Department of Electrical Engineering,IIT Jodhpur
BTP Report
January 2020

Title: **POSE TRAINER**

Name & Roll Numbers: **1. Anshika Gupta-B18EE004**
2.Harit Gedia- B18EE017

Name of Supervisor: **1. Dr. Himanshu Kumar**

ABSTRACT:

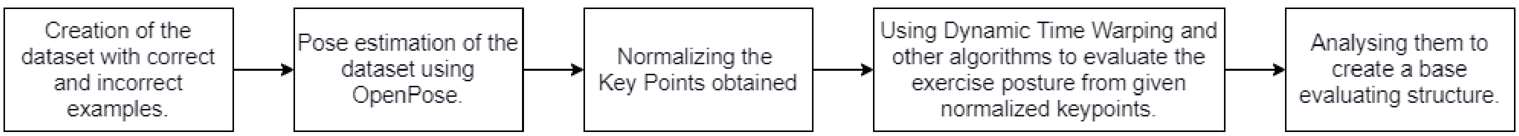
The importance of fitness and health is well known in today's era, and also we have seen the number of diseases it can cause if neglected. People have to spend a considerable amount of money on maintaining their fitness, but fitness can not be achieved without proper supervision and guidance. During this unprecedented time, it is almost impossible to get the desired guidance and if the exercise we perform is not done in proper form or pose, it can be dangerous. Thus, we have tried to bring a viable solution to it. In our project, we have developed the application that detects the error in your pose while performing a form of exercise, specifically yoga, by comparing it with the dataset taken into account to improve their form. We have collected the dataset for a single yoga pose specifically in the form of videos which has the collection of the correct and incorrect pose of that particular yoga.

MOTIVATION:

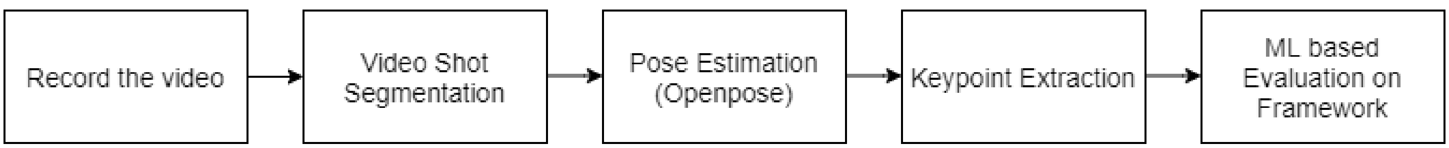
Fitness is crucial in one's life since it affects not only one's health but also once mind to the same level. The productivity level of a person increases if he is fit. Also, his efficiency in completing the task is directly proportional to how healthy he is. But everyone around us is unaware of the adverse effects it may cause to us if not performed in the right manner. Also, everyone does not find it so necessary to invest their money in hiring someone to guide them. In rural areas, there is a shortage of trainers or gyms for overseeing when they go wrong while performing the exercise, which is right according to their point of view but considered wrong if seen concerning the trainer.

In this modern era, everything around us is technology-based, and thus we chose the topic which reduces human interdependence in today's time and to make it simple for the people to correct their poses independently without the help of any trainer. This application will tell the user the mistakes he is making while performing the exercise so that he can look at the correct form of it and improve his way of doing it. This application will work on a real-time domain.

Flowchart:



Pipeline of Application:



METHODOLOGY:

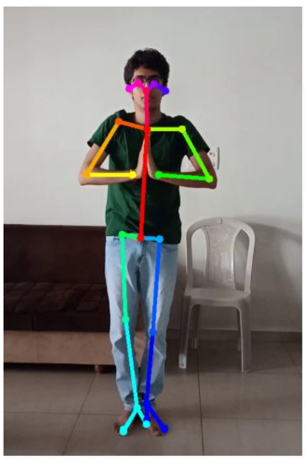
1. DATA SET:

In this, the user has to record its video from head to toe such that all the joints are visible while performing, whichsoever exercise is taken into account. All the joints must be visible. The recording must be done in front-facing cameras only. The distance does not matter as long as the whole body can be seen. The camera's quality is not specified as such, but the one with an unobstructed view is preferred for better results. Openpose supports all typical kinds of format so that any software can be used.

