Climate Change Parameter Dataset (CCPD): A Benchmark Dataset for Climate Change parameters in Jammu and Kashmir

Tajamul Ashraf[1[0000-0002-7372-3782]], Balaji Prabu BV[2[]], and Omkar Subbaram Jois Narasipura[3[0000-0002-0806-8339]]

[1] Indian Institute of Technology Delhi
tajamul@sit.iitd.ac.in
[2] Malnad College of Enginnering, Hassan
[3] Indian Institute of Science , Bangalore, 560012, India

Abstract. Human pose estimation is the process of continuously monitoring a person's action and movement to track and monitor the activity of a person or an object. Human pose estimation is usually done by capturing the key points which describe the pose of a person. A guiding practicing framework that enables people to learn and exercise activities like yoga, fitness, dancing, etc., might be built using human posture recognition remotely and accurately without the help of a personal trainer. This work has proposed a framework to detect and recognize various yoga and exercise poses to help the individual practice the same correctly. A popular Blaze-pose model extracts key points from the student end and compares the same with the instructor pose. The extracted key points are fed to the Human Pose Juxtaposition model (HPJT) to compare the student pose with the instructor. The model will assess the correctness of the pose by comparing the extracted key points and give feedback to students if any corrections need to be made. The proposed model is trained with 40+ yoga and exercise poses, and evaluated the model's performance with the mAP, and the model achieved an accuracy of 99.04%. The results proved that any person could use the proposed framework in real-time to practice exercise, yoga, dance, etc. At their respective location without the help of a physical instructor with precision and accuracy, leading to a healthy life.

Keywords: Human Pose Estimation, Deep learning, client server model

1 Introduction

A long-standing issue in computer vision, human posture estimate has presented several difficulties. Pose estimation anticipates and extracts the key-point position of a person or item to follow the activities taken by the individual while executing any activity to draw insights. Many industries, including video surveillance, biometrics, assisted living, at-home health monitoring, helping yoga and exercise, etc., can benefit from human activity analysis. In today's fast-paced

society, people choose to exercise at home by copying the teacher through online video chats or YouTube videos. Humans are prone to making mistakes while mimicking the virtual world when there is no one to guide them. Therefore they feel the need for an instructor to grade their activities. A self-learning system can be created using human posture recognition. [18].
Due to the complexity of the human body, pose estimation has been a challenging issue and a topic of current research [4]. Although not all motions between joints are visible, the human body contains 244 degrees of freedom and 230 joints.

20 degrees of freedom and 10 big components, according to [6]. Deep learning has tremendously aided human posture estimation recently, and enormous performance improvements have been made. [13]. This paper presents a novel and state-of-the-art Human Pose Tracking and Juxtaposition(HPTJ) Model to aid students in practicing the exercise properly in a remote place with proper guidance. It is the first of its kind where we propose an Instructor Based Model, which gives proper instructions to the user through feedback at each step of the exercise to practice it properly. The developed model lays the foundation for building a system that can recognize human poses on both pre-recorded videos and live to stream. The model can autonomously teach proper workout regimes, sports techniques, and dance activities. Also, it can be used to understand full-body sign language, motion Capture and augmented reality, and Training Robots. The present work considers Yoga postures for implementing the model, which could recognize and assist the users in practicing all yoga postures correctly by following a pre- recorded video or a live class.
The rest of the paper is organized as follows. Section 2 highlights the previous works done on pose estimation. Section 3 discusses the proposed model and implementation for the same. Section 4 discusses the results obtained and the evaluation of the proposed model. Future research directions are provided in Section 5 of the paper.

2 Related Work

Many research employs a variety of methods, including posture estimation using the thiazepine model and posture identification through machine learning and deep learning techniques. Human position detection plays a significant part in these investigations. [9].

