JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

SURYA NAMASKAR WITH REAL-TIME FEEDBACK USING MEDIAPIPE

[1]Dr. K. Shirisha Reddy, [2]Jhanavi Putcha, [3]Neeraj Kumar Pola, [4]Ankith Reddy Subhanpuram
[1]Associate Professor & HOD, [2]Student/Research Scholar, [3]Student/Research Scholar, [4]Student/Research Scholar
Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning)
Vignana Bharathi Institute of Technology, Aushapur(V), Ghatkesar (M), Medchal Dist 501301, Telangana, India.

Abstract: Yoga is a holistic discipline that has been practiced for thousands of years. It is a fusion of the mind, body, and spirit. Sun Salutation, sometimes referred to as Surya Namaskar, is a series of rhythmically flowing yoga poses, or asanas. Surya Namaskar can be helpful for physical fitness, flexibility and joint mobility, energy and vitality, mind-body connection, stress reduction and relaxation, improved digestion and metabolism, and spiritual connection. The goal of this project is to give people a tool or system that will direct and help them practice Surya Namaskar correctly and efficiently. The Surya Namaskar is a series of defined asanas that has to be performed in an order. There are total of 12 asanas in a sequence with 8 of them are being unique and the other 4 are repeated asanas of the 8. The goal is to provide a platform or application that uses less computation power, high speed, interactive and incorporates visual cues, aural cues, and real-time feedback to assist users in maintaining good posture and aligning their movements.
Keywords: Computer vision, Open CV, MediaPipe, Real-time feedback, Landmarks, Classification, Object detection, Pre-defined angles.

I. INTRODUCTION

While existing Surya Namaskar practice models utilizing computer vision and Long Short-Term Memory (LSTM) networks effectively categorize pose variations and provide feedback on form, they lack a crucial element real-time guidance during practice. This presents a challenge for users who may struggle to remember the complex sequence of postures and identify minor form errors. Therefore, there is a need for a new method that builds upon the strengths of existing pose detection models but offers a more interactive and user-friendly experience for Surya Namaskar practice in real-time.

In this paper, we will be discussing about how the usage of MediaPipe and OpenCV drastically improves the performance of the model, with less computation power.

1.1 OpenCV:

OpenCV is an open-source library containing hundreds of computer vision algorithms, offering robust handling of various image formats, efficient image manipulation and analysis, extensive documentation, community support, and a modular structure for seamless integration with other libraries and frameworks.

1.2 MediaPipe:

MediaPipe, introduced by Google in 2019, is another powerful library that focuses on building cross-platform, scalable, and efficient pipelines for media processing tasks. Unlike traditional computer vision libraries, MediaPipe offers a holistic framework for developing end-to-end pipelines, encompassing tasks such as pose estimation, hand tracking, object detection, and augmented reality (AR) effects.

One of the distinguishing features of MediaPipe is its emphasis on real-time processing and mobile deployment. Leveraging hardware acceleration techniques such as GPU and DSP (Digital Signal Processor), MediaPipe enables high-performance inference on a variety of devices, including smartphones, tablets, and embedded systems. Furthermore, MediaPipe provides pre-trained models and ready-to-use components, streamlining the development process and reducing the need for extensive manual annotation and training.

The main feature of MediaPipe is its human pose landmark detection, it has been trained with CNN: MobileNetV2-like with customized blocks for real-time performance. The model returns 33 key points describing the approximate location of body parts.

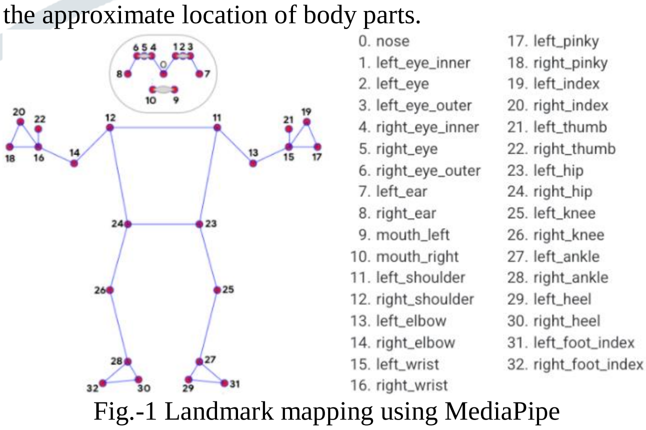

Fig.-1 Landmark mapping using MediaPipe

Each landmark has 5 values (X, Y, Z, visibility, presence)

- X, Y coordinates are local to the region of interest and range from [0.0, 255.0].
- Z coordinate is measured in "image pixels" like the X and Y coordinates and represents the distance relative to the plane of the subject's hips, which is the origin of the Z axis. Negative values are between the hips and the camera; positive values are behind the hips. Z coordinate scale is similar with X, Y scales but has different nature as obtained not via human annotation, by fitting

synthetic data (GHUM model) to the 2D annotation.

