Learning-based Video Analysis for Optimizing Exercise Posture

Learning-Based Multimedia Processing, 2nd semester of 2022/23

1st Gustavo Caria
MSc in Applied Mathematics and Computing
Nº 92633
Lisbon, Portugal
gustavocaria@sapo.pt

2nd João Gonçalves
MSc in Eletrical and Computer Eng.
Nº 85211
Lisbon, Portugal
jpedrogoncalves@tecnico.ulisboa.pt

3rd Leonardo Brito
MSc in Data Science and Eng.
Nº 105257
Lisbon, Portugal
leonardo.amado.brito@tecnico.ulisboa.pt

***Abstract*—Pose estimation is the task of accurately estimating the positions and orientations of key points on human bodies or objects in images or videos. It plays a crucial role in various applications, including action recognition, motion analysis, human-computer interaction, virtual reality, augmented reality, robotics, and object tracking. There are two main approaches to pose estimation: 2D and 3D, but we focused on 2D.**

2D pose estimation focuses on estimating the 2D coordinates of key points in an image. Convolutional neural networks (CNNs) are commonly used for this purpose, but are also used methods like Angle Analysis and Joint Distances.

Recent advancements in pose estimation, driven by annotated datasets, improved deep learning techniques, and computer vision advancements, have led to more accurate and robust pose estimation systems. These advancements have opened up avenues for various applications, empowering industries to leverage pose estimation technology for enhanced human-computer interaction, immersive experiences, robotics, and more.

***Index Terms*—Angle Analysis, Joint Distances, Convolutional Neural Network, Deep Learning, Pose Detection, Robust Pose Estimation Systems.**

I. Introduction

Optimizing posture during exercise is a crucial aspect of any training routine. Incorrect posture can lead to ineffective results and potentially serious injuries. However, not everyone has access to personal trainers or fitness experts who can guide them in the proper execution of exercises. This problem becomes even more relevant in the home workout environment, where individuals often perform exercises without any supervision.

Therefore, it is essential to have a system that can analyze posture during exercise through videos and provide real-time feedback for posture correction. This system would use machine learning techniques to analyze the user's posture during exercise, compare it with the ideal posture for that exercise, and provide feedback.

With the increasing number of people training at home, maintaining proper posture during exercise is crucial to prevent injuries, maximize exercise efficiency, promote correct body mechanics, and support healthy aging.

Others have addressed this issue using diverse models, such as OpenPose and MediaPipe, which employ a CNN-based approach. For the development of our project, we choose three methods to tackle the problem.

To address this issue, an image analysis system based on learning techniques was employed. It utilized deep learning techniques, particularly CNNs and pose estimation models, to recognize and analyze the user's posture during various exercises. The observed posture was then compared with the ideal posture for that specific exercise, providing feedback to assist the user in correcting their form.

The proposed system consisted of 5 main modules, as shown in Fig 1

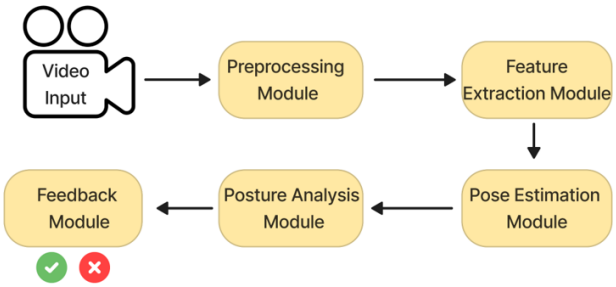

Fig. 1. Architecture pipeline diagram.

II. Dataset

We had two datasets, one for train and one for test. The dataset for train was derived from a video dataset, that was divided into 50 correct and 50 incorrect samples of the push-up exercise. In each video, we extracted two frames: one when the subject was in an upper position with arms unflexed, and another when they were in the lowest position possible. We developed code for the automatization of this task. The code was able to, in recurse of buttons, travel between the videos of the dataset, go frame to frame in each video, and save the specific frame that we wanted. Using two extreme frames representing different phases of the movement (such as

push-ups in the high and low positions) allows for capturing information about pose variation and joint angles throughout the movement. For the test, we used a dataset with a few examples taken from the Google search engine and from videos created by us and with of one of us doing exercise.

III. Methods

In order to obtain a model as robust as possible, seeking more reliable and accurate results, a combination of various methods was used. These include analysing angles between joints, distances between joints, and the fine tuning of pre-trained Deep Learning models.