Prior until now, numerous people's poses were classified using machine learning for real-time posture recognition utilizing posture estimation using a 3D posture from the camera. [17]. In those models, basically, we compared the poses of two people in general, it can be an instructor and a student, captured by a webcam camera. Several businesses have created a range of tech related goods for sports and fitness using this methodology. For instance, the Company offered the NADI X-Smart Yoga Pants as a wearable product (which could guide exercise form via a mobile application) [11], Smart Mat developed an intelligent yoga mat with a sophisticated sensor to detect the pressure node of stance on the mat and deliver real-time feedback on how the user has executed the yoga posture, i.e., properly or wrongly, via a mobile application [5]. Yoga Notch came up with a yoga wearable device guide, placed on the body, which provides audio feedback on alignments when the user imitates instruction at home [20]. We tend to observe that all the existing systems use a sensor-based system which is not only expensive but hard to realize in real-life scenarios with sensors all around our bodies [19]. Since the invention of the deep pose, approaches based on deep learning have replaced the classic skeletonization approach. [2]. It directly applies deep neural network-based regressors on joint coordinates. It predicts a person's behavior and the location of concealed bodily components. Using OpenPose and long short-term memory networks, researchers have devised a way to detect human behaviors in real-time. [14]. This method is based on screenshots that are timed and taken from real-time photographs that are obtained by attaching the camera.. OpenPose is a real-time, open-source project that aims to jointly identify the human body with hands, faces, and legs on a single picture [12]. The output of body features is split into sub-sequence called windows using a sliding window approach. Whereas Long Short- term Memory is an RNN (Recurrent Neural Network) [10] different from feed-forward Neural Networks. It can handle both the sequence of data and a single data point thanks to its feedback links. Long Short Term Memory (LSTM) model is suitable for this scenario and provides good results [16]. It efficiently learns the key point features and returns an activity class. Although OpenPose, LTSM methodology became quite revolutionary, it was also filled with many flaws. A new topology employing 33 points on the human body was created by taking the superset of the points utilised by Blaze Face, Blaze Palm, and Coco in order to fix the flaw in the prior technique. We are able to maintain consistency with the relevant datasets and inference networks as a result. As opposed to OpenPose and Kinect [15] topologies, We estimate the rotation, size, and position of the region of interest for the subsequent model using just a few critical points on the hands, feet, and face. [8]. Blaze pose estimation is a real-time multi-person system presented to detect a human body, hand, facial, and foot key points on single images [1]. Convolutional neural networks (CNN)-based architecture is used to provide the positions of the human body's joints, which represents a significant advancement in the field of pose recognition. The goal of this project was to create a coaching system for human posture detection based on transfer learning. In this methodology, eight joint angles are calculated, and the students; joint angles are compared with the instructors. This method removed the deficiency of sensor-based pose identification as well as improved the time lag faced in open pose estimation methods as well as LTSM [3]. For comparison, an earlier segment was constructed across the two images, and through histogram, a comparison was done, which was prone to many errors as image normalization was very difficult to obtain in that case [7]. To improve these shortcomings, In this project, we used angle comparison techniques between different points and transferred calculated angles between two servers that easily detect errors with greater accuracy. We have also added commands which constantly tell users if they are making any mistakes. What lagged in the previous model was real-time comparison and giving feedback at the same time. Each segment will be extremely quick, taking only a few milliseconds per frame, thanks to the real-time speed of a comprehensive ML pipeline that integrates positioning and tracking models. The proposed system can recognize a complete human pose in real-time and from pre- recorded videos as well while performing real-time comparisons on poses concerning an instructors pose to enable an appropriate mimicking mechanism.

3 Methodology

This section discusses model development, validation, and implementation of the proposed model. The architecture of the proposed model is shown in figure 1 and is divided into three main stages namely 1) Pose Extraction 2) Real-time angle transfer and Comparison 3) Evaluation metrics and user feedback. In the pose estimation phase, input from both users and instructors is considered as frames and each frame is processed using the BlazePose model to extract the key-points. The extracted key points from both the user and instructor are used as input in phase 2 and the angles between them are computed to check the similarity between them. The final stage computes the error and accuracy of the user's pose and appropriate feedback is given to the user to correct the pose on par with the instructors.

