Pose Based Form Correction Trainer

Harish Srinivasan
Computer Science
University of Texas at Dallas
hxs200044@utdallas.edu

Aditya Singh
Computer Science
University of Texas at Dallas
axs210045@utdallas.edu

Shejo Mathew
Computer Science
University of Texas at Dallas
sxm200018@utdallas.edu

Abstract

Practicing proper form when performing exercises and other physical activities is crucial for better performance, effectively targeting specific muscle group and avoiding strain and injury. Finding a personal trainer or a coach requires both effort and money and the limited physical contact during Covid-19 has worsened the situation. In our work, we propose Form Correction Trainer, an application which utilises state of the art pose estimation frameworks combined with verification and feedback module that allows people to learn and practice their form for exercises, yoga and sports correctly by themselves and reduces the need for physical presence and costs for trainers. We use Pose estimation approach to solve the task of estimating a person's exercise pose from an image or video using spatial locations of key points.

1. Introduction

Good posture when performing any physical activity is immensely helpful for the body to function effectively and avoiding muscle stress and injury. When we are exercising, regardless of the exercise being performed, our body will be in several different positions. It is even more important when working out with weights, such as dumbbells for strength training or increasing exercise intensity to a higher level to maintain proper form and posture. We propose Form Correction Trainer, a software application that helps users maintain correct form when performing any physical exercise or activity while also being cost efficient. The first module of our application will involve human pose detection.

Our model will first correctly identify the predicted keypoints on the person in the video/image frame. Then we will verify if the posture of the exercise being performed by the user is correct. In this method, we plan on using two different approaches - geometric angles between different keypoints and a Machine Learning approach.

Finally, we give the user useful feedback on their poses if they were performing it incorrectly so that they can improve their form. We also provide a rep counter for easy counting. This whole end to end application is designed to be easily accessible to users performing their exercises in different environments such as their home, gym or a field. Our report is organised as follows:

1. Related Work: We discuss the corresponding works done in this field and the current motivation for our problem statement
2. Method: The methodology followed by our model is explained here in this section
3. Experiments: The experiments we have performed so far in this task have been discussed here
4. Conclusion: The summary of our approach and future work to be done is explained.

2. Related Work

Various methods have been proposed for pose estimation that involves using wearable devices and webcam.

T. Zhu et al [1] have used a two-stage process for detecting multiple poses, identifying people and then detecting their key points. For detection and regression, X. Chu et al [2] use CNN's generalization. It estimates human poses using a shared CNN through common input of detection and regression sub network. When multiple people are in the real time frame, Z. Cao, T. Simon and others [3] used part affinity fields that extracted features with the first 10 layers of VGG-19 [4] without the need to detect individual person for identification. It predicts joint location, limb direction and orientation using three branch CNN architecture with part affinity field keeping the initial features. They have opensourced their work as a project called OpenPose [15].

Wearable device-based approaches have also been used in this pose estimation task involving yoga postures. They require attaching sensors to each joint of the human body during exercise. Wu et al. proposed a pose recognition and quantitative evaluation approach [5]. A device which can be worn onto the human body is fixed with eleven inertial measurement units (IMUs) to measure yoga pose data. Then, the artificial neural network and fuzzy C-means are combined to classify the input pose into a category. Furthermore, the angular differences are calculated between nonstandard parts and the standard pose model (correct/expected pose) to help yoga learners better. Puranik et al. also proposed a wearable system [6] with a wrist subsystem to monitor pose with the help of a flex sensor, and a waist subsystem is also built to monitor the pose along with the use of a flex sensor. These approaches are however not very practical for everyday purposes.

Pose Trainer uses pose estimation method to detect the exercise pose and provides useful feedback for the user. The model uses an recording a dataset of over 100 exercise videos of correct and incorrect form, based on commonly followed training guidelines, and build geometric and machine learning method for evaluation. However, This method fails to analyze the performance of multiple persons appearing in the frame, and does not count the repetitions of the performed exercise. The selection of exercises is also limited to just four exercises; bicep curl, front raises, shrugs, and shoulder press. The camera perspective of the user recording is also limited making it difficult to allow free movement of users.

