenet. It uses a six-degree monocular relocation system. A convolutional neural network converges all the information into a single image. The algorithm has an image processing layer that allows it to be used in both indoor and outdoor environments. The convolutional network is also used to correct complex problems that are outside the image plane regression.

3. Approaches

This section describes the approaches used to build the models, baselines, and the application of Naïve Bayes to indicate the classification of the user's perceived pose. An overview of the proposed framework is shown in Figure 1. The architecture includes human pose configuration, RGB scene candidate joint generation, and the human skeleton joint point display and joint classification model.

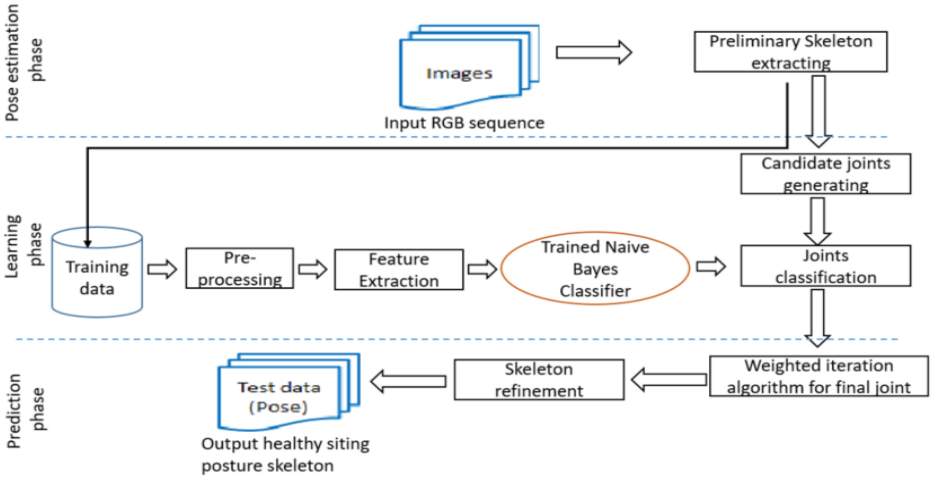


Fig 1. Overview of the framework.

3.1. Image Acquisition

This research project uses an RGB (Red, Green, Blue) camera as a data input source [3]. A square reference box is shown on the screen and the user is asked to stand at a specific distance so that their entire body falls within this square boundary. The camera captures the user's image continuously during the routine and sends the images to the system, which consists of the processing layer and classification of body joint markers through the neural network.

3.2. Human Posture Configuration

Pose detection is done using a pre-trained MediaPipe model. MediaPipe is an open-source, cross-platform, and customizable machine learning solution for real-time media streaming [3, 9]. Some of the solutions provided by the

MediaPipe library include pose estimation, hair segmentation, facial mesh, motion tracking, and Keypoint Neural Invariant Feature Transform (KNIFT). The input data in the MediaPipe library is the image captured by the camera.

The input image is processed by the MediaPipe library to detect key points on the user's body. The output is a list of coordinates on the X, Y, and Z axes for 33 main key points on the human body. This list of coordinates defines the location of each main body part in the input image. Using these coordinates, we can construct an accurate skeletal orientation of the user.

The first 11 landmarks of the Mediapipe (from 0 to 10) are used for the facial labeling procedure. The next 11 landmarks (from 11 to 22) are used for upper body detection. The upper body includes shoulders, elbows, wrists, hands, and about 3 fingers, i.e., little finger, index finger, and thumb on both hands. The 11 key points/landmarks (from 23 to 32) are used to define the lower body, consisting of hips, knees, legs, and feet.

The output of the MediaPipe library is a list of corresponding keypoints in Cartesian X, Y, and Z coordinates [5]. These keypoints can be used to obtain an estimate of the structure and orientation of the human body in a given image or video stream in real time.

3.3. Database of Phisioterapy Pose

A set of angles made between the joints defined each physical therapy posture. These individual angles are used to determine the correct position of the patient's posture in relation to the template (reference image).