3.1 Pose Extraction

The Blazepose model is used to extract the crucial details that aid in pose detection. A Blaze posture is a real-time pose recognition method that can identify human poses in an image or video by identifying various body components, including elbows, hips, wrists, knees, and ankles. By connecting these points, the method creates a pose's skeletal structure. The user's stream is recorded using a camera and fed into the BlazePose model in order to extract the critical points. As illustrated in figure 2, the BlazePose model estimates the person's posture using 33 important points, including the person's ears, eyes, nose, neck, shoulders, buttocks, knees, ankles, elbows, and wrists. To extract the human posture, BlazePose employs two machine learning models, a Detector and an Estimator. The Estimator calculates the important locations while the detector determines the human area from the supplied image. The detector runs on each frame when the frame is supplied to the Blazepose model, locating the region of interest (ROI) to detect a person. The tracker is used to monitor the individual in consecutive frames once they have been spotted, and it predicts all 33 posture key-points from this ROI. When it comes to movies, the detector is only performed on the first frame and the ROI is derived from the key-points of the preceding frame. The pipeline's pose estimation component forecasts the locations of each of the 33 essential points with three degrees of freedom (x, y location, and visibility). The detector may be used in either box mode or alignment mode. Box mode uses its location (x, y) and size to establish the bounding box (w, h). While in alignment mode, the bounding box with rotation may be

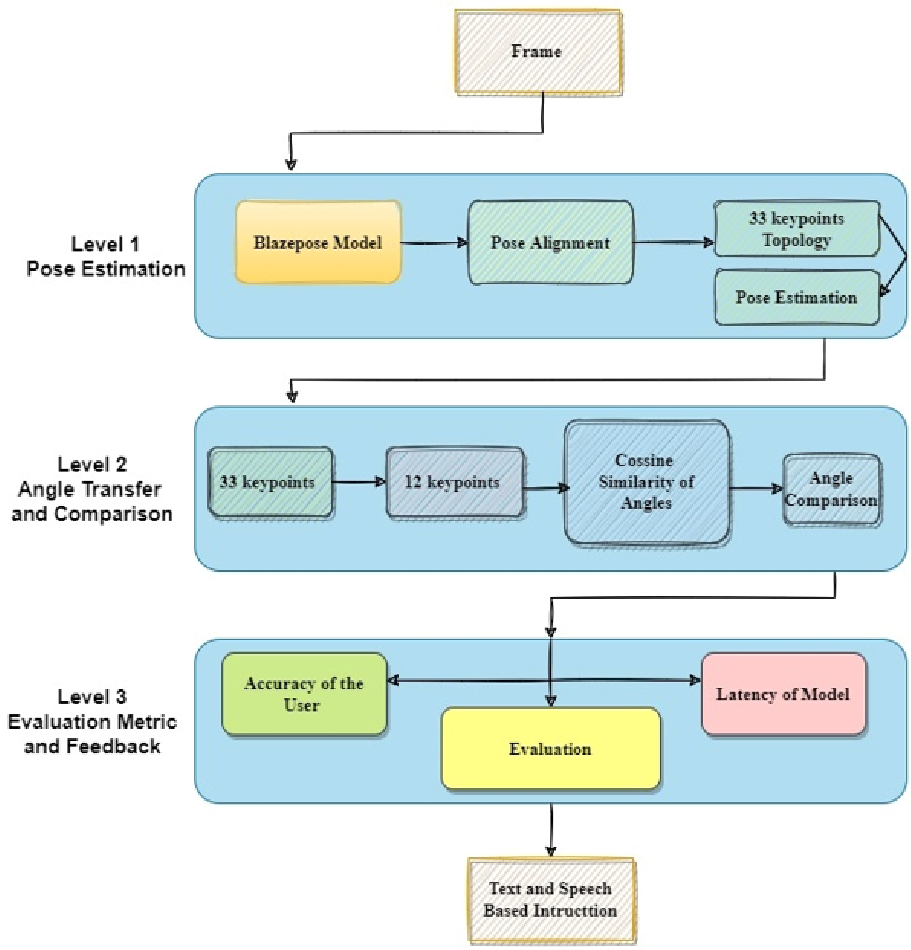

Fig. 1. Model Structure

anticipated and the scale and angle are derived from (kp1x, kp1y) and (kp2x, kp2y). The suggested model detects the human area in box mode. The model's first output is a set of landmarks, which total 165 components and have (x, y, z, visibility, and presence) for every 33 key-points. A sigmoid function is used to transform the visibility and presence from key point values in the [min to max] range to probability. The visibility function returns the likelihood that any key points are present in the frame and unobstructed by other objects. The likelihood of important points existing in the frame is returned by presence. The attribute of visibility describes whether a key-point is visible, not visible, or not labelled.