3. Method

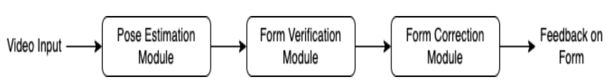

Figure 1. Form Corrector Pipeline

Our Form Correction Trainer Framework consists of a pose estimation module, form verification module and finally the form correction module. The pipeline of different stages is represented in Figure 1. First, the user selects the activity they are going to perform and then we get our input video either from a webcam or a prerecorded video and then we pass it onto our Pose estimation module. Our Pose Estimation Module uses MediaPipe [13] framework for estimating keypoints and pose in our video input. Then we pass the estimated keypoints on to the form verification module which verifies the form of the user performing our physical activity or exercise. It measures the angles between different keypoints associated various body parts used in each exercise and then compares it with the required correct form for the exercises and labels it as correct or incorrect form. Then, for the final stage of the our framework, we have the form correction module which will suggest the appropriate changes to be made for the incorrect form.

3.1. Pose Estimation Module

The user can use a live webcam feed or record themselves performing the exercise. The user selects the activity they are going to perform and performs the selected exercise. There are no specific requirements for the camera position, just that the user needs to be visible when performing the movement. The action can be performed in whichever angle they prefer and our model will correctly estimate the pose and suggest changes in form if required.

For our Pose Detection, we use a Machine Learning Framework called MediaPipe [13] which uses BlazePose [14] for skeleton keypoint extractions. It infers 33 3D landmarks (see Figure 3) and background segmentation mask on a single frame with 17 landmarks across the torso, arms, legs, and face. The ML pipeline initially locates the person/pose region-of-interest (ROI) within the frame using a detector. Using the ROI-cropped frame as input, the tracker then predicts the posture landmarks and segmentation mask within the ROI. The ML model has been trained and evaluated on the COCO dataset from the COCO Keypoint Detection Task. More than 200,000 images and 250,000 person instances tagged with key points make up the COCO dataset, which is divided into train, validation, and test sets.

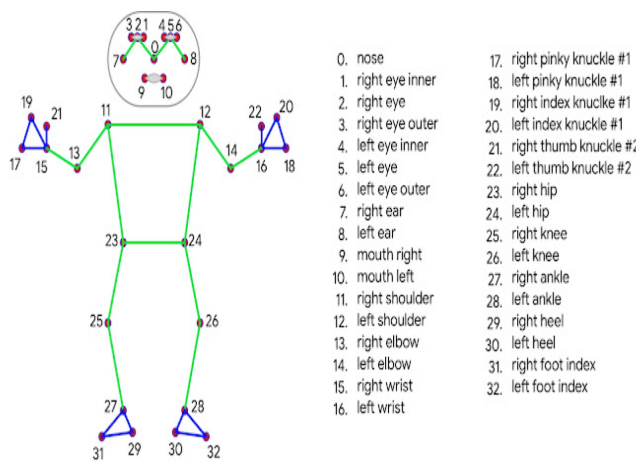

Figure 2. BlazePose [14] Keypoint Topology

3.2. Form Verification Module

In this module, to verify the correct form of posture, we use two different approaches to calculate it.

3.2.1 Angle Evaluation

We calculate the angle between relevant keypoints depending on the exercise being performed in this method. For example, if we take bicep curl as our sample exercise, we first collect references for the correct form when performing the exercise through multiple online exercise resources.

Next we performed few experiments and recorded their observations. This helps us in determine angle thresholds of joints for specific exercises. We noticed the following relationships between different keypoints and angles between them for bicep curl. We have 180°angle between the upper arm and forearm at the starting position. Then, the weight must be brought up for the bicep to contract, so that the angle between the upper arm and forearm should decrease and be below 90°and as the weight is lowered, the angle is increased again. We also need to calculate the angle between the shoulder and body which should be kept lesser than 35°to make sure that the user is not extending their arm too much. With the help of these angle thresholds, we then offer suggestion to the users to help improve their form.

3.2.2 Machine Learning Approach

Our second approach is more data driven, we are determining pose correction using a Machine Learning model. We collect videos performing different exercises and poses and use it as a ground truth value and assign labels to them based on their correctness. We are using the Kth neighbor classifier to calculate similarities between vectors calculated by applying Euclidean distance [7] to original pose keypoints. Euclidean distances are used to measure distances between two points in a Euclidean space. After pose estimation, the keypoints are extracted and then similarity is recorded with each other. After this we execute Kth nearest neighbor classifier to output pose classes such as left/right shoulder wrong, correct pose based on the calculated similarities.

