# Human pose estimation in fitness

Rita Rodrigues (54859)

FCUL fc54859@alunos.fc.ul.pt

**Abstract.** Although fitness exercises are fundamental to a healthy lifestyle, the incorrect form while performing these exercises can lead to injuries. In this report, Pose Trainer, an application that aims to prevent injuries by providing feedback to improve form, is presented.

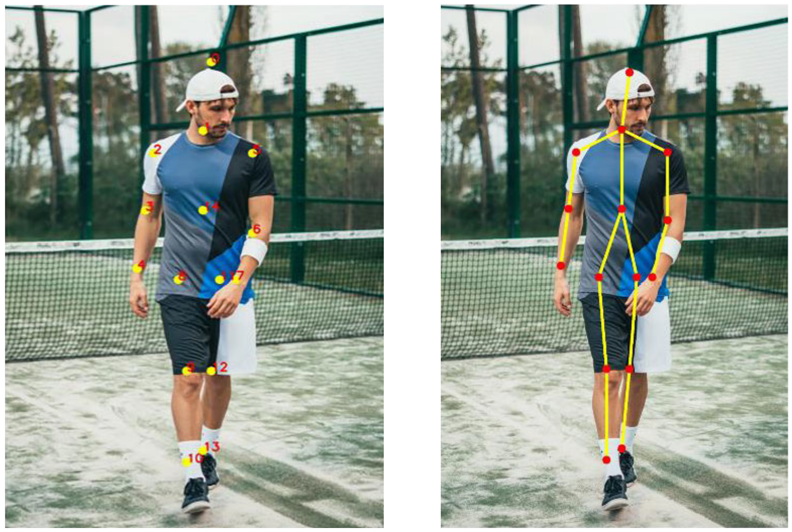
**Keywords:** Pose Trainer · OpenPose · Human pose estimation.

## 1 Introduction

One essential thing to a healthy life is exercise. It is a fundamental factor to minimize the risk of chronic diseases and provides benefits for mental health. But sports and weightlifting come with a great risk of injuries, as a result of a lack of formal training through classes or a personal trainer, muscle fatigue, or even the use of too much weight. Consequently, people do not use the proper form in exercises, which can be potentially dangerous. Weights involved in exercises such as squats, deadlifts, and shoulder presses can cause severe injuries to the muscles or ligaments. In this essay, a brief introduction to human pose estimation is given followed by an explanation of OpenPose, which is used in Pose Trainer, the principal focus of this essay. Pose Trainer is an application that detects the user's exercise pose and provides personalized, detailed recommendations on how the user can improve their form. Then, the overall pipeline is described and a detailed example of an exercise is given. In addition, some results and conclusions are analyzed. The essay ends with an ethics analysis, some examples of companies using the technology, and challenges of human pose estimation in Artificial Intelligence (AI) fitness apps.

## 2 Human pose estimation

Human pose estimation is a field of computer vision that aims to find the position of body keypoints of a person, e.g. head, hand, knee, from a given image or video. This process is divided into two main steps. The first step consists of localizing human body joints/keypoints. The objective of the second step is to group those keypoints into a valid human pose configuration (Fig. 1). In this way, human pose estimation, using image-based observations, yields an articulated human body [1].



(a) Localization of keypoints (b) Group keypoints

Fig. 1: Steps in human pose estimation

Applications of human pose estimation include video surveillance, human-computer interaction, digital entertainment, medical imaging, and sports scenes. Main challenges are variation of body poses, complicated background, and depth ambiguities [2].

## 3 OpenPose

OpenPose estimates multi-person 2D poses in real-time. It was developed by researchers at Carnegie Mellon University and can be considered state-of-the-art in real-time human pose estimation.