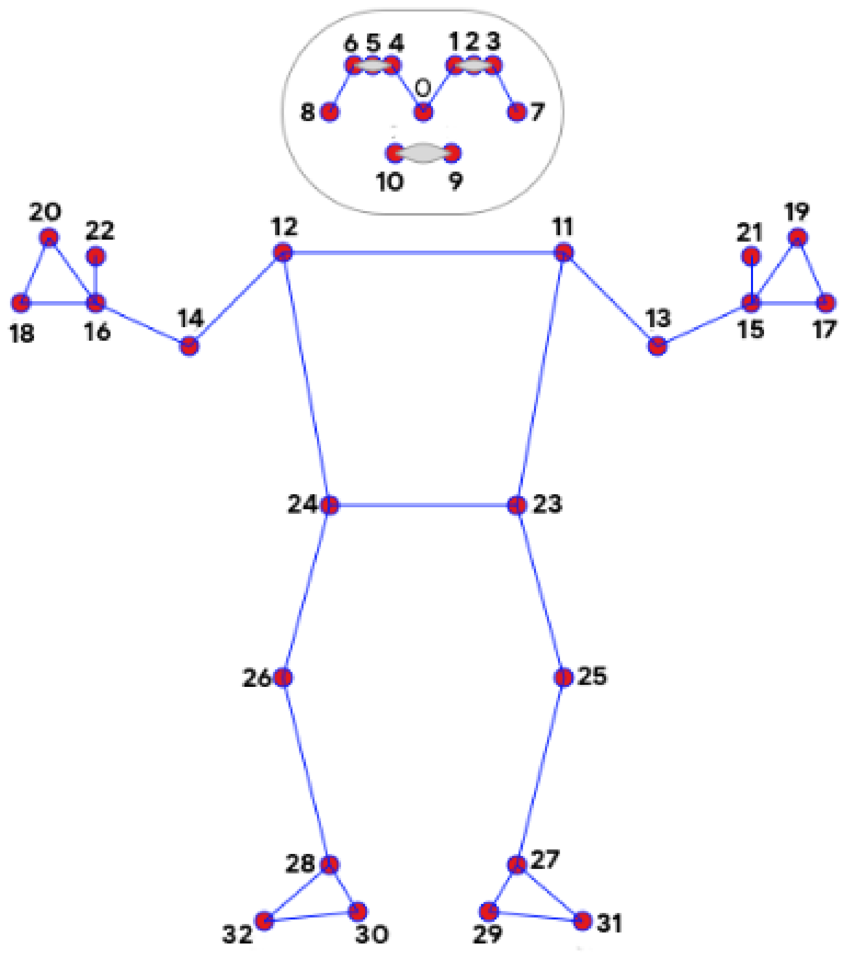

Fig. 2. BlazPose Keypoint Model

3.2 Real-time angle transfer and comparison

Once the key points are identified, the next step is to compute the angles between them. As mentioned earlier, BlazePose uses 33 key points for human pose estimation but all 33 key points are not necessary for identifying a human pose. Based on the type of posture interested to detect, selected key points could be used. The proposed model uses 12 key points namely: 16, 14, 12, 24, 26, 28, 15, 13, 11, 23, 25 and 27. The x and y coordinates from those points are used to calculate the angle using the 2-argument arc tangent function as shown in equation 1, which returns a value between -180 degrees and 180 degrees. After this step, the radian is calculated and finally, it is converted to an angle using NumPy's abs [27].

$$\theta = \tan^{-1}\left(\frac{Perpendicular}{Base}\right) \tag{1}$$

Here, θ is the angle between the hypotenuse and the base of a right-angled triangle.

If the measured angle was greater than 180, then the new angle was obtained by subtracting the obtained angle by 360 as the input to the function is defined from -180 to 180. Likewise using 12 points, 8 angles were determined. Once the angles for student and instructor frames are computed, they need to be transferred to the server for comparison. This work uses sockets for transferring the angles These angles are calculated simultaneously on the server and client side and then transmitted from the server to clients using socket links.