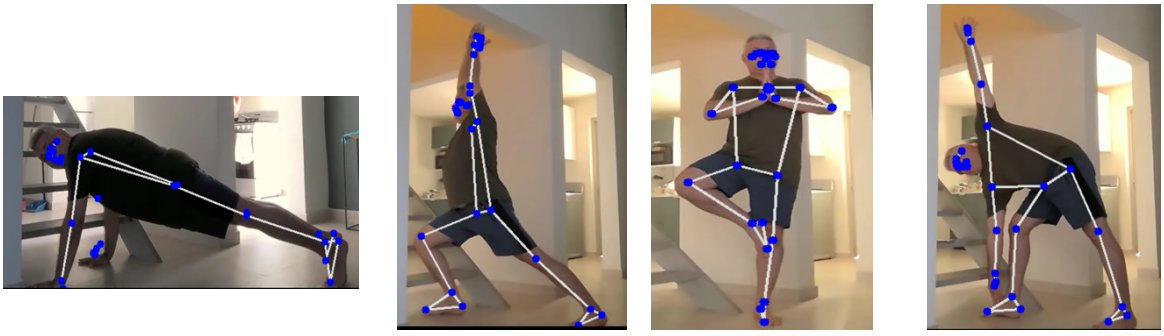
The points identified by Mediapipe are grouped by similarity and arranged in an adjacency matrix using the Naïve Bayes algorithm. Naïve Bayes is an algorithm that generates a table of probabilities from a data classification process, creating classes to be considered when preparing answers, based on pre-established criteria [10].

The research uses as a basis some postures already defined in yoga exercises. Five (04) positions from the Asana bench were selected, called: Plank Pose, Warrior Pose, Triangle Pose and Tree Pose [4, 10].

3.4. Physiotherapy Poses

A set of four poses were used to experiment with identifying the points of interest of the body joints and establish the necessary landmarks for the accuracy calculations. They are: Plank pose, Worrior pose, Triangle pose, and Tree pose (Fig. 2).

The plank pose is a preparatory yoga pose. This pose is performed to prepare the body for more intense yoga exercises/routine [4]. It helps to strengthen the arms, spine and wrist. Fig. 2(a) shows this pose.



(a) (b) (c) (d)

Fig 2. (a) Plank Pose. (b) Worrior Pose. (c) Triangle Pose. (d) Tree Pose.

The warrior pose is used in yoga and physiotherapy to help strengthen the human back, arms, thighs, shoulders, ankles and calves. This pose also helps to stretch the thigh, shoulders, chest, ankles, neck, calves, groin, lungs and navel [4]. The warrior pose is a standing pose (Fig. 2(b)).

The triangle pose gets its name from the distinctive triangular shape formed by the legs, with the patient standing [4]. In this position, the legs are spread at 45 degrees, the knees are bent and the hands are extended with the palms facing downwards. The right hand is then lowered to touch the shin (or floor) of the right foot. And the left hand is stretched upwards, and the body is twisted, which is also repeated on the left side (Fig. 2(c)). This exercise strengthens the knee, thigh, and ankle. Performing this pose also helps to stretch the spine, knee, thigh, shoulder, hip, ankle, hamstring, calf, and chest [4].

The tree pose is a balancing pose [4]. It is also classified as a standing pose (Fig. 2(d)). Performing this pose helps to strengthen the thigh, spine, ankle, and calf.

3.5. Model Formulation

Our goal is to estimate the correct posture in real time. The features and relationships we need to deal with can bring high complexity to the process, making it unfeasible to use. To simplify the process, we chose a Naïve Bayes classifier as our classification model, and trained a classifier for each joint.

In our model, a feature vector represents each candidate articulation joint X = (x 1, …, x n). Each xn is a specific relationship restricted to the correct angle, considering the maximum margins of variation (10%). In this vector, xn ∈ {1, 0}, "1" means this relationship satisfies a mapped condition and "0" means unsatisfied status. The Naïve Bayes classifier modeled the relationships between a specific joint and other identified joints. For an input feature vector X, our classifier assigns a class label ŷ = ck as:

$$\hat{y} = c_k^{argmax} P(Y = c_k) \prod \prod_{k=1}^{n} P\left(X^j = x^j \middle| Y = c_k\right) \quad (1)$$

where the class label is defined as Y = (c 1, …, c k), k ∈ {1, 2, …, K}, K15 = 2, where c 1 is "Acceptable" and c 2 is "not acceptable".

A maximum likelihood method was applied to conduct parameter estimation. In our model, parameters need to be estimated, including P(Y = c k) and P(X (j) = x (j)|Y = c k). And the maximum prior probability P(Y = c k) is,

$$P(Y = c_k) = \frac{\sum_{i=1}^{N} I\ (y_i - c_k)}{N}, k = 1,2, \ldots, K \quad (2)$$

where N is the number of training samples and K = 2.