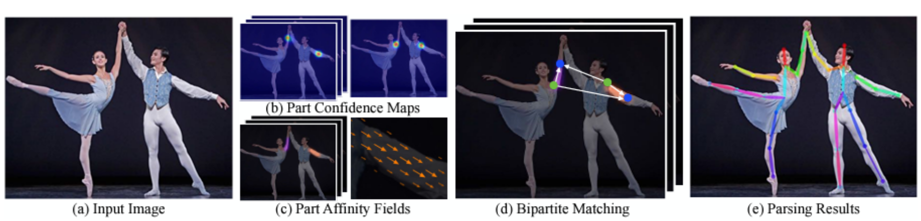



Fig. 2: Overall pipeline

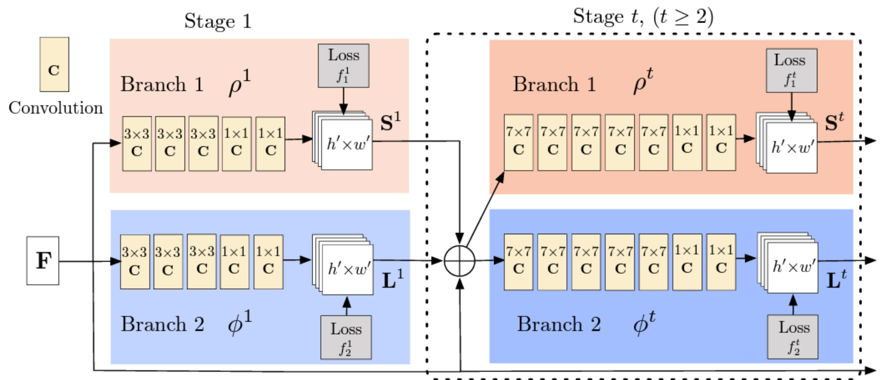



Fig. 3: Architecture

The overall pipeline is represented in Fig. 2. First, an input RGB image (Fig. 2a) is fed as input into a two-branch multi-stage Convolutional Neural Network (CNN), whose architecture is represented in Fig. 3. Two-branch means the CNN produces two different types of outputs. The top branch, represented in beige, predicts Part Confidence Maps $\mathbf{S^t}$, which shows the confidence in the specific location of different body parts such as the left elbow, while the bottom branch, represented in blue predicts Part Affinity Fields (PAFs) $\mathbf{L^t}$, which indicate the degree of association between different body parts. Multi-stage means that at the first stage, the network produces an initial set of confidence maps and a set of PAFs, then, in each subsequent stage, the predictions from both branches along with the original image are concatenated (represented by the + sign) and used to produce more refined predictions. Finally, the confidence maps and the PAFs are parsed by greedy inference (Fig. 2d) to output the 2D keypoints for all the people in the image (Fig. 2e) [3].

## 4 Pose Trainer

Pose Trainer [4] is an application that detects the user's exercise pose and provides personalized and detailed recommendations on how the user can improve their form. The goal is to help prevent injuries and improve the quality of people's workouts with just a computer and a webcam. In the first step of Pose Trainer, visual data is given, which could be an RGB image and/or a depth map, and then a trained model predicts a person's joints as a list of skeletal keypoints. The second step consists of detecting the quality of a user's predicted pose for a given exercise with the use of heuristic-based and machine learning models, using the poses and instruction of personal trainers and other qualified professionals as the ground truth for proper form. The full application uses these two steps and can take a video of an exercise and provide useful exercise form feedback to the user.

### 4.1 Technical Approach

Now a description of each stage of Pose Trainer will be made. The overall pipeline is represented in Fig. 4.