3.3 Evaluation metrics and User feedback

The error between the teachers' angles and students' angles can be stated as the performance of the student to the teacher. Per frame, accuracy was calculated for both real-time and recorded videos and the average mean of all the frames by total frames was taken as the overall accuracy of the student. To calculate the latency i.e time taken by the algorithm to detect the key points on both teacher's and students' sides and transferring through sockets and doing the comparison and instructing the students about his/her pose. To efficiently take the latency, timeit.timeit() library was used and per instruction, latency was calculated. At the end of the session, the average mean latency of all the instructions, and the no of times the key points were compared were taken to get the overall latency of the algorithm. The pose and model evaluation metrics were calculated on the client side. The pose accuracy was calculated to angles from the server script and the overall performance of the user was determined using object key-point similarity (OKS) based mAP. It is defined as

$$\sum_i \frac{(-d_i^2/2s^2k_i^2)\sigma(v_i > 0)}{\sum_i \sigma(v_i > 0)} \tag{2}$$

Where,
The euclidean distance between the projected keypoint and the ground reality

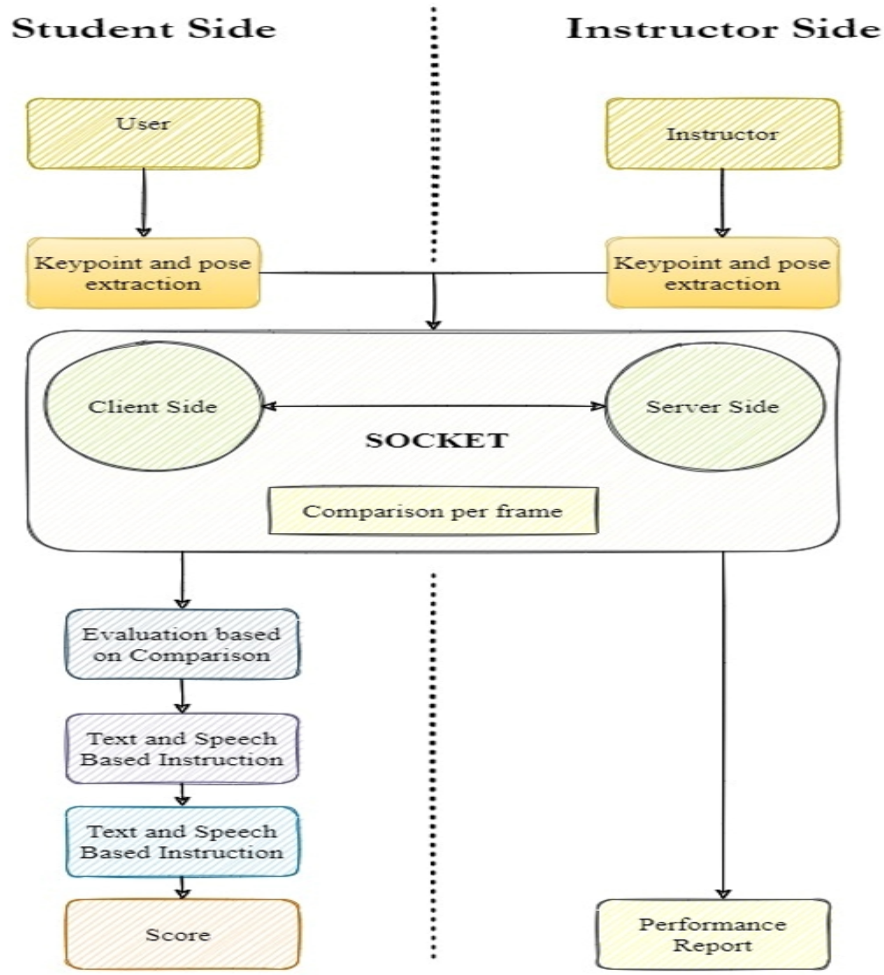

Fig. 3. Socket Architecture

is di.
S is the object segment area's square root.
The per-keypoint constant known as k regulates fall off.
Vi is regarded as a visibility flag that may be set to 0 for not labelled, 1 for labelled but not visible, or 2 for both labelled and visible.
OKS is used to calculate the distance (0-1), it shows how close a predicted keypoint is to the true keypoint. The pose error between the 8 corresponding calculated angles was determined and compared to certain predetermined threshold values which assisted in building an audio feedback mechanism for the user to correct their pose with respect to the instructors. The algorithm latency

was calculated between each error calculation which aided in finding the overall algorithm latency and efficiency.