A. Angle Analysis

An intuitive way of analyzing the posture for a given exercise is to check the angles between different joints of the body. This approach allows for an objective evaluation of the exercise since it will rely on precise angle measurements that can be compared to established standards [1].

The MediaPipe model used already provided accurate coordinates for the body joints that formed the crucial angles for our analysis. In total MediaPipe creates 33 pose landmarks, but only 12 of those were useful when analyzing angles for the push-up exercise, as shown in Fig 2.

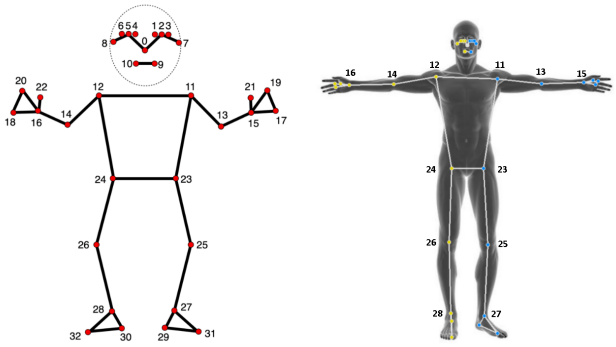

Fig. 2. Illustration of pose landmarks, all (left) and useful (right).

We mainly analyzed angles of 180º, corresponding to the torso, legs, and arms that should be in a straight position. For the downward position only the legs and torso are taken into consideration since the angle given by the arms position will vary greatly according to the technique used to perform the exercise. The upward position considers 4 more angles as shown in Fig 3.

The value of the angles was calculated using cosine similarity since we knew the coordinates of all the needed points, as shown in 1.

$$cos(\theta) = \frac{A \cdot B}{||A||\,||B||} \tag{1}$$

To access the correct posture of the sample we had to choose an ideal angle and a threshold, that when surpassed would invalidate the given pose. The obvious conclusion would be to set the ideal angle to 180º for most of the angles, but upon

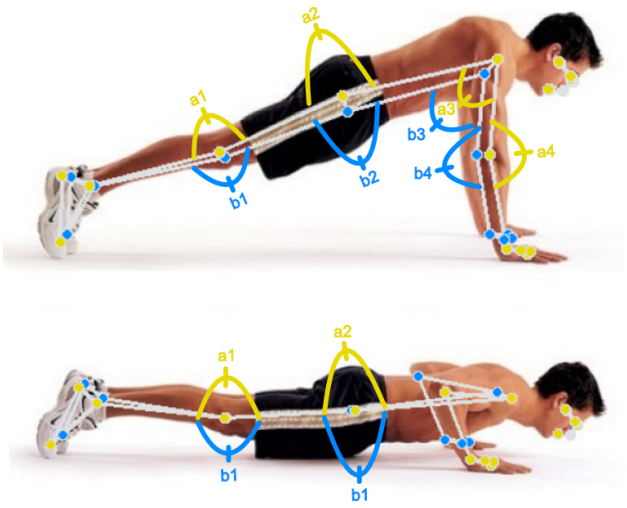

Fig. 3. Chosen angles for upward and downward positions.

calculating the mean and standard deviation of the angles using a dataset of correct poses we reached the results shown in I.

Angle	Ideal angle (º)	Threshold (º)
a1	*165/160 (up/down)*	*30*
a2	*170*	*20*
a3	*170*	*20*
a4	*170*	*20*
b1	*165/160 (up/down)*	*30*
b2	*170*	*20*
b3	*170*	*20*
b4	*170*	*20*

TABLE I
IDEAL ANGLES AND THRESHOLDS.

For example, angle *a1*, the one in the back of the knee represented in Fig 3, has an ideal angle of 165º (for the upward position) and a threshold of 30º, which means that we consider that the angle is correct if the angle varies 30º for the side of the angle that the leg bends. This happens because, if 165º is the leg unflexed, it can only bend to one side, otherwise it would break, and the variance is 30º at maximum and not 60º (30º for each side). This is true for every angle except the hips where the threshold will happen for both sides.

B. Joint Distances

Another pose validation method used was the measure of distances between joints [1]. First, we obtained the coordinates of the desired joints, as before. Using the dataset with correct postures an average of the correct joint positions was found. Then a method was developed to normalize and center the landmarks of the frame we want to analyze, using as reference the ideal position found before, as seen in 5.