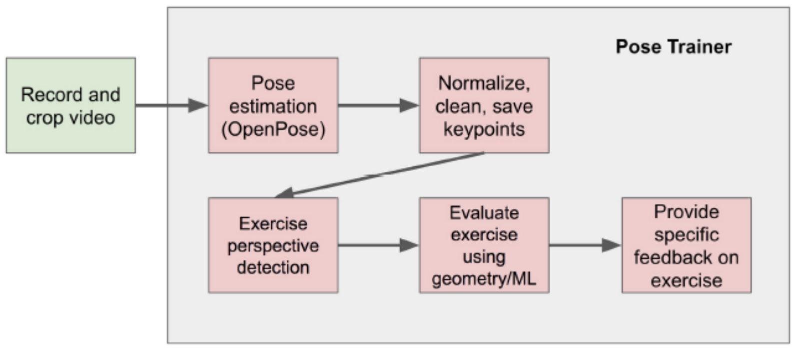



Fig. 4: Pose Trainer pipeline

Pose Trainer starts with the user recording a video of an exercise and ends with the application providing specific feedback on the exercise form to the user. First, the user records a video performing a specific exercise. There are no requirements on camera type or distance from the camera and the application supports all common video formats. The user himself trims the video using the preferred software so that it includes only the frames of the exercise. The next step is pose estimation. The authors experimented with multiple state-of-the-art pose estimators and chose to use the pre-trained model, OpenPose, for pose detection. The output of OpenPose consists of lists containing the coordinate predictions of all keypoint locations and their corresponding prediction confidence. The authors consider the predictions of 18 keypoints of the pose. After using the model, there's a need to generalize the application to account for differences in body length measurements, distance from the camera, among others. Because of that, the pose is normalized based on the torso's length in pixels. The torso length is calculated by the average distance from the neck keypoint to the right and left hip keypoints. The authors observed that this normalization worked extremely well: the torso length stays very constant through all the frames of input videos. Consequently, distances are represented as ratios of torso length: for example, an upper arm length of 0.6 means that the upper arm is 0.6 the length of the torso. Some exercises depend on the camera perspective. For example, the bicep curl exercise is recorded from the side of the body and could be performed with either the left or right arm. To detect the perspective of the video, the application identifies which arm is performing the exercise by measuring which keypoints are most visible (left or right side keypoints) throughout all frames of the exercise. The following step consists of evaluating the exercise.

Two approaches are used: geometry and machine learning. In the geometry evaluation, body vectors from keypoints of interest are computed. Then, personal training guidelines and the authors recorded videos are used to design geometric heuristics, evaluating on the body vectors. This evaluation will become clearer with the example of a bicep curl. The second approach to evaluate exercise posture uses machine learning. Dynamic time warping (DTW) [5] is a metric used to measure the non-linear similarity between two time series. In this case, the authors compute the DTW distance of an input keypoint sequence with all the training sequences and built a binary nearest neighbor classifier that predicts "correct" or "incorrect" form based on the DTW distances. In the final step, specific feedback is provided on the exercise.

The paper presents quantitative and qualitative results of Pose Trainer on four different dumbbell (free motion) exercises: bicep curl, front raise, shoulder shrug, and standing shoulder press. In the following section, an analysis of bicep curl will be made.

### 4.2 Example of bicep curl

The single arm bicep curl is a dumbbell exercise that isolates the biceps. The right way to do the exercise is to lift the dumbbell from a resting extended position, with rotation around the elbow, while keeping other parts of the body still. The most common mistakes while doing this exercise are to use the shoulder to help swing the weight up, and thus rotating around the shoulder, as well as not lifting the weight fully up. The output of a correctly performing bicep curl is represented in Fig. 5.

```
python main.py --video videos\bicep_good_1.mp4
 ↪  --output_folder temp --mode evaluate
 ↪  --exercise bicep_curl
processing video file...
Exercise arm detected as: right.
Upper arm and torso angle range:
 ↪  21.150955500327434
Upper arm and forearm minimum angle:
 ↪  40.74447650965106
Exercise performed correctly!
Exercise performed correctly! Weight was lifted
 ↪  fully up, and upper arm did not move
 ↪  significantly.
```

Fig. 5: Output in a correctly performed bicep curl

To help avoid the mistake of using the shoulder to help swing the way up, the authors consider, in the geometric algorithm, the angle range between the upper arm and the torso, measuring whether the user rotates the shoulder when lifting.