Assuming that the possible value of j_th feature x^j is {a_j1,…,a_j s_j}, and the maximum likelihood of conditional probability P(X^j= x^j| Y=c_k) is

$$P\left(X^j = a_{jl} \middle| Y = c_k\right) = \frac{\sum_{i=1}^{N} I(x_i^j = a_{jl}, y_i = c_k)}{\sum_{i=1}^{N} I(y_i = c_k)} \quad (3)$$

$$J=1, 2,\ldots, n;\ l=1,2,\ldots, S_j;\ k= 1, 2,\ldots, K; \quad (4)$$

where x (j) i is the j th feature of i th sample, a jl is the l th possible value of j th feature, I is the indicator function. Here we set S j = K = 2 as x n ∈ {1, 0} and c k ∈ {c 1, c 2}.

The Naïve Bayes Classifier trained in this study was used to determine whether the candidate joint is "Acceptable" or "Not Acceptable". For candidate joints labeled as "Acceptable", we conducted a weighted iteration algorithm to calculate their center of gravity, which is considered as the final correct joint. The final center of gravity of the "acceptable" candidates was calculated as follows: denoting J as the coordinates of the candidate joints, J = {(x1, y1), (x2, y2),…, (xi, yi)}. First, we sort these joint points by y-coordinate in ascending order, then select 4 joints to compose a tetrahedron starting from the minimum yi with a step size of 1, and iteratively calculate the volume and center of gravity of the tetrahedron, until the number of remaining joints is less than 4. Then, we calculate the volume-weighted center of gravity of the center of gravity of these tetrahedra and take it as our final center of gravity. The volume of each tetrahedron was treated as the weight of its center of gravity.

The following elements were used in the algorithm: TetrahedronVolume J: used to indicate the volume of the j-th tetrahedron composed of 4 joints. J is represented by: Input: J = {(x 1 , y 1), (x 2 , y 2), …, (x i , y i)}. The sum of all the tetrahedron volumes relates the volumes perceived in each instance of J (AllVolume.push_back(TetrahedronVolume j). Once the volume of the tetrahedra has been calculated, they are

organized in a vector called IterationGravity (IterationGravity.push_back(C_o G_j). The weighted center of gravity (CoG) is obtained by the final volume of all the centers of gravity calculated in IterationGravity. The CoG represents the following operation:

$$CoG = \left\{\frac{\sum_i x_i}{i}, \frac{\sum_i y_i}{i}\right\}.$$

After identifying the points considered "acceptable", the algorithm enters the identification mode for the position selected by the user for training. When the user poses in front of the camera, the frame is captured and the geometric analysis is performed using data from the MediaPipe library. The output of the geometric analysis is the angles made between each joint. If these angles are beyond the limit defined by the user, an alert message is displayed on the screen to indicate the error.

3.6. Pose Comparition

The output of the MediaPipe library contains only the coordinates of the user's main key points in the image [5]. A function is written in the program to take these coordinate data and calculate the angles at each joint. Given three key points, we can easily calculate the angle made between the two lines using analytic geometry. Let the three points be A(x1, y1), B(x2, y2), and C(x3, y3). Let the lines AB and BC intersect at B, then the angle between AB and BC can be calculated as:

The slope of line AB is given as,

$$m_1 = \frac{y_2 - y_1}{x_2 - x_1} \tag{5}$$

m1 is the slope between the line joining point A and B. y1 and x1 are the coordinates of point A and x2 and y2 are the coordinates of point B. This same principle is then applied again to the line BC to obtain the slope of BC line. The slope of line BC is given as,

$$m_1 = \frac{y_3 - y_2}{x_3 - x_2} \tag{6}$$

m2 is the slope between the line joining points B and C, y2 and x2 are the coordinates of point B, and x3 and y3 are the coordinates of point C. Point B is the common point between the three points A, B and C and hence the angle is formed at joint B. This is the angle between AB and BC. Now the angle between AB and BC can be calculated as,

$$\tan\theta = \frac{m_1 - m_2}{1 + m_1 * m_2} \tag{7}$$

In Eq. 7, tan is calculated and can be positive or negative based on the given angle. Taking the inverse of tan, we obtain the angle made at B between AB and BC.