- Visibility is in the range of [min_float, max_float] and after user-applied sigmoid denotes the probability that a keypoint is located within the frame and not occluded by another bigger body part or another object.
- Presence is in the range of [min_float, max_float] and after user-applied sigmoid denotes the probability that a keypoint is located within the frame.

II. LITERATURE REVIEW

This section reviews existing theories and research within the scope of the report, aiming to contextualize the proposed system. A literature survey is conducted to provide a comprehensive understanding of the exploration or project, involving a meticulous study of pre-existing materials and logical framing of the system being developed.

2.1 Surya Namaskar: Real-Time Advanced Yoga Pose Recognition and Correction for Smart Healthcare

Authored by Abhishek sharma, Pranjal Sharma, Darshan Pincha, Prateek Jain, explores about creating a real time Surya Namaskar application using LSTM model.

In the paper, they go on to discuss about the procedure of training LSTM model. Initially using OpenCV, the frames of the users are captured via webcam, then the human pose landmarks are exctracted using MediaPipe, these keypoints are concatenate into numpy arrays which are then stored into folder for training and testing. These Datasets are trained using LSTM with help of tensorflow and keras. The trained model is used for prediction of each asana in realtime.

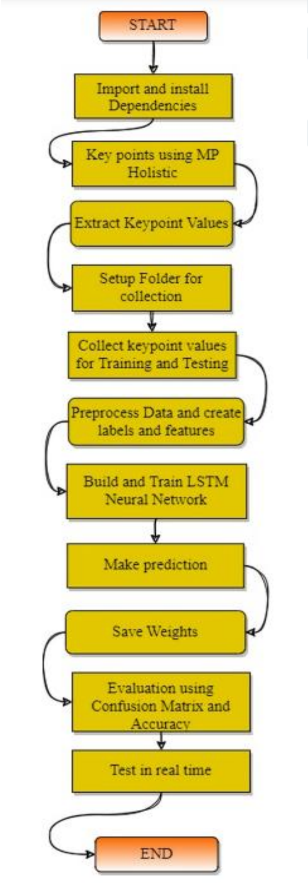

Fig.-2.1 Methodology

2.2 Yoga Pose Estimation and Feedback Generation Using Deep Learning

Authored by, Vivek Anand Thoutam, Anugrah Srivastava, Tapas Badal, Vipul Kumar Mishra, G. R. Sinha , Aditi Sakalle , Harshit Bhardwaj , and Manish Raj explores about classification of yoga poses specifically and not the poses where sequence is defined(like suryanamaskara).

The paper discusses the methodology in the following way:

Initially the video frames or images are captured using OpenCV via webcam, these images are then given input to the MediaPipe, which returns 33 landmarks as discussed earlier, all the angles between the landmarks are calculated and these are sent to classification model for training, a confidence level of 30% is initialized.

Now in realtime, when a user performs an asana, the model captures and extracts landmarks using OpenCV and MediaPipe respectively, the angles are calculated and sent to the model, if the confidence is greater than 30%, the model returns an asana name along with realtime feedback with the difference in the angles made by user and the actual angles that are needed be.