In Fig. 6 the left image shows a proper form, where the upper arm is parallel to the torso, so the angle between the upper arm and torso is small and does not change. The right image shows an improper form, where the upper arm has moved during the curl; this happens mainly because of the use of too much weight. The angle in this case changes, which should not.

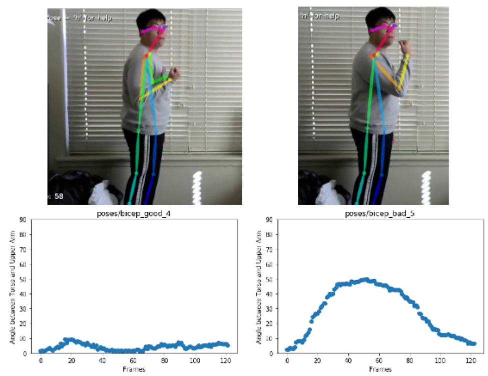



Fig. 6: Angle between upper arm and torso

If the angle is above 35 degrees, the application flags this as too much shoulder rotation. In Fig. 7 is an example of Pose Trainer's output on a bicep curl where there is too much shoulder rotation.

```
C:\Users\stevenzc\Documents\git\pose-trainer>python
  ↪  main.py --video videos\bicep_bad_1.mp4
  ↪  --output_folder temp --mode evaluate
  ↪  --exercise bicep_curl
processing video file...
Exercise arm detected as: right.
Upper arm and torso angle range:
  ↪  35.23131076818897
Upper arm and forearm minimum angle:
  ↪  31.89380019853305
Exercise could be improved:
Your upper arm shows significant rotation around
  ↪  the shoulder when curling. Try holding your
  ↪  upper arm still, parallel to your chest, and
  ↪  concentrate on rotating around your elbow
  ↪  only.
```

Fig. 7: Output of an incorrectly performed bicep curl

To help with the mistake of not lifting the weight fully up, the authors consider the minimum angle between the upper arm and forearm, measuring how high the user lifts. In Fig. 8 the left image shows proper midpoint form: the arm contracts until the forearm is as close to parallel with the torso as possible. In this case, at the midpoint of the exercise, the angle should be small, which means that the arm has fully contracted. The right images show an incomplete exercise and it's possible to observe in the plot that the angle does not get that small.

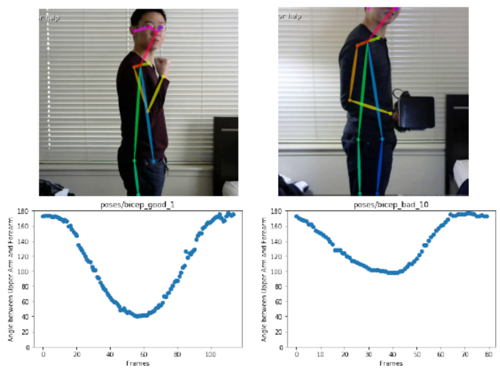



Fig. 8: Angle between upper arm and forearm

If the angle is above 70 degrees, this is flagged as not curling the weight all the way up.

### 4.3 Results

For each of the four dumbbell exercises, the dataset of the videos is split into a training and a testing set. Precision, recall, and F1 are summarized in Table 1.