First, we centered the image:

- Specified two specific landmarks, landmarks 23 and 24 (hip), to be used as the center points.
- Calculated the centroid of these two landmarks by taking the mean of their x and y coordinates.

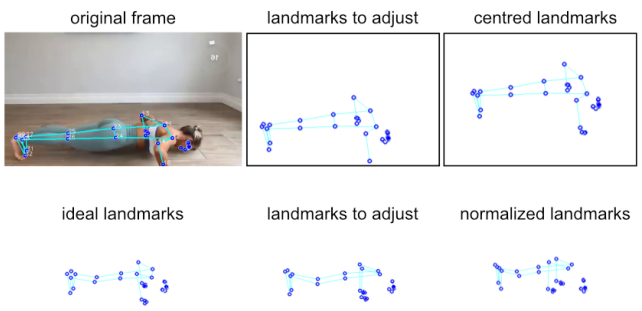

Fig. 4. Centralization (up) and normalization (down) of landmarks.

- Subtracted the centroid from all landmarks in the array to shift the coordinates relative to the centroid.
- Reposition all landmarks a little bit at the center of the image with a specific value.
- Convert the centered landmarks back to their original format.

Then we did a normalization. The normalization consisted in:

- Find the best pose landmarks of the dataset that we used by looking at them.
- Do a mean of all distances between the landmarks of all good examples.
- Convert all distances of the image we want to normalize to all landmarks distances of the normalized ones, but keeping the angles of the image not normalized. We started always with the distance of the hip landmarks, and then we kept passing to all distances without repeating anyone.

Once the frame is ready to be analyzed, the computations are very simple. The distances between key joint positions are compared to the ideal distances and the posture is considered incorrect once a threshold is surpassed. To choose an ideal threshold an Evolutionary Algorithm was developed.

We used a genetic algorithm is used to optimize a set of parameters called alphas for the task. The goal is to find the best values for these parameters that result in high accuracy for classifying push-up movements. The alphas are the factor that we used to multiply the threshold values of this joint distances method. So, we considered that the distance was correct if the difference between ideal values (that was the mean of good examples) and the distance between joints of the test example were smaller than the threshold value times the alpha value.

The alpha vector consists of different alpha values, which control the importance of specific landmarks in the classification process. The range for these values is defined as the alpha minimum and the alpha maximum.

The initial values of alpha were based on the standard deviation of each joint (they are 12) and the algorithm found the best alphas for each joint distance to maximize the accuracy of the correct ones.

This method was chosen over grid search because we did not have the necessary hardware for this type of computation.

C. Deep Learning Image Detection

We used several pre-trained models to do the first detection: ResNet 152, ResNet 152, ResNet 50, 34 and 18. ResNet (Residual Network) is a popular deep neural network architecture that introduces the concept of residual connections to address the degradation problem faced by deep networks. The numbers (e.g., 152, 50, 34) in ResNet represent the total number of layers in the network, including both convolutional layers and fully connected layers. If the model has a bigger number of layers, that results in a larger model size compared to the others. Of course, if the model is deeper and has more potential intricate features and representations, leading to potentially better performance on complex tasks given sufficient training data, it will be slower and will need more resources, although we did not see many differences. Knowing this, we choose the less complex model, the Resnet18, and fine-tuned only the last layer to give us 2 outputs.

For pre-processing of the data, data augmentation introduced variations and diversity into training data by applying transformations such as cropping for the standard measures received by Resnet18, flipping, rotation, and changes in brightness or contrast. This technique helps the model generalize better to unseen data and improves its ability to capture important features. By combining a less complex model with augmented data, you can mitigate the risk of overfitting and improve the model's performance. Also, the images were normalized regarding the standard deviations and mean of each model, the one for up push-ups and down push-ups. Normalization was used every time we tested one image.

To increase the accuracy of the models, we thought of only taking the skeleton with the landmarks or overlapping the images with the landmarks for the train because achieving high accuracy with a small dataset is challenging. Since Resnet was done not for skeletons but for human body detection, we changed the dataset for training with landmarks overlapping and guaranteed that every image before classification in the test followed the same procedure. We used MediaPipe to predict landmarks and removed the images of training that was not well applied.

As was said in one of the paragraphs before, we created 2 models, one for push-ups and down push-ups. We did that because of the nature of each position of the push-ups (up and down) since they are quite different. One was trained with the dataset of ups and the other with downs.

