Yoga Pose Classification using Features Extracted from Key-Point Detection

Aniket Verma
aniket19233@iiitd.ac.in

Hardik Garg
hardik19040@iiitd.ac.in

Tarini Sharma
tarini19451@iiitd.ac.in

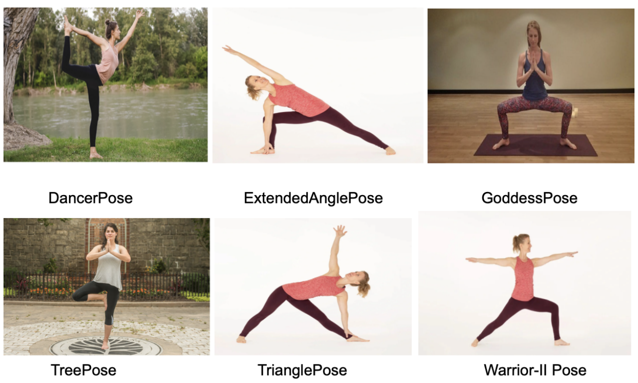

Figure 1. some poses used in the paper

Abstract

In this paper, we present a method to classify and score six yoga poses - Tree Pose, Warrior-2 Pose, Goddess Pose, Triangle Pose, Extended Side Angle Pose, and Dancer Pose by designing custom features from human pose key-points. Real life instruction for a large number of poses involves emphasis on angle, the shape that is being formed and the ratio of distances between body parts. To evaluate the effectiveness of our proposed features, we create a dataset for the six poses and apply ML techniques on the feature data extracted from them. The predictions are evaluated using metrics such as accuracy and F1-score and the yoga pose is scored using cosine similarity which can be used as a corrective measure by a self-learner.

1. Introduction

Yoga is an ancient Indian practice that has gained prominence among medical researchers due to its holistic focus on physical, mental, and spiritual development. It is shown to produce positive results in facilitating rehabilitation and sometimes even wholly curing diseases. With the advent of the COVID-19 pandemic, many people resorted to yogic practices for better health and immunity, much of which was done online. In such a scenario, it becomes essential that the participants perform the pose correctly because an incorrect pose can potentially have detrimental effects on their health.

Our aim, therefore, is to design a robust system that can help in the classification and grading of yoga poses using various Computer Vision, Machine Learning, and Deep Learning Techniques. The system will be given a yoga pose image as the input. The output will be the correct pose with a score indicating the pose accuracy, making it a multi-class classification problem that can thus be evaluated using metrics such as accuracy and F1-score. We consider six different poses for now - Tree Pose, Warrior-2 Pose, Goddess Pose, Triangle Pose, Extended Side Angle Pose, and Dancer Pose. These specific poses are picked due to their similarities - making partial modifications in either the position of hands or legs one can transition between the poses. Therefore, we assume that if our proposed system can distinguish between similar poses, it will work well for the plethora of different poses as well. Our primary motivation to take up this project is to make yoga more accessible and easy to practice for a self-learner.

Despite many studies and experiments on yoga pose classification, there are quite a few challenges involved in this task. Firstly, there are no publicly available datasets for the poses mentioned above. We found two datasets - Yoga-82 and a Kaggle dataset. However, they either did not have the six poses or had distorted images of the poses. Secondly, after a thorough literature review, we could find no work which did feature engineering for yoga poses. Most of the works employed either a CNN with the image being fed directly into the model or a deep Neural Network architecture where image keypoints data was fed to the model. There was no direct mapping, or invisible to the user if any, between the features and the outputs, which makes the system of little use to a self-learner who wants to figure out the error in their pose. Moreover, Deep Learning based architectures are computationally expensive. They might not be feasible in lightweight applications such as a group yoga class where the model has to yield fast results.

We address the challenges mentioned above in the following manner: we create a dataset of the six yoga poses using Google Images and verify each image from online Yoga Websites curated by Professional Yoga teachers. Next, we design image-level features using key points of the human pose which is done using AlphaPose technique [5], which also accounts for challenges posed by background and foreground variations. These image-level features are designed to emulate real-life instructions given by Yoga Teachers [1] [2] such as angles, shapes, and ratios between different body parts. These features are then used to develop ML-based systems to classify the pose and score them using cosine similarity. We also conduct experiments to verify whether the classification can be done without using ML techniques with comparable results, which would imply that the features designed can differentiate between the poses clearly.

The specific contributions made by this paper are as follows: (i) Several image-level features designed from key points of the human body to emulate angles, shapes, and ratios between body parts while performing yoga poses. (ii) Building ML-based systems to classify the poses correctly. (iii) Score the poses using their similarity score, i.e., the cosine similarity.