3.3. Form Correction Module

In this module, after identifying the incorrect pose performed by the user, we suggest appropriate feedback for them. Since, different exercises have different technical aspects to them, our Form Correction module will suggest all the necessary improvements that needs to be made on the user pose. For users who are performing the exercise in real time with webcam input, the suggested changes will be shown on the upper left corner of the screen so that they can immediately review and correct their posture. We provide the angles on joints as the output to the user along with a color indicator turning from green to red if their form is incorrect. This way, they can identify which specific joint is being performed incorrectly and how much changes they need to make for the specific joint to get the right form. We also provide reps performed by the user so that they can accurately keep count of their exercises.

4. Experiments

For our first method, we use our webcam video as input to our model to perform angle evaluation. In case of the Data centric approach, due to scarcity of data available online, the authors have themselves collected the necessary videos performing exercises. We perform the exercises correctly, partially correct and completely incorrect and assign labels accordingly and store them. We use Precision, Recall, F1 Score as our evaluation metrics for our classification model.

4.1. Angle Evaluation

We consider two different exercises - bicep curl and lunges for our approach. Both involve working out a different group of muscles and used different keypoints making it perfect for testing our Form Correction Trainer.

4.1.1 Bicep Curl

We use the elbow, shoulder and hip joints as seen in figure 2 for calculating our shoulder angle. With regards to bicep curl, after extensive experimentation and referring online resources, we calculated this angle threshold to be greater than or equal to 35°for proper arm rotation. If the user maintains this threshold, we count that as correct form. We provide the following four pose classes for bicep curl.

1. Left Shoulder Incorrect
2. Right Shoulder Incorrect
3. Both Shoulder Incorrect
4. Correct Form

4.1.2 Lunges

In Lunges, incorrect form occurs when a person bends their knees too much. So, we use the hip, knee and ankle joints as seen in figure 2 for calculating our knee angle for Lunges. We calculated this angle threshold to be greater than or equal to 80°to avoid excess knee bending. If this threshold is maintained then the form is correct.

The output for this approach can be seen in figures 3 and 4. We see the correct and incorrect form detected by our Form Correction Trainer and the subsequent correction message along with the wrong joint displayed on the user's menu.

4.2. K-Nearest Neighbors

For our Machine Learning approach, we use K nearest neighbors algorithm for classifying our joint keypoint angles into appropriate pose classes. For bicep curl exercise, We collect our input data as left shoulder wrong, right shoulder wrong, both shoulders wrong and correct form and their corresponding angle values. We then pass our values into our model. We test our model by getting input angles from live webcam feed and then classify it into the respective classes and provide feedback to the user. Our model gives us an accuracy of 97 percent.

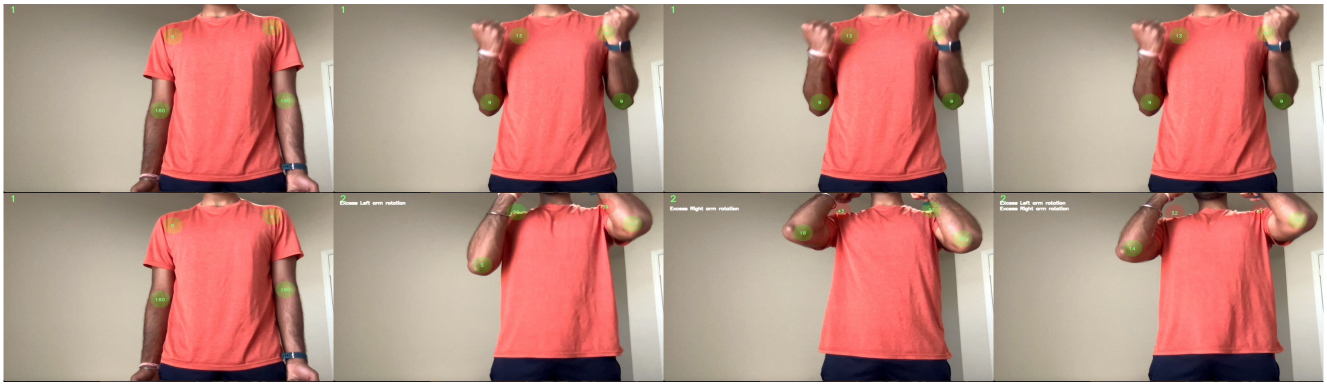

Figure 3. First sequence of images show the user correctly performing the bicep curl exercise. The second set of images show the incorrect form and the corresponding feedback message and red mask over incorrect joints.