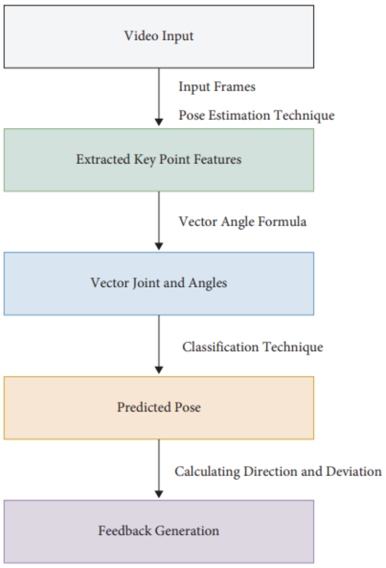

Fig.-2.2 Methodology

III. PROPOSED METHOD

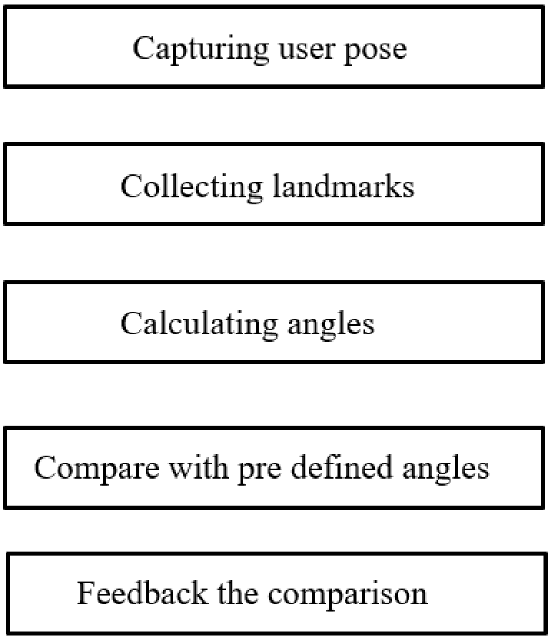

Fig.-3 Overview of the proposed method

1. Capturing user pose

It is done with the help of OpenCV. Every frame(image) is a part of the video where user performing an asana in front of the webcam. These frames are synchronously passed to MediaPipe.

2. Collecting landmarks

Using pose detection of MediaPipe, 33 landmarks are extracted from a frame if human is present else no landmarks can be extracted from the frame.

3. Calculating angles

Angles are calculated as using any three landmarks from the body parts that have been extracted using pose detection. The three landmarks are used as coordinate points and the angle between them is found out by using arctan function which is provided by math library in python

4. Compare with pre-defined angles & Feedback the comparison

Every asana has different sets of pre-defined angles, these pre-defined angles can be stored in separate files to increase the readability and modularity.

The pre-defined or ideal angles for each pose is extracted with the help of expert who perform an asana with no errors.

The landmarks that are extracted from 2nd step are given as input to a classification function, the function uses these landmarks, to calculate crucial angles that are required for different asana, these angles are then given as input to the predefined functions for each asana which are arranged in different python files as we have discussed earlier.

This predefined function returns feedback to classification function, which is then displayed in real time for the user. Further, this feedback is used to increase the timer and move on to next asana once the target is reached.

IV. IMPLEMENTATION

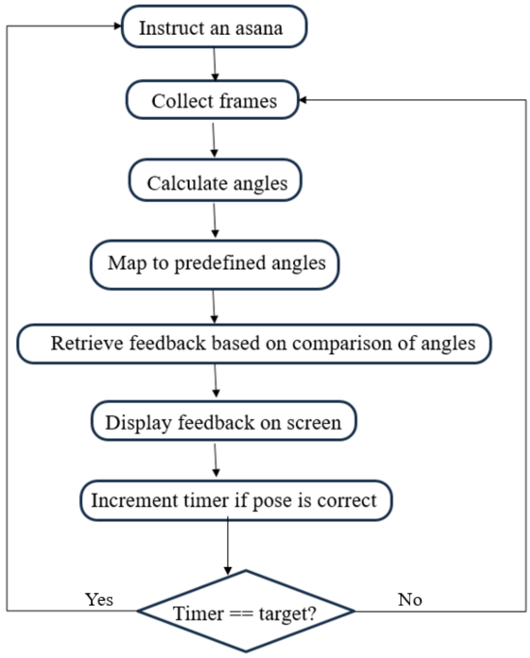

Fig.-4 System Architecture

1. **Instruct an asana**

Initially, when the user starts the applications, as Surya Namaskar consists of sequentially defined asanas, the first asana (Pranamasana) is displayed along with an image of showing the asana by which the user can get basic idea to follow the asana.

2. **Collect frames**

Once, the asana is instructed, start collecting the frames using OpenCV.

3. **Calculate angles**

Each frame that is being extracted is given as an input to MediaPipe, it returns 33 landmarks. Now for each asana different landmarks are considered crucial, for example for the first asana i.e, Padmasana elbows are considered crucial as they are required for praying, so the angles around elbow is calculated with help of other two neighbouring landmarks, in this case shoulder and wrist.

The angle between three landmarks is calculates using an arctan function which is provided by math library in python.

4. **Map to predefined angles**

For each asana, the predefined angles are created with the help of an expert, he does the asana which is being captured by Mediapipe, so the angles that derive are accurate to the asana that is being done.