AB and BC can be considered two bones or skeletal structures of the human body. assuming line AB as the elbow and line BC as the hand, the angle made between the elbow and the hand can be calculated in this way. By further applying this analysis to all other joints, we can calculate the angles made at each joint. During the data preparation phase, the angles taken for each yoga pose are calculated in advance and stored in the database. This analysis is done to get all the angles of a particular yoga pose. These angles are calculated for all 5 yoga poses before use and are stored in the database for reference.

3.7. Feedback to the User

Providing feedback to the user is extremely important as it helps guide the user to the correct posture and thus learn how to perform the exercise correctly [10]. Feedback on the user's performance is provided in real time through audio messages and display. Users can observe the correction and make necessary adjustments to their posture to accurately practice the suggested movement routine.

Each user has varying levels of flexibility, meaning that one user may not be able to flex their body as much as another user [9, 11]. To address this issue, a user-changeable threshold parameter is included. A beginner user can set the threshold higher, say 20 degrees, so that they can have a deviation of about 20 degrees in either direction. An experienced user can set it to less than 10 degrees to practice the pose accurately. This feature allows beginner users to slowly and steadily improve their body’s flexibility for performing physical therapy exercises.

In Figure 3, the user has posed at a 180-degree angle. The exercise is still considered valid if the angle is within a 10-degree limit. When such a large deviation occurs, a warning message is displayed on the screen to indicate to the user that the posture needs to be corrected.

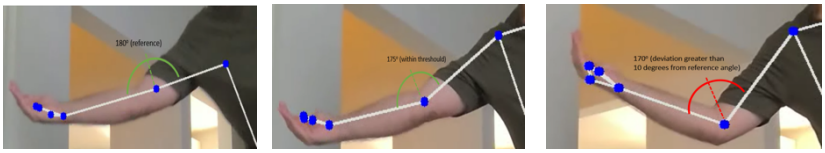


Fig. 3. Gesture execution with angles within and outside acceptable limits for positive classification.

4. Experiments

The user selects which physiotherapy pose he/she will perform. When the user strikes a pose, the angles are compared with the angle data for that specific pose. If there is a deviation of the pose angle made by the user from the reference data, the user is notified about it in real time via the display or audio feedback. The rationale behind choosing angles instead of length for pose detection is because during a physiotherapy routine, the patient may not always be in the center of the frame. But if the patient is assumed to be the center, the angles can be measured.

5. Results

5.1. Experimental Settings

A total of 4 key yoga poses are available for practice with real-time feedback capability on screen or via a wireless headset for correction. We used a 9th Gen Intel Core i7 CPU with 32GB RAM to run the codes, which provided a consistent processing output of 30 frames per second (FPS). Currently, the system only works in two-dimensional space, so there is no depth data considered for calculation.

5.2. Results and Evaluations

Comparing studies can be complex and sometimes unfair, as authors adopt very different methodologies, databases, and image formats. Nevertheless, it is interesting to identify how this research stands in relation to other studies. Using the level of precision as a criterion, the studies are positioned as indicated in the bar chart in Figure 5.

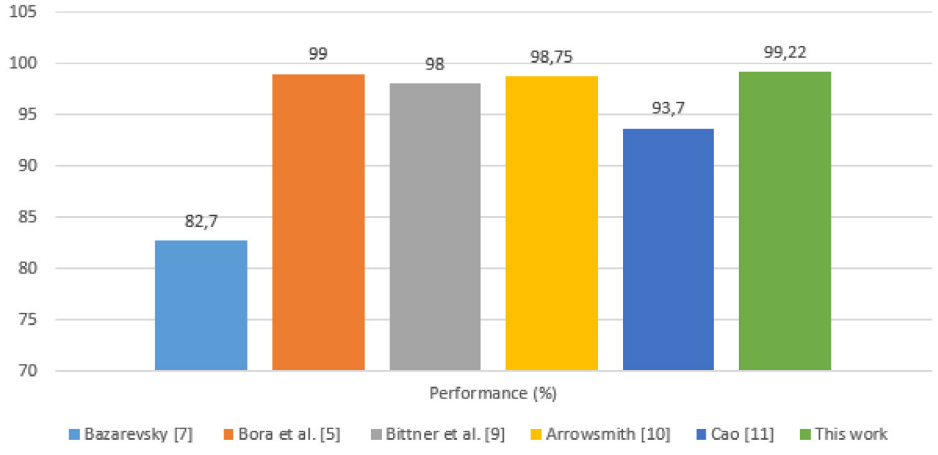


Fig 4. Analyze using the success level as a criterion.