We have collected the dataset for a single yoga specifically, which is Pranamasana/Prayer Pose in the form of videos that have the collection of all the possible correct and incorrect forms. Videos are recorded from a 60 fps camera and a 30 fps with variant backgrounds. The dataset contains 5 reverence correct videos and 10 incorrect videos for each part of the body taken into consideration. To test the algorithm, further 15 videos are recorded. Variables in every video have been labeled ‘0’ for correct and ‘1’ for incorrect in a separate .csv file.

2. POSE ESTIMATION :

For this project, we have chosen the open-sourced pre-trained software, **OpenPose**, for pose detection. OpenPose is a real-time system that is used to jointly detect the human body parts and the key-points defining them(total 25 key-points, including the background). The model comprises a multi-stage CNN with two branches, one to learn the confidence mapping of a critical location on an image, and the other to determine the part affinity fields. It is considered both efficient, and one of the most accurate since it also scales up to multiple people without scaling up the run-time.

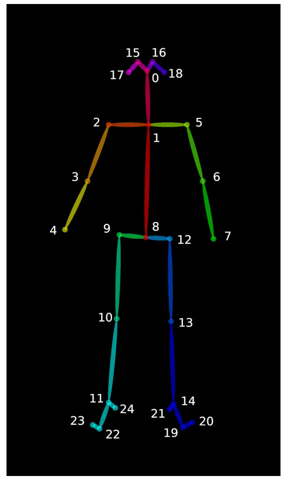

Correct Pose (Reference)


Incorrect Pose (Test)

3. KEYPOINT EXTRACTION:

OpenPose gives us raw keypoint output; hence we need to improve it, normalize, clean, and save that keypoint so that the points detected can be free from error as maximum as possible. First of all, we see whether there is more than one person in the inputs. So we begin with reading the **part_candidates** list of keypoint predictions corresponding to every video and segment that list and store its part objects in x,y, and confidence for each keypoint as shown in the figure.

Here, each number represents the [x,y,confidence] of its respective node in the given figure. (If it is empty, that means that no node is detected in the image)

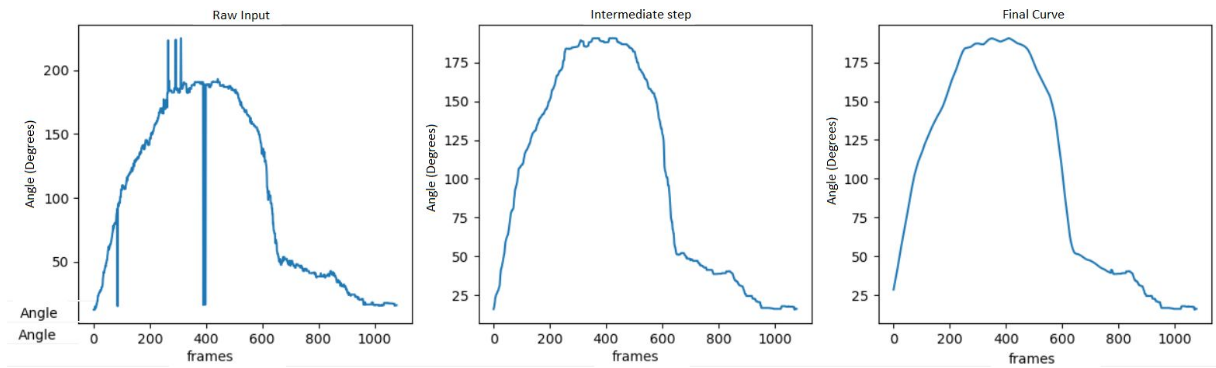


We take the key-points with the maximum confidence as our input. Then, we segment the key-points into their respective locations. If any keypoint is missing, we give it the value of zero. Then for every frame in the video, we compose each joint keypoint to construct a pose object that will represent the full skeletal estimation of a person in the frame. For the full video, we form a PoseSequence object that connects the Pose estimations from each video frame.