Each asana has certain crucial landmarks and predefined angles.

5. **Retrieve feedback based on comparison of angles**

Compare the predefined angles with angles made by the current user, if the user made angles fall in the predefined angles, the feedback would be good, else the difference are calculated and appropriate feedback is returned

6. **Display feedback on screen**
   Once the feedback is returned, display the feedback to screen.

7. **Increment time if pose is correct:**
   Initialize a counter variable, if the user's angles fall in the range of predefined angles, increment the counter

8. **Timer == target?**
   Assign a target of duration for each asana to be done, once the counter reached target, instruct the user with next asana and also the counter should be initialized to 0.

V. COMPONENT OF NOVELITY IN THE PAPER

A novel approach is proposed in this paper, combining the strengths of two existing models to enhance the practice of Surya Namaskar. Traditionally, modelling Surya Namaskar with LSTM-based architectures [1] has been computationally intensive and demanding in terms of data requirements. Meanwhile, generic yoga asana models [2] provide valuable insights into posture correctness but lack the sequential understanding necessary for Surya Namaskar.

Our innovation lies in integrating the feedback mechanism from the generic yoga asana classification model [2] into the sequential model [1] designed specifically for Surya Namaskar. By leveraging the strengths of both models, we eliminate the need for LSTM, thereby reducing computational complexity, data dependency, and training time.

The resulting hybrid model not only accurately captures the sequential nature of Surya Namaskar but also offers real-time feedback on posture correctness without asynchronous actions. This integration not only enhances computational efficiency but also ensures that practitioners receive precise guidance throughout their practice.

VI. RESULTS

Fig.-5 Implementation of Pranamasana

Fig.-6 Incorrect posture detected

Fig.-7 Error detected while performing Hasta Padasana

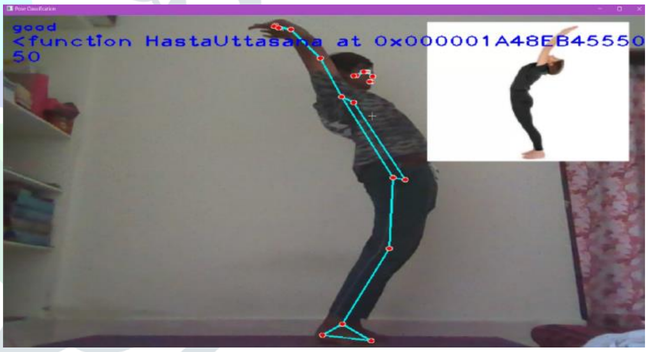

Fig.-8 Implementation of Hasta Uttasana

VII. OBSERVATIONS FROM RESULTS

- Minor mistakes are identified accurately and feedback is returned.
- Identify complex posturers.
- Real time feedback with no delay for the users.
- Sequence of asanas is not compromised.
- Low computation power is sufficient (unlike LSTM models which require high computation power).
- As the data for Surya Namaskar is scarce in the internet, it becomes harder to train an LSTM model with high accuracy, our model performs accurately with less data (only one expert performing the entire Surya Namaskar).

VIII. CONCLUSION

This project explored a novel approach to analysing Surya Namaskar poses using static image capture and MediaPipe. The proposed methodology offers several advantages:

- Personalized Guidance: Immediate guidance and corrections.
- Improved Learning Experience: Engaging and dynamic practice environment.
- Minimize Faults: Reduce errors in execution.
- Facilitate Self-Practice: Enable independent and accurate practice.
- Simplified User Interaction: By capturing a single image, the system eliminates the need for video processing, making it user-friendly and convenient.
- Detailed Feedback: The analysis of joint angles provides in-depth insights into pose accuracy, allowing users to identify and correct specific misalignments.

IX. ACKNOWLEDGEMENT

We would like to thank Dr. K. Shirisha Reddy for her valuable comments and suggestions to improve the quality of the paper. We are also grateful to Mrs. P. Navya for helping us review our work regularly. We would also like to thank the Department of Computer Science Engineering (AI & ML), VBIT Hyderabad for providing us Jetson Nano Boards and all the help we needed.

REFERENCES

[1]. L. Prasanna Venkatesh , Vandhana S “Insights on Surya namaskar from its origin to application towards health” Journal of Ayurveda and Integrative Medicine Volume 13, Issue 2, April–June 2022 .

[2]. Deepak Kumar, Anurag Sinha “Yoga Pose Detection and Classification Using Deep Learning” November 2020 International Journal of Scientific Research in Computer