Pose Estimation and Correcting Exercise Posture

Rahul Ravikant Kanase ^a rahulkanase2600@gmail.com

Akash Narayan Kumavat ^a kumavatakash001@gmail.com

Rohit Datta Sinalkar ^a rohit4sinalkar@gmail.com

Sakshi Somani ^a sssomanisakshi@gmail.com

Abstract: Our posture shows an impact on health both mentally and physically. Various methods have been proposed in order to detect different postures of a human being. Posture analysis also plays an essential role in the field of medicine such as finding out sleeping posture of a patient. Image processing based and sensor based approach are the leading posture analysis approaches. Sensor based approach is used by numerous models to focus on posture detection in which the person needs to wear some particular devices or sensors which is helpful in cases such as fall detection. Image processing based approach helps to analyze postures such as standing and sitting postures. Fitness exercises are exceptionally beneficial to individual health, but, they can also be ineffectual and quite possibly harmful if performed incorrectly. When someone does not use the proper posture, exercise mistakes occur. This proposed application utilizes pose estimation and detects the user's exercise posture and provides detailed, customized recommendations on how the user can improve their posture. A pose estimator called OpenPose is used in this application. OpenPose is a pre trained model composed of a multi-stage CNN to detect a user's posture. This application then evaluates the vector geometry of the pose through an exercise to provide helpful feedback. Pose estimation is a method in which spatial locations of key body joints is calculated using image or video of the person. This computer vision technique detects human posture in images or videos and shows the keypoints such as elbow or knee in the output image.

Keywords: CNN, OpenPose, pose estimation, posture detection.

I. INTRODUCTION

Various exercises such as dead lifts, squats and shoulder presses are favourable to human body fitness but it can also be very harmful if the exercises are performed improperly. Injuries to the muscles or ligaments can be caused due to heavy weights involved in these exercises. Due to the lack of training or knowledge many people do not follow the correct posture to be maintained while performing these exercises regularly. This may lead to muscle fatigue and muscle strain. In this course project, using the latest techniques in pose estimation we help people in performing exercises with correct posture by developing a project that detects the users pose while exercising and provides feedback and suggests improvements if necessary. The goal for this project is to prevent injuries and improve the form of human workout with just a computer and a camera. The initial step of the project is to use human pose estimation which is a highly applicable domain of computer vision. A trained model determines a person's joints as a list of skeletal key points from the given data which could be an RGB image or a depth map. Pose estimation plays an important role in solving problems related to human detection and activity recognition. It also helps in solving complex problems involving movement detection. OpenPose is used which utilizes neural network for the inference of this project. The latter part of this project involves detecting the quality of human pose for a given exercise. It is approached using heuristic based and machine learning models. The full application consists of two main components which can take a video of an exercise and provide feedback to the user.

^a Department of Electronics Engineering, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai,400 706,India

II. RELATED WORK

Various methods have been studied for pose estimation that evaluate human poses using sensors, videos and machine learning approach. A few of them are mentioned below. A. Toshev and C. Szegedy [7] used neural network for the first time to improve pose detection using regression on CNN for finding the location of body joints. A stacked hourglass neural network architecture was introduced by A. Newell, K. Yang, and J. Deng [4] which works on bottom up and top-down approach for finding pose predictions. J. Shotton, A. Fitzgibbon and others [6] use single depth maps for prediction of 3D positions of joints using object recognition. F. Bogo, A. Kanazawa and others [2] used single RGB images to predict 3D pose and 3D mesh shape. Research work on detecting multiple human pose in a single frame has also been focused and worked on. G. Papandreou,

T. Zhu and others [5] used a two-stage process for detecting multiple poses, first step includes identifying people and the second includes detecting of their key points. For analysis of physical movements, P. Zell, B. Wandt, and B. Rosenhahn [9] used a method in which body is represented as mass-spring system to find the forces and torques that travel through the joints of the body.