Since users are going to have different specifications of the camera, body measurements, distance from the camera, and other relative factors from the one which we have taken as standard, we realize that there is a need to generalize our application for users.

So, instead of taking the key-points for the evaluation, we convert them into relative terms. Here, we have taken the angle of the joints and the ratio of the parts connecting them. For example, instead of using keypoints of wrist, elbow, shoulders, and waist, we have calculated the angles between the torso and the forehand, and the forearm and upper hand. To estimate the plane of the hands, we also have taken their ratios into consideration. Similarly, we have taken the angles and the ratios of the lower body for evaluations. This prevents the irregularities caused by unequal camera angles and distances for each time.

4. NOISE/FLUCTUATION REMOVAL



We have noticed that key-points of OpenPose are not always accurate. There are certain noise fluctuations and irregularities in some frames that need to be removed for the smoother operation. Thus, we created an algorithm that specifically removes the fluctuations caused for short time durations. For that, we first identify the number of frames per second of the input and then, according to it, take the median of frames corresponding to the next 0.3 seconds and assign the value to the current frame. This removes the large fluctuations happening in small time intervals. To smoothen the curves further, we take the mean of frames corresponding to the next 0.3 seconds and assign the value to the current frame.

5. ML BASED ACCURACY EVALUATION:

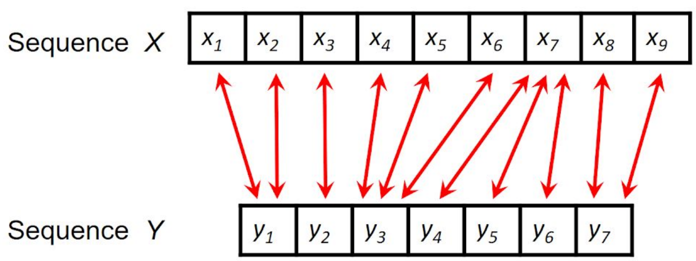
In this section we evaluate posture from the given normalized points using machine learning. For this we use dynamic time warping. The reason for choosing this approach is since the recorded videos can be of arbitrary length, this results in a different keypoint vector length which presents a challenge.

The DTW is a distance *metric* used to measure the nonlinear similarity between two input time series. The similarity or dissimilarity of a two-time series(i.e. shifted in the time dimension) is typically calculated by converting the data into vectors and calculating the Euclidean distance between those points in vector space. In DTW, we try to dynamically identify the keypoint in the second sequence that corresponds to a given point in the first.

For example, if two sequences are given then their respective DTW matrix is drawn where

$$D[i, j] = | x_i - y_j | + \min(D[i-1, j-1], D[i, j-1], D[i-1, j])$$

And a linear sequence from top corner to bottom corner including the minimum D[i,j] is chosen as given below. Their addition is called DTW distance.



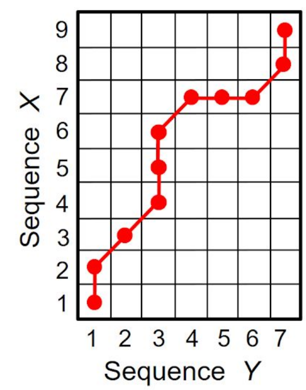


Figure 3.12 from [Müller, FMP, Springer 2015]

Thus in the above diagram, we get the adjusted sequence as
Sequence X - x1, x2, x3, x4, x5, x6, x7, x7, x7, x8, x9
Sequence Y - y1, y1, y2, y3, y3, y3, y4, y5, y6, y7, y7

A drawback with DTW is that it's not robust to noise. When OpenPose generates noisy keypoints, this would affect the performance of DTW. But as we have already filtered and cleaned the data, DTW becomes quite accurate.