Figure 4. First sequence of images show the user correctly performing the lunges exercise. The second set of images show the incorrect form and the corresponding feedback message and red mask over incorrect joints.

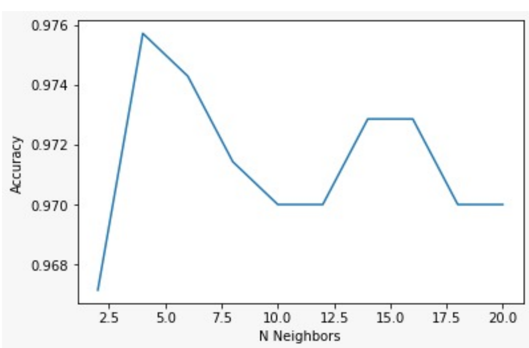

Figure 5. Choosing K value

Form Correction Trainer - Bicep Curl			
Class	Precision	Recall	F1-Score
Left Shoulder Incorrect	0.76	1.0	0.86
Right Shoulder Incorrect	0.81	1.0	0.90
Both Shoulder Incorrect	1.0	0.86	0.93
Correct Form	1.0	0.99	0.99

5. Conclusion

In this work, we have successfully implemented a Form Correction Trainer that accurately predicts whether a user is performing an activity right and provides precise feedback on their posture for improvement. We evaluate our model by collecting real world data and use two different approaches - Angle evaluation and KNN and two distinct exercises bicep curl and lunges for our experimentation. We get good results from both methods and are able to successfully predict incorrect posture and provide specific feedback. For our future work, we plan on implementing our Form Correction Framework on different exercises and physical activities such as yoga, body weight exercises and sports activities of varying complexities and counting calories. We plan on making our framework robust to external noises and be capable of detecting the correct exercise posture with different angles and with multiple people present in the video frame and work in different settings such as home, gym or an open physical space.

References

[1] G. Papandreou, T. Zhu, N. Kanazawa, A.Toshev, J.Tompson, C. Bregler, and K.Murphy. ”Towards accurate multi-person pose estimation in the wild”. In CVPR, 2017.

[2] X. Chu, W. Ouyang, H. Li and X. Wang, ”Structured Feature Learning for Pose Estimation,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4715-4723, doi:10.1109/CVPR.2016.510.

[3] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. ”Realtime multiperson 2d pose estimation using part affinity fields.” In CVPR, 2017.

[4] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-scale Image Recognition,” Published as a conference paper at ICLR, Cornel University Library, 2015.

[5] S. Wei, V. Ramakrishna, T. Kanade and Y.Sheikh, ”Convolutional Pose Machines,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 4724

[6] W. Rawat and Z. Wang. (2017). ”Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Computation.” 29. 1-98.10.1162/NECOa00990

[7] Abu Alfeilat, Haneen Arafat, et al. ”Effects of distance measure choice on k-nearest neighbor classifier performance: a review.” Big data 7.4 (2019): 221-248.

[8] Alatiah, Talal, and Chen Chen. ”Recognizing exercises and counting repetitions in real time.” arXiv preprint arXiv:2005.03194 (2020).

[9] 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.

[10] Chen, Steven, and Richard R. Yang. ”Pose Trainer: correcting exercise posture using pose estimation.” arXiv preprint arXiv:2006.11718 (2020).

[11] Andriluka, Mykhaylo, et al. ”2d human pose estimation: New benchmark and state of the art analysis.” Proceedings of the IEEE Conference on computer Vision and Pattern Recognition. 2014.

[12] Soomro, Khurram, Amir Roshan Zamir, and Mubarak Shah. ”UCF101: A dataset of 101 human actions classes from videos in the wild.” arXiv preprint arXiv:1212.0402 (2012).

[13] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg and Matthias Grundmann. ”MediaPipe: A Framework for Building Perception Pipelines” arXiv preprint arXiv:2006.10204v1 (2019).

[14] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang and Matthias Grundmann. ”BlazePose: On-device Real-time Body Pose tracking” arXiv preprint arXiv:2006.10204v1 (2020).

[15] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei and Yaser Sheikh. ”OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields” arXiv preprint arXiv:1812.08008 (2019).