Based on sequential prediction framework, S. Wei, V. Ramakrishna and others [7] propose a different architecture which uses multiple convolutional networks to clarify joint estimates over sequential passes and design a cascaded CNN network to represent texture and spatial information with convolution layers, sequentially incorporating global context to refine part confidence maps from previous iterations. During training period, due to the problem of vanishing gradients intermediate supervisions are added at the end of each stage. For reducing the information loss of neighbouring joints, the size of the receptive field is increased. To solve the problem caused by increasing the depth of network, intermediate supervision is used. This method also works against part occlusions. Each layer of a multi stage CNN becomes independent after training. These methods have high performances as compared to single CNN methods. For detection and regression, X. Chu, H. Li and others [13] use CNN's generalization. This model estimated human poses through combined detection and regression results by using shared CNN as the common input of detection and regression sub network. To identify multiple

people in a 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 [9] without the need to detect individual person. It consists of three branch CNN architecture which predicts joint location, limb direction and orientation with part affinity field keeping the initial features. This method has improved accuracy of regression because it combines and enhances the output of three branches. For this application a simpler approach is used which includes the analysis of the angles of distances between joint key points to provide feedback to the user without a full physical simulation.

III. SYSTEM DESIGN

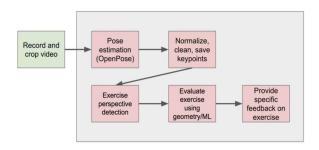


Fig 1. Block Diagram of Pose Estimation and Correction.

For the first step of the project, the users need to record a video performing a particular exercise with a certain point of view (front side, back side, etc.) which allows the exercise to be seen properly. There are no restrictions on the distance from the camera or the type of camera, the only thing that the user needs to take care is that their posture is visible properly. Further, the user should take care that the recorded video only contains the frames of exercise. The editing part of the video can be done using any software. OpenPose supports all common video formats that are available in modern computers and smartphones. Facial recognition includes facial key point detection which contains its own identifier and coordinates related to the image which is usually provided in pixels. Classification and localization are the important tasks in the training of key point detection in computer vision models. The same method is applied to the human posture for pose detection. It works in a much similar way which includes classification of body points, typically joints, determining the location within the video or image. Pose estimation methods are so highly developed that many calculate how we can pixels

the movement of the hand and if the height of the human body is known, it can be converted into distance. Velocity can be calculated with help of the frame rate of the video. Considering these measurements, we can say if the 5 body is in ascending or descending position. It also helps in classifying the activity or state of the human posture and analyzing what a person is doing and how they are doing it. OpenPose provides the output of key points in a frame-by-frame text in JSON format. Python uses pandas data frames to read the JSON files. Calculations are performed on key points provided with the help of Python. For providing feedback to the users, we generate video with the help of Open CV, the raw coordinates from OpenPose and our analyzed points.

IV. SYSTEM IMPLEMENTATION

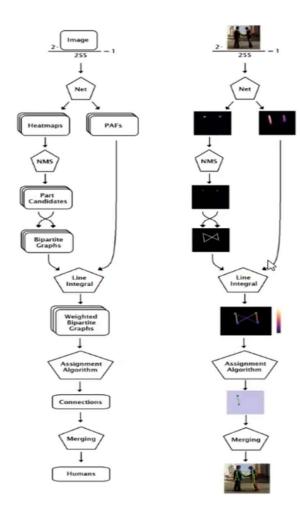


Fig. 2 Flowchart of Pose Estimation and Correction.

A. Pose Estimation

CNN is used for labeling RGB images in pose estimation. OpenPose software is used for posture detection which provides output in part affinity fields which shows the position and orientation of limbs. It has a high accuracy and efficiency as compared to other softwares. OpenPose is scalable and is easy for installation. It provides prediction of 18 key points including nose, neck, shoulders, wrists, elbows, hips, knees, and ankles.