IV. Final Model and Result Analysis

In order to reach optimal results an ensemble method was developed. This will improve our model for several reasons, mainly:

- Improved Predictive Performance: ensemble methods combine the predictions of multiple models, each with its own advantages and disadvantages. By combining them it's possible to extract the best from each one.
- Reduction of Overfitting: overfitting occurs when a model learns to fit the training data too close to the model it has

trained on, resulting in poor generalization to unseen data. By combining multiple models, ensemble methods will be less impacted by individual model biases and variance, leading to a better generalization of unseen data.

- Increased Robustness: Ensemble methods are typically more robust to noise and outliers in the data. If an individual model is sensitive to certain outliers or noisy instances, the ensemble can mitigate their impact by considering the predictions from other models that are less affected by those instances.

A. Ensemble

We ensembled all methods in one last classification in order to have a conjunct method to determine if the position is correct or not. First, we created a method to identify if the position of the person was up or down. We took the movement frame by frame of each video.

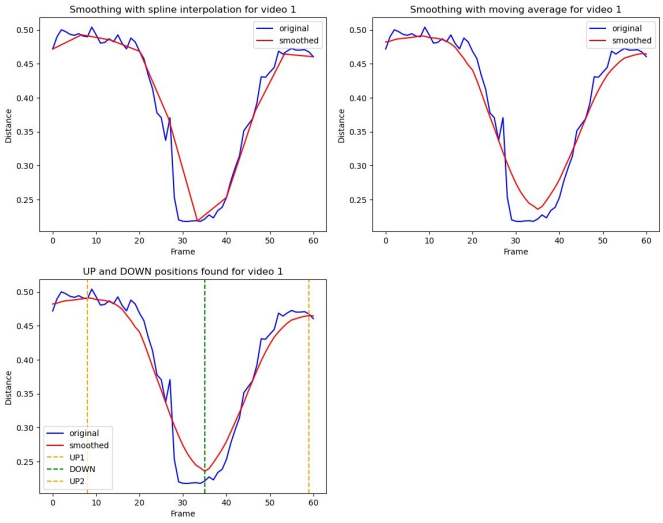

Fig. 5. Graphs of movement of body (up and down) and smoothing of different methods

The graphs are the movement of a push-up when the person is up and going down and vice-versa. Since the original graph was difficult for discovering the minimum and maximum in order to discover if the person is up or down, we smoothed the graph. We used splines and moving averages, but as the graphs above show, the moving average was the best to smooth.

After that, we create a method to, given a video, discovered the up and down of the push-up, and then these two positions were classified by the three methods that we developed.

The chosen approach involved each method performing three classifications for each video, where each video contains two upward poses and one downward pose. If all three classifications indicated (of the respective method) that all poses were correct, the exercise was considered correct. Otherwise, it was deemed incorrect.

Next, the final classification from each method for each video was combined into a single result. If at least two methods indicated that the poses were correct, the exercise was considered correct. If fewer than two methods agreed on the correctness of the poses, the exercise was considered incorrect.

During the development of the three methods, we gave priority to a better classification of a good position to the detriment of a good classification of a bad position. We had this decision because we considered that it is better to have a position well performed with the incorrect classification than a position badly performed with the incorrect classification since one of the main goals is to prevent injuries.

The results given were 97,83% accuracy for the training data that have a label that the exercise is with the correct pose.

V. Conclusion

Our approach was multifaceted, involving a blend of various techniques - joint angle analysis, joint distance measurement, and deep learning image detection. These techniques were subsequently consolidated into an ensemble for optimal performance.

The final model combines the predictions of all these techniques to offer an efficient and robust assessment of the user's posture during push-ups, making the system versatile and adaptable to different users and environments.

The outcomes demonstrated that our system is capable of accurately detecting and analyzing posture during push-ups, thereby supporting individuals in their home workout routine, encouraging proper posture, and ultimately preventing potential injuries.

In the future, this system could be expanded to accommodate a wider range of exercises, offering a comprehensive virtual fitness assistant. Furthermore, improvements could be made by incorporating a larger dataset for training, using more sophisticated pose estimation models, or integrating real-time feedback mechanisms that guide users to correct their postures instantly.

Thus, the present work marks a promising stride in the application of machine learning techniques to enhance the safety and efficiency of home workouts, potentially transforming the landscape of personal fitness.

References

[1] Hilman Zafri bin Mazlan, "FitAI: Home Workout Posture Analysis using Computer Vision", 2022.