2. Related Works

A lot of work has been done with regards to the topic and one of them was [3] [4] which proposed human pose estimation and then classifies the activities on the basis of extracted pose key points computed using the Open-Pose Framework. 18 different key points have been used to associate the 2D structure of the body and then the problem is formulated as a multi-classification problem. [9] [10] used 28.4K images and clubbed them into 82 yoga pose classes. Further they were merged and collapsed to 20 subclasses and finally 6 super classes at the top level. The paper showed different results corresponding to various CNN architectures used for examples DenseNet-201 which boosted the accuracy. Further [8] [6] [7] used the MediaPipe framework to get keypoints from video frames. Media Pipe pose is an estimation library which outputs 33 key points (body joints). These key points correspond to 3d coordinates - x, y coordinates and visibility. Normalized features are formed using the 33 key points and joint angles formed between each of the keypoints (calculated using arccos of 2 distinct key points). Finally results are predicted using supervised ML algorithms.

3. Methodology

We follow a basic pipeline and each step is listed down and explained further.

3.1. Data Extraction

We extract key-points of human pose using Media-Pipe pose which outputs 33 key points (body joints). These key points correspond to 3d coordinates - x, y, z coordinates and visibility. Normalized features are formed using the 33 key points and joint angles formed between each of the key-points (calculated using arccos of 2 distinct key points). We generate our own custom features that improve the explanability as well as provide a logical understanding of pose balance.

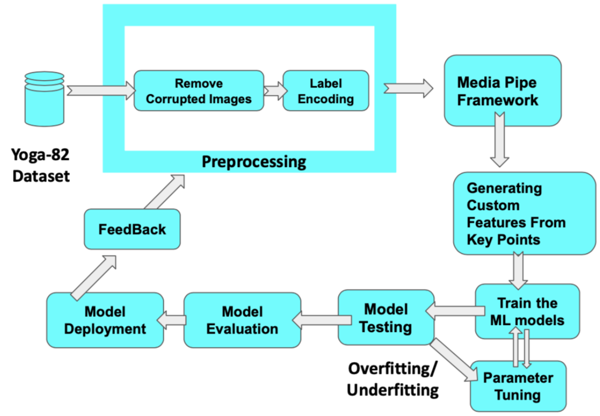

Figure 2. Work-flow/Pipeline of our model

Class	Features
Hand Features	Max hand angle
Hand Features	Min hand angle
Hand Features	Max Hand to Leg angle
Hand Features	Min Hand to Leg angle
Elbow Features	Elbow to elbow
Knee Features	Knee to knee
Leg Features	Max leg angle
Leg Features	Min leg angle
Leg Features	Max Hand to Leg angle
Leg Features	Min Hand to Leg angle
Waist Features	Nose-waist-max
Shoulder Features	Feet to shoulder ratio
Shoulder Features	Hand to shoulder ratio

Table 1. some of the designed features

3.2. Custom Features

From the KeyPoints obtained using the MediaPipe framework we have designed 25 custom pose features. These 25 custom features can be merged and collapsed to 6 super classes, i.e. Hand Features, Elbow features, Knee Features, Leg features, waist features, Shoulder features. These features have been designed and classified on the basis of body balance and pose correctness which plays an important role while performing Yoga Asanas. While describing the features we'll use the term P_i to denote the ith key-point given by media-pipe.

Following is an explanation behind each feature.

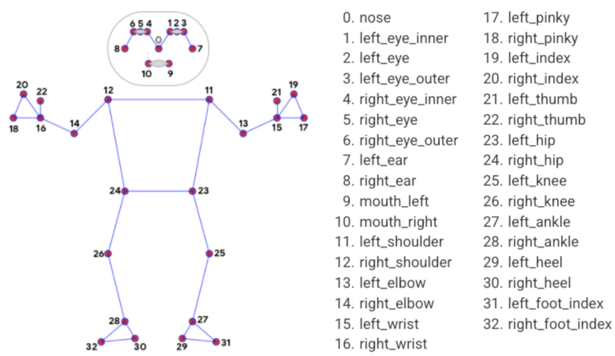

Figure 3. MediaPipe KeyPoints obtained for a humanPose

3.3. Hand features

We have designed five features under the Hand Features class i.e, Max hand angle , Min hand angle, Max Hand to Leg angle, Min Hand to Leg angle.