The fundamental difference between this study and the studies developed by Hao and Bittner was considering the object context from a 2D perspective [7, 9]. Bazarevsky and Hao adopted a more complete approach to posture monitoring, inserting 30 to 33 key points into the captured images of people, which caused greater processing consumption, perceived by the frame count every second [7, 8]. Authors Arrowsmith and Cao used an approach with 2D RGB cameras. However, their posture recognition models kept the images in RGB format, which made it difficult to identify certain characteristics in environments without lighting control [10, 11]. In this study, images are converted to the HSV format (hue, saturation and value), which allows greater control over variations in luminosity and thus greater precision in environments where lighting is not possible. is controlled.

6. Conclusion

This research had the target problem to be solved: how can feedback be given to physical therapy self-coaching to avoid incorrect movements that lead to injury or worsen the problem you are trying to solve?

Briefly, the answers obtained are:

This research considered that patients can present the suggested gestures of the physiotherapeutic exercise with small variations in angle and distance; therefore, the limits established were 0.5 m and 5 degrees. This allowed different profiles of people to perform gestures more fluidly.

Another important decision was to adopt a model that considers 9 keypoints to be observed, which reduced processor and memory consumption, leading to a value of 23 frames per second (FPS).

The system built to carry out the proof of concept showed that the proposal is viable for monitoring patients performing exercises without the need for a special trainer. One of the contributions directly linked to the use of the tool is to reduce (or avoid) injuries due to inadequate technique.

The natural evolution of this research is to transform it into a responsive application on Android and the web. Furthermore, the advent of depth sensors in smartphones will help improve the accuracy of pose detection, which in turn helps improve angle calculation for comparison.

References

[1] De Souza, J. C., Saraiva Ferreira, J., & Rocha Macedo de Souza, G. (2021). Reabilitação funcional para pacientes acometidos por covid-19. Revista Cuidarte, 12(3).

[2] da Conceição Furtado, M. V., da Costa, A. C. F., Silva, J. C., do Amaral, C. A., do Nascimento, P. G. D., Marques, L. M., ... & de Moraes, R. M. (2020). Atuação da fisioterapia na UTI. Brazilian Journal of Health Review, 3(6), 16335-16349.

[3] Trumble, M., Gilbert, A., Malleson, C., Hilton, A., & Collomosse, J. (2017, September). Total capture: 3d human pose estimation fusing video and inertial sensors. In Proceedings of 28th British Machine Vision Conference (pp. 1-13).

[4] Yadav, S. K., Singh, A., Gupta, A., & Raheja, J. L. Real-time Yoga recognition using deep learning. Neural computing and applications, 31, 9349-9361, 2019. https://doi.org/10.1007/s00521-019-04232-7;

[5] Bora, J., Dehingia, S., Boruah, A., Chetia, A. A., & Gogoi, D. (2023). Real-time assamese sign language recognition using mediapipe and deep learning. Procedia Computer Science, 218, 1384-1393.

[6] Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A. M., & Wang, X. (2021). Computer vision techniques in construction: a critical review. Archives of Computational Methods in Engineering, 28, 3383-3397.

[7] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). Blazepose: On-device real-time body pose tracking. arXiv preprint arXiv:2006.10204.

[8] Hao, S., Liu, P., Zhan, Y., Jin, K., Liu, Z., Song, M. & Wang, G. (2024). Divotrack: A novel dataset and baseline method for cross-view multi-object tracking in diverse open scenes. International Journal of Computer Vision, 132(4), 1075-1090.

[9] Bittner, M., Yang, W. T., Zhang, X., Seth, A., van Gemert, J., & van der Helm, F. C. (2022). Towards single camera human 3d-kinematics. Sensors, 23(1), 341.

[10] Arrowsmith, C., Burns, D., Mak, T., Hardisty, M., & Whyne, C. (2022). Physiotherapy exercise classification with single-camera pose detection and machine learning. Sensors, 23(1), 363.

[11] Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7291-7299).

[12] Kendall, A., Grimes, M., & Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE international conference on computer vision (pp. 2938-2946).