B. Key point Normalization

Key point normalization includes list of predictions of key point for every video and part objects that store confidence of each key point. For representing full skeletal prediction of a human pose, joint key point is created to build a pose object of every frame in the video. A pose sequence object is created that combines each video frame for the entire video. The system is generalized considering different body length measurements, camera quality, distance from the camera and other factors. To compensate with these differences, the pose is detected based on torso length in pixels. The average of distance from neck key point to left and right hip key point is called as the torso length. It remains constant throughout the frames in the video.

C. Perspective Detection

A particular exercise has to be recorded from a particular camera perspective. For example, for front raise, the video has to be recorded from the side of the body and could be performed with left or right arm. With the help of the most visible key points, the system determines which arm is used.

D. Geometry Evaluation

In geometric evaluation, body vectors are calculated with the help of key points.

E. Example: Bicep Curl

In bicep curl, the angle between upper arm and torso vector is calculated which denotes the movement of the upper arm. If the upper arm does not move significantly, then it should be parallel to torso for the entire frames of the video. A large change in the angle between these vectors will indicate that there is a movement in the upper arm. The dumbbell should be brought up completely so that the bicep is fully contracted beyond 90° between upper arm and forearm. A minimum threshold is set for the angle between upper arm and forearm. It will point out for incorrect form due to heavy weight used etc. At the start, the angle between the upper arm and

forearm is 180° which decreases when we the bicep is contracted upwards and again increases when the dumbbell is brought downwards. The minimum angle will be found at which the weight is lifted for the complete video frames. From the available data, a feedback will be suggested on the posture of the user regarding the rotation of the arm or the weight of the dumbbell. Making use of shoulder to help lifting the weight upwards denotes incorrect form of exercise which results in excess rotation of shoulder. (Refer Fig. 3 and 4)

F. Example: Front Raise

In front raise, the user should lift the weights slightly above the shoulders and avoid using torso movement for lifting the weights. To identify the form the posture during this exercise, two things are identified: Maximum angle between the arm and torso and the horizontal range of motion of the back. To detect the swinging in the upper body between the frames of exercise, the change in vector difference is calculated for monitoring back motion. (Refer Fig.5 and 6)

G. Example: Shoulder Shrug

In shoulder shrug, the motion of shoulders should be in the full range and the elbows should not be bent. Also using arms for lifting indicates incorrect form. It is calculated by the angle between upper arm and forearm, torso length indicates the range of motion of the shoulders. Low shoulder movement indicates incomplete motion of shoulders and the small angle denote bending of elbows. (Refer Fig. 7 and 8)

H. Example: Shoulder Press

In shoulder press, the weights should not be lifted far forward or backward. Moving the torso too much will indicate heavy lifting or improper lifting. The torso should remain constant and vertical throughout the exercise. The shoulder press movement is observed using the motion of the back which is calculated by the neck and hip key points. It also depends on the motion of arm calculated by elbow and neck key points and the maximum angle between forearm and upper arm vector. The feedback is provided to keep the back straight if large motion is observed for the back. During lifting of dumbbells, rolling of shoulders is observed if the elbow locates behind the neck indicating incorrect form of the exercise. If the angle between forearm and upper arm is small, a feedback is given suggesting to improve the range of motion while lifting the weight. (Refer Fig.9 and 10)

I. Machine Learning Evaluation

This approach includes detection of pose in a data driven format. As the recorded video can be of any length, it creates an error in vector length for each example. To solve this error, dynamic time warping (DTW) with the nearest neighbour classifier is used. DTW is used to measure nonlinear similarity between two time series. Euclidean distance fails because sequences are phase shifted in time. In DTW, we dynamically calculate key points to a given point in the next sequence that correlates with the first. A distance matrix is calculated where the matrix element represents the Euclidean distance between two key point sequence. The most ideal match of points is found and the distance between them is determined. The limitation of DTW is that its not strong enough against noise. To avoid this error in DTW measures, a key point sequence is passed two times through a five-median filter. For detecting correct or incorrect form based on the calculated DTW distances, a nearest neighbour classifier is developed.