$$Max\ hand\ angle = \max(LeftElbowAngle, RightElbowAngle) \quad (1)$$

$$LeftElbowAngle = \cos^{-1}\frac{(P_{13}-P_{11})^2+(P_{15}-P_{13})^2-(P_{15}-P_{11})^2}{2|P_{13}-P_{11}||P_{13}-P_{15}|} \quad (2)$$

$$RightElbowAngle = \cos^{-1}\frac{(P_{14}-P_{12})^2+(P_{16}-P_{14})^2-(P_{16}-P_{12})^2}{2|P_{14}-P_{12}||P_{14}-P_{16}|} \quad (3)$$

$$Max\ Hand\ to\ Leg\ Angle = \max(LeftHandtoLegAngle, RightHandtoLegAngle) \quad (4)$$

$$Min\ Hand\ to\ Leg\ Angle = \min(LeftHandtoLegAngle, RightHandtoLegAngle) \quad (5)$$

$$LeftHandtoLegAngle = \cos^{-1}\frac{(P_{13}-P_{23})^2+(P_{25}-P_{23})^2-(P_{13}-P_{25})^2}{2|P_{13}-P_{23}||P_{23}-P_{25}|} \quad (6)$$

$$RightHandtoLegAngle = \cos^{-1}\frac{(P_{14}-P_{24})^2+(P_{26}-P_{24})^2-(P_{14}-P_{26})^2}{2|P_{14}-P_{24}||P_{24}-P_{26}|} \quad (7)$$

$$Hand\ to\ Shoulder\ Ratio = \frac{||P_{15}-P_{16}||_2}{||P_{11}-P_{12}||_2} \quad (8)$$

These features cater towards measuring the flexibility of hand, arms and correctness while performing the Yoga asanas.

3.4. Elbow features

We have designed one feature under elbow feature class i.e., elbow to elbow distance.

$$Elbow\ to\ Elbow\ Midpoint = \frac{||P_{13}-P_{14}||_2}{2} \quad (9)$$

This feature again measures the distance between one elbow to another and is designed keeping in the mind the flexibility and correctness of elbow posture while performing the yoga asanas.

3.5. Knee features

We have designed one feature under the knee feature class i.e., knee to knee distance.

$$Knee\ to\ Knee\ Midpoint = \frac{||P_{25}-P_{26}||_2}{2} \quad (10)$$

This feature caters towards maintaining apt distance between knees and computes the middle point of the line joining between 2 knees that helps in measuring the stability as the more is the distance between the knees, closer is the COG close to the ground, enhancing the stability and balance of the body.

3.6. Leg features

We have designed four features under the Hand Features class i.e, Max leg angle , Min leg angle Max Hand to Leg angle Min Hand to Leg angle.

$$Max\ leg\ angle = \max(LeftKneeAngle, RightKneeAngle) \quad (11)$$

$$Min\ leg\ angle = \min(LeftKneeAngle, RightKneeAngle) \quad (12)$$

$$RightKneeAngle = \cos^{-1}\frac{(P_{24}-P_{26})^2+(P_{26}-P_{28})^2-(P_{28}-P_{24})^2}{2|P_{28}-P_{26}||P_{24}-P_{26}|} \quad (13)$$

$$LeftKneeAngle = \cos^{-1}\frac{(P_{23}-P_{25})^2+(P_{25}-P_{27})^2-(P_{27}-P_{23})^2}{2|P_{27}-P_{25}||P_{23}-P_{25}|} \quad (14)$$

The rest 2 features are given in equation(4) ,(5), (6), (7).

3.7. waist features

We have designed one feature under the waist Feature class i.e, Nose-waist-max(heel)

$$NoseWaistMax(heel) = \max(NoseToLeftHeelAngle, NoseToRightHeelAngle) \quad (15)$$

$$NoseToLeftHeelAngle = \cos^{-1}\frac{(P_0-P_{waist})^2+(P_{waist}-P_{30})^2-(P_{30}-P_0)^2}{2|P_{30}-P_{waist}||P_0-P_{waist}|} \quad (16)$$

$$NoseToRightHeelAngle = \cos^{-1}\frac{(P_0-P_{waist})^2+(P_{waist}-P_{29})^2-(P_{29}-P_0)^2}{2|P_{29}-P_{waist}||P_0-P_{waist}|} \quad (17)$$

$$P_{waist} = \frac{||P_{23}-P_{24}||_2}{2} \quad (18)$$

3.8. Shoulder Features

We have designed two features under the Shoulder feature class i.e, Feet to shoulder ratio, Hand to shoulder ratio.

$$Feet\ to\ Shoulder\ Ratio = \frac{||P_{31}-P_{32}||_2}{||P_{11}-P_{12}||_2} \quad (19)$$

The hand to shoulder ratio is given in equation (8).