Through Analysis, we have found a main variable which drives the whole Exercise. For example, the main variable in Pranamasana is the angle between torso and upper hand. Thus, we adjust the whole Input and reference data-frame using the matrix formed by Dynamic Time Warping between Torso-upper hand angles of both the data-frames. Then we find the Euclidean distance between each of the adjusted input variables and their respective adjusted reference variables. We specify the minimum criteria of the DTW distance to be termed as wrong posture.

Let us consider the Euclidean distances between the

$$C > \frac{d \times 100}{N} \quad - \quad (1)$$

Where c is our criteria constant.

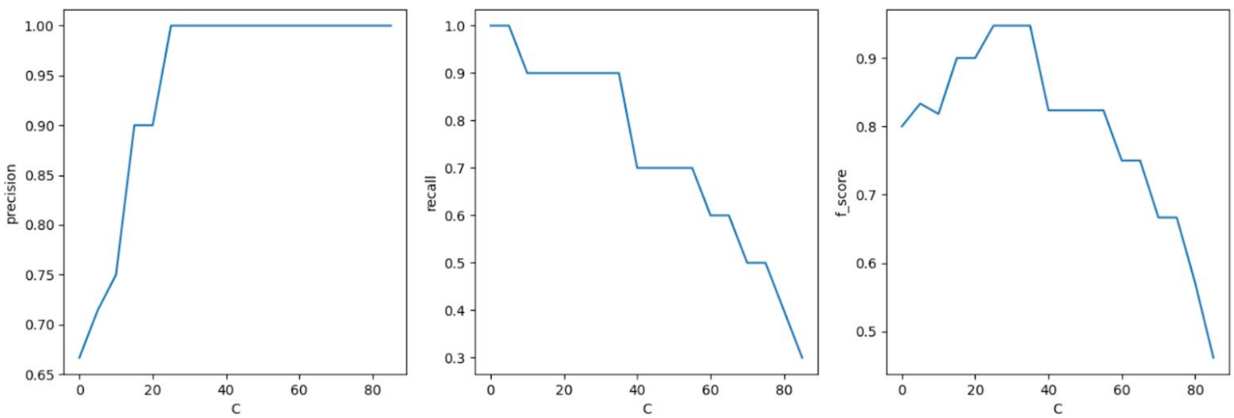
DTW distance is calculated for all the variables and if it is greater than their respective C, then the pose is not correct in consideration to that variable. To set the value of C, we create a training dataset containing 5 reference corrects poses and 10 incorrect poses. We measure the recall and precision of our dataset and calculate the f-score on the basis of them. We labeled each of the variables of the videos as '0' and '1', '0' being correct and '1' being incorrect in a separate .csv file.

$$precision = \frac{true\ positive}{total\ predicted\ positive} \quad - \ (2)$$

$$recall = \frac{true\ positive}{total\ actual\ \ positive} \quad - \ (3)$$

$$f1\ score\ = 2 * \frac{true\ positive}{total\ predicted\ positive} \quad - \ (4)$$

To select the optimum value of C, we plot the graph of C vs the F1-score of our training set.



The above graphs represent $C_{left\_shoulder\_angle}$ vs precision, recall and f_score calculated. As we can see, the precision increases when the value of C increases. However, the recall decreases. To find the optimum value of C, we calculate the f1 score from the equation (4).Here the f1_score is highest when C = 35 Thus, optimum value was taken as c = 35;

Similarly, we calculate the optimum C for all the other variables.

RESULTS:

Right now, we are able to tell which variable deviates from its original path. From our analysis, we have given preset values of Cs of individual variables. A person may change the sensitivity of the application by adjusting them.

To test the accuracy of the model, we created another dataset with 15 incorrect videos. We labeled each of the variables of the videos as ‘0’ and ‘1’, ‘0’ being correct and ‘1’ being incorrect in a separate .csv file.

We tested each video by using our algorithm keeping our reference videos as the five correct videos from the training dataset and the preset C values and compared with the actual results from the above .csv file. The results for our test Data-set are as below.