V. RESULTS

Videos for different exercises are uploaded on this system. The results for four exercises are shown below:

A. Bicep Curl

1. Input:



Fig.3 Input for Bicep curl

2. Output feedback for bicep curl:

Bicep curl
Output on command prompt:

Exercise arm detected as: left.
Upper arm and torso angle range: 17.36865202142973
Upper arm and forearm minimum angle: 44.669291603102366
Exercise performed correctly!

Exercise performed correctly! Weight was lifted fully up, and upper arm did not move significantly.

Fig.4 Output for bicep curl

B. Front Raise

1. Input:



Fig.5 Input for Front Raise

2. Output feedback for front raise:

Front Raise Output on command prompt:

Exercise arm detected as: right.
Horizontal range of motion for back: 0.2816274951491309
Max angle between torso and arm when lifting: 99.16085343152268
Exercise performed correctly!
Exercise performed correctly! Weight was lifted fully up,
and no significant back movement was detected.

Fig.6 Output for Front Raise

C. Shoulder Shrug

1. Input:



Fig.7 Input for Shoulder Shrug

2. Output feedback for shoulder shrug:

Shoulder Shrug
Output on command prompt:

and arms remained straight.

Range of motion for shoulders: 0.25897065694603627

Max upper arm and forearm angle: 5.201748222054894

Exercise performed correctly!

Exercise performed correctly! Shoulders went through range of motion,

Fig.8 Output for Shoulder Shrug

D. Shoulder Press

1. Input:



Fig.9 Input for Shoulder Press

2. Output feedback for shoulder press:

Shoulder Press Output on command prompt:

Exercise arm detected as: right.

Range of motion for back: 0.2658138515490096

Minimum distance between elbow and neck: -0.26585470576202264

Max upper arm and forearm angle: 175.1610366573757

Exercise could be improved:

You are rolling your shoulders when you lift the weights. Try to steady your shoulders and keep them parallel. You are not lifting the weight all the way up. Extend your arms through the full range of motion. Lower the weight if necessary.

Fig.10 Output for Shoulder Press

VI. CONCLUSION

In this project, an application is presented which provides feedback on human posture while performing exercises using pose detection, visual geometry and machine learning. The output of pose estimation is used to calculate human body key points from the video provided. Machine learning algorithm is used for deciding posture correctness and geometric algorithms for providing feedback on exercise improvement. Four exercises have been considered which can be extended to many other exercises for future work. For getting a feedback at any place, a mobile application can be developed that allows the user to record a video from the smartphone an upload on the application. A web page can be developed to display the output provided by the project. Feedback improvement can be increased suggesting targeted action and specific suggestions regarding the body part used and the weight of the equipments. The exercise performed with correct form can be represented using graphical simulation to highlight the mistake of the user and how to improve it.

VII. REFERENCES

- D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In AAAIWS, 1994.
- F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In ECCV, 2016.
- 3) Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multiperson 2d pose estimation using part affinity fields. In CVPR, 2017.
- A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016
- 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.
- 6) J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In CVPR, 2011
- 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-

- 8) W. Rawat and Z. Wang. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Computation. 29. 1-98. 10.1162/NECOa00990
- J 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.
- 10) J. Tompson, A. Jain et al. "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation." Eprint Arxiv(2014):1799-1807.
- 11) X. Chen, and A. Yuille. "Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations." Eprint Arxiv(2014):1736-1744.
- 12) 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.
- 13) 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.
- 14) J W. Yang, W. Ouyang, H. Li and X. Wang, "End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3073-3082, doi: 10.1109/CVPR.2016.335.
- 15) Steven Chen & Richard Yang (2018). Pose Trainer: Correcting Exercise Posture using Pose Estimation.