4 Results

Input video is passed to the system as frames. First the frames are taken as input from student and teacher side. Detection model is performed on the teacher and student frames. From both these sides, the poses is extracted as shown in fig 4 the points are detected and send to the client side(teacher side).

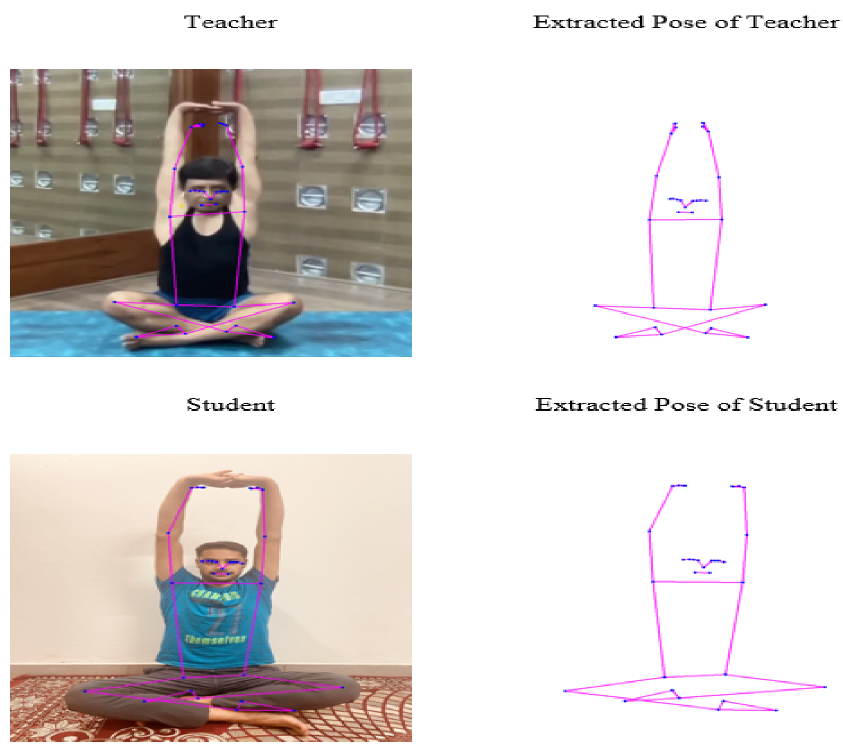

Fig. 4. Initial Input and output frame

It should be emphasised that our method may be used with input from a normal RGB camera, eliminating the requirement for Kinect or any other specialist gear for Yoga position detection. After the pose is detected, the comparison is calculated and the instrcutions are sent to the student using text and speech, so as to correct the pose. With each frame a comparison score is calculated, which is used to evaluate the student performance. Figure 5 shows the comparison accuracy for 100 frames.

The graph showing the per frame accuracy is shown in figure 5 The model

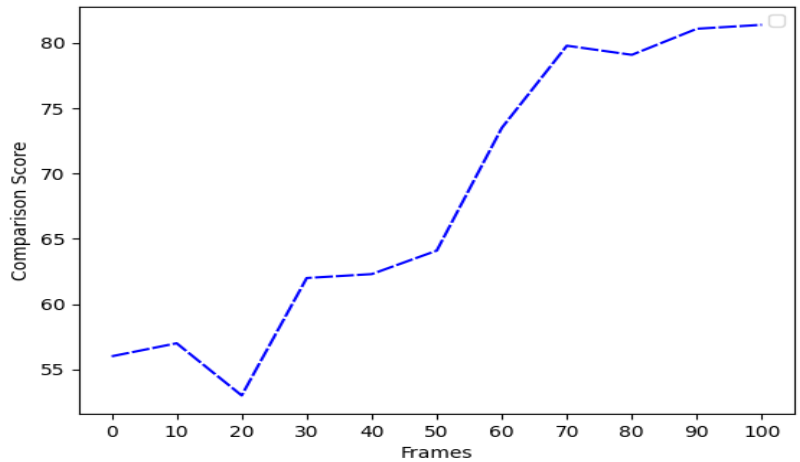

Fig. 5. Frame wise accuracy of the client side wrt to teacheer side

was able to use the information gained from client and server side at each frame in order to compare the actions performed by two people.The results are shown in figure 6

Additionally, the system may be put into use on a portable device for self-training and real-time forecasts. For practical applications like yoga, this study serves as a demonstration of activity recognition systems. For posture recognition in a variety of applications, including surveillance, sports, healthcare, picture classification, etc., a similar method may be employed.