Score using sum of Euclidean Distance

Relevant Variables	Precision	Recall	F_score	No. of Examples (‘0’-Correct ‘1’-Incorrect)
l_shoulder_angle	0.666	1.000	0.799	‘0’>11, ‘1’>4
r_shoulder_angle	0.666	1.000	0.799	‘0’>11, ‘1’>4
l_elbow_angle	0.800	0.800	0.800	‘0’>10, ‘1’>5
r_elbow_angle	0.800	0.800	0.800	‘0’>10, ‘1’>5
l_hip_angle	0.600	0.600	0.600	‘0’>10, ‘1’>5
r_hip_angle	0.574	0.666	0.611	‘0’>9, ‘1’>6
l_shoulder_ratio	1.000	1.000	1.000	‘0’>11, ‘1’>4
r_shoulder_ratio	1.000	1.000	1.000	‘0’>11, ‘1’>4
l_elbow_ratio	0.666	0.800	0.726	‘0’>10, ‘1’>5
r_elbow_ratio	0.666	0.800	0.726	‘0’>10, ‘1’>5

From the above table, we saw that as the lower body remains stagnant during the Yoga exercise, time warping does not help much. Thus, instead of finding Euclidean distance in the adjusted Dataframes, we can directly find the difference of their means. This algorithm suits better considering this exercise

Score using difference of Mean

Relevant Variables	Precision	Recall	F_score	No. of Examples (‘0’-Correct ‘1’-Incorrect)
l_hip_angle	0.666	0.800	0.726	‘0’>10, ‘1’>5
l_hip_angle	0.714	0.833	0.769	‘0’>9, ‘1’>6

The above test results can be considered good results. And hence, proving the efficacy of the Algorithm.

CONCLUSION AND FUTURE WORK :

In this report, We have developed the application, Pose Trainer, which calculates the error in the exercise performed concerning the ideal way of performing it. We have used the output of pose estimation to evaluate videos of activities through human pose keypoints. We have worked with a specific exercise: Pranamasana/Prayer Pose. We went through recording training videos for each of them and further using machine learning algorithms to determine the correctness of the posture concerning the dataset.

We have further planned to extend this from a yoga exercise to a larger dataset. We are also planning to Use KNN to identify the exercise from the larger Dataset of Yoga Exercises. We are also planning to include dynamic related approaches like real-time yoga detection etc. Further, we can improve its pose feedback by providing suggestions for improvement along with error.

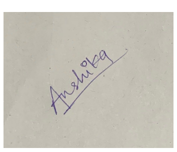
References:

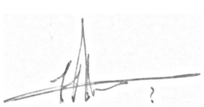
1. Steven Chen and Richard Yang. (2018, March): *Pose Trainer: Correcting Exercise Posture using Pose Estimation*. ResearchGate. https://www.researchgate.net/publication/324759769_Pose_Trainer_Correcting_Exercise_Posture_using_Pose_Estimation
2. pawan_gfg (2020, June). *OpenPose : Human Pose Estimation Method*. GeeksforGeeks. https://www.geeksforgeeks.org/openpose-human-pose-estimation-method/
3. Chathurangi Shyalika. (2019, May). Dynamic Time Warping (DTW). Medium. https://medium.com/datadriveninvestor/dynamic-time-warping-dtw-d51d1a1e4afc
4. *OpenPose: Real-time multi-person keypoint detection library for body, face, hands, and foot estimation*. (2016). Github. https://github.com/CMU-Perceptual-Computing-Lab/openpose
5. Shruti Saxena. (2018, May). *Precision vs Recall*. Medium. https://towardsdatascience.com/precision-vs-recall-386cf9f89488#:~:text=Precision%20and%20recall%20are%20two,correctly%20classified%20by%20your%20algorithm.
6. Meinard Müller

   https://www.researchgate.net/project/Fundamentals-of-Music-Processing

SIGNATURES:

Students:

1. 

2. 

Supervisors:

1.