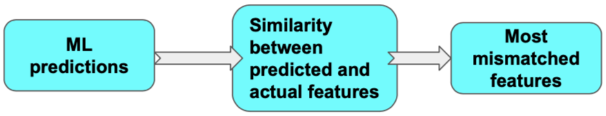

Figure 4. Similarity used for Pose correction

3.9. Pose correction

Alongside feature engineering we also propose a method for pose scoring and correction base upon similarity of feature values. We make use of the training set to get the benchmark feature values for each pose. Therefore we have 82 benchmarks one for each pose. We define the similarity as:

$$Similarity = 1 - \frac{||x - y||}{||x + y||} \tag{20}$$

x and y represent the feature values which can either be an angle or distance.

4. Experiments

In this section, we first discuss the dataset used. Second, we discuss the details of the different experiments conducted. Third, we discuss the different results we have obtained.

4.1. Dataset

We have used the Yoga-82 dataset for training our machine learning models. Yoga-82 dataset has about 28.4K images which contains diverse yoga pose images classified into 82 subclasses which are then merged and collapsed to 20 superclasses and finally merged and collapsed to 6 superclasses at the top level. The dataset is classified using 6 hierarchical annotations namely, standing, sitting, balancing, inverted, reclining, wheel, etc. Hierarchical annotations are beneficial for learning the network as they provide rich information to users not only about the pose names but also about the body postures.

NOTE: While using this dataset, we found appx. 9K images were corrupted and our experiments are conducted on the 19K images.

4.2. Experiment Details

We use various machine learning algorithms such as XGBoost ,SGD-Classifier, Gaussian Naive Bayes (NB), Random Forests (RF) , KNN to learn models that can predict the presence of a particular disorder. We report the classification accuracies of both training phase and testing phase. We conduct two sets of experiments one is by training our models using vanilla values (media pipe features). Second is by using the proposed custom features which are fed into the machine learning models and then the predictions are made. Finally we compare the results from both the models.

ML Algorithm	Testing Score	F1-score
SGD classifier	0.223253	0.191090
Gaussian Naive Bayes	0.883306	0.884005
Random Forest Classifier	0.9213714	0.9210666
XGBoost	0.882443	0.881529
KNN	0.705640	0.700567

Table 2. Experimental results using Custom Features

ML Algorithm	Testing Score	F1-score
SGD classifier	0.513276	0.494961
Gaussian Naive Bayes	0.278680	0.268506
Random Forest Classifier	0.763341	0.757041
XGBoost	0.750708	0.747487
KNN	0.746517	0.742577

Table 3. Experimental results using raw Features

4.3. Results

From the tables we can see that Random Forest Classifier gives the best results with testing accuracy of 0.9213714 and F1-score of 0.921666 for our custom features. We can clearly see that custom features designed perform much better than the raw features given by media pipe. Also we have improved a lot on the F1-score which was proposed in paper Yoga-82 paper.

5. Contribution

The paper contributed in reducing features from 132 to 25 and introducing more practical and explainable features by Taking into account the geometric relations between keypoints given by the media pipe pose framework. Secondly, Scoring and pose correction of yoga poses using custom designed features and similarity which gives great results.

6. Conclusion

We develop several custom 3D pose features, like hand, shoulder, knee, elbow, waist features mentioned in the Methodology section. We employ the key-points of the human pose for designing them. Our experiments demonstrate their superior performance in comparison to the existing ones.

7. References

References

[1] Yoga with adriene youtube channel.

[2] Yoga journal — online yoga archives, Mar 2022.

[3] Kshama Gupta Kapil Gupta Abhay Gupta, Kuldeep Gupta. Human activity recognition using pose estimation and machine learning, Feb 2021.

[4] Ajay Chaudhari, Omkar Dalvi, Onkar Ramade, and Dayanand Ambawade. Yog-guru: Real-time yoga pose correction system using deep learning methods. In *2021 International Conference on Communication information and Computing Technology (ICCICT)*. IEEE, June 2021.

[5] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation, Feb 2018.

[6] Daksh Goyal, Koteswar Rao Jerripothula, and Ankush Mittal. Detection of gait abnormalities caused by neurological disorders. *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020.

[7] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, and et al. Mediapipe: A framework for building perception pipelines, Jun 2019.

[8] Shikha Gupta Utkarsh Bahukhandi. Yoga pose detection and classification using machine learning techniques, Dec 2021.

[9] Manisha Verma, Sudhakar Kumawat, Yuta Nakashima, and Shanmuganathan Raman. Yoga-82: A new dataset for fine-grained classification of human poses, Apr 2020.

[10] Santosh Kumar Yadav, Amitojdeep Singh, Abhishek Gupta, and Jagdish Lal Raheja. Real-time yoga recognition using deep learning. *Neural Computing and Applications*, 31(12):9349–9361, May 2019.