| | **Precision** | **Recall** | **F1 Score** | **Examples** |
|---|---|---|---|---|
| | | Bicep Curl | | |
| Correct | 0.80 | 1.00 | 0.89 | 4 |
| Incorrect | 1.00 | 0.67 | 0.80 | 3 |
| **Avg/Total** | **0.89** | **0.86** | **0.85** | **7** |
| | | Front Raise | | |
| Correct | 1.00 | 1.00 | 1.00 | 6 |
| Incorrect | 1.00 | 1.00 | 1.00 | 6 |
| **Avg/Total** | **1.00** | **1.00** | **1.00** | **12** |
| | | Shoulder Shrug | | |
| Correct | 1.00 | 0.75 | 0.86 | 8 |
| Incorrect | 0.71 | 1.00 | 0.83 | 5 |
| **Avg/Total** | **0.89** | **0.85** | **0.85** | **13** |
| | | Shoulder Press | | |
| Correct | 0.67 | 0.86 | 0.75 | 7 |
| Incorrect | 0.83 | 0.62 | 0.71 | 8 |
| **Avg/Total** | **0.76** | **0.73** | **0.73** | **15** |

Table 1: Confusion matrix for the DTW classification model

It is possible to observe that the scores are overall good. For front raise, the model was able to correctly classify all examples of both correct and incorrect exercises. It is also possible to see that the F1 score is higher for correctly performed exercises.

### 4.4 Conclusion and challenges of the paper

This paper introduces Pose Trainer, an end-to-end computer vision application that provides personalized feedback on fitness exercise form through pose estimation, visual geometry, and machine learning. Some challenges include exporting Pose Trainer to smartphones through an application that allows users to record a video and get pose feedback anywhere and anytime; improving the pose feedback, providing specific suggestions on where the user's pose needs improvement; and improving graphics, like comparing the user's labeled pose diagram to a labeled pose diagram of a ground truth trainer.

## 5 Conclusion

Fitness mobile apps are in trend as more people become health-conscious. AI has several benefits in the development of fitness apps, such as providing form feedback. Apps that use human pose estimation usually follow the following steps:

- When the user starts the app, the camera records the user's movement during the exercise
- The recorded video will be split into individual frames, the user's keypoints are detected, and a virtual skeleton is formed
- The virtual skeleton is analyzed through AI, and mistakes in the exercise technique are shown
- The user receives the description of mistakes made along with recommendations on how to solve them.

Personally, I do not think these apps will ever replace personal trainers because they are the ones that really know how to perform exercises correctly. An app that uses pose estimation in fitness is Vay. This app provides 30 landmarks on and around the body, at precise joint rotation axes; automatic repetition counting, clocking, and progress tracking during workouts; calculation of time under tension and at what velocity movements are performed; precise computation of relevant metrics like angles, distances, velocities; measures if users go all the way through the entire exercise motion and form correction. Another remarkable example is Sword Health, a healthcare company that provides virtual and digital physical therapy founded in Portugal in 2015. Their digital therapist provides the users with instruction and real-time feedback so they know they are doing the exercises correctly.

One of the challenges of human pose estimation in AI fitness apps is body specifics for men and women: to provide accurate results for both male and female users, it is necessary to consider the physiological difference between the body of men and women and train the model with images of both. Another is the different specifics of physiology: all of us have some disproportionalities, and the apps should be able to analyze the user's body properly. Also, frontal view error: most of the datasets used to train models in human pose estimation are with a frontal view, and the user should not be forced to have the camera at that angle.

## References


1. T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang and C. Yang, "The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation", in IEEE Access, vol. 8, pp. 133330-133348, 2020, doi: 10.1109/ACCESS.2020.3010248.
2. Zhao Liu, Jianke Zhu, Jiajun Bu, Chun Chen, A survey of human pose estimation: The body parts parsing based methods, Journal of Visual Communication and Image Representation, Volume 32, 2015, Pages 10-19, ISSN 1047-3203, https://doi.org/10.1016/j.jvcir.2015.06.013.
3. Cao, Zhe, et al. "Realtime multi-person 2d pose estimation using part affinity fields." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
4. Chen, Steven, and Richard R. Yang. "Pose Trainer: correcting exercise posture using pose estimation." arXiv preprint arXiv:2006.11718 (2020).
5. Berndt, Donald J., and James Clifford. "Using dynamic time warping to find patterns in time series." KDD workshop. Vol. 10. No. 16. 1994.