5 Conclusion

An integrated system for the localization of points and the recognition of human body positions, followed by a procedure for error identification, has been established in this study. With the use of this approach, people will be able to practice yoga correctly on their own and avoid injuries that could result from improper technique. The proposed model uses the student pose and compares the correctness with teachers through key point extraction. Appropriate feedback is given to students when the pose performing is not on par with the teacher. The proposed model helps people to practice yoga at their places without the help of a physical instructor with precision and accuracy, leading to a healthy life. A mobile application with voice feedback could be developed in the future to help people to access the system easily. Also, the system could be extended to practice any exercise any time.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article

References

1. Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., Grundmann, M.: Blazepose: On-device real-time body pose tracking. arXiv preprint arXiv:2006.10204 (2020)
2. Chen, Y., Tian, Y., He, M.: Monocular human pose estimation: A survey of deep learning-based methods. Computer Vision and Image Understanding **192**, 102897 (2020)
3. Chong Loo, K.B.K., et al.: Detection of violent behavior in open environments using pose estimation and neural networks
4. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding **108**(1-2), 52–73 (2007)
5. Kasman, K., Moshnyaga, V.G.: New technique for posture identification in smart prayer mat. Electronics **6**(3), 61 (2017)
6. Koolstra, J.H.: Dynamics of the human masticatory system. Critical reviews in oral biology & medicine **13**(4), 366–376 (2002)
7. Koprinska, I., Carrato, S.: Temporal video segmentation: A survey. Signal processing: Image communication **16**(5), 477–500 (2001)
8. LaViola Jr, J.J., Feliz, D.A., Keefe, D.F., Zeleznik, R.C.: Hands-free multi-scale navigation in virtual environments. In: Proceedings of the 2001 symposium on Interactive 3D graphics. pp. 9–15 (2001)
9. Liu, Z., Zhu, J., Bu, J., Chen, C.: A survey of human pose estimation: the body parts parsing based methods. Journal of Visual Communication and Image Representation **32**, 10–19 (2015)
10. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech. vol. 2, pp. 1045–1048. Makuhari (2010)
11. Muhammad Sayem, A.S., Hon Teay, S., Shahariar, H., Luise Fink, P., Albarbar, A.: Review on smart electro-clothing systems (secss). Sensors **20**(3), 587 (2020)
12. Osokin, D.: Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. arXiv preprint arXiv:1811.12004 (2018)
13. Rafi, U., Leibe, B., Gall, J., Kostrikov, I.: An efficient convolutional network for human pose estimation. In: BMVC. vol. 1, p. 2 (2016)
14. Sawant, C.: Human activity recognition with openpose and long short-term memory on real time images. EasyChair Preprint (2297) (2020)
15. Smisek, J., Jancosek, M., Pajdla, T.: 3d with kinect. In: Consumer depth cameras for computer vision, pp. 3–25. Springer (2013)
16. Staudemeyer, R.C., Morris, E.R.: Understanding lstm–a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586 (2019)
17. Takahashi, K., Sakaguchi, T., Ohya, J.: Remarks on a real-time 3d human body posture estimation method using trinocular images. In: Proceedings 15th International Conference on Pattern Recognition. ICPR-2000. vol. 4, pp. 693–697. IEEE (2000)

18. Tawar, R., Jagtap, S., Hirve, D., Gundgal, T., Kale, N.: Real-time yoga pose detection
19. Xu, T., Zhou, Y., Zhu, J.: New advances and challenges of fall detection systems: A survey. Applied Sciences **8**(3), 418 (2018)
20. Yoshida, K., Murao, K.: Load position estimation method for wearable devices based on difference in pulse wave arrival time. Sensors **22**(3), 1090 (2022)

Fig. 6. Client and server side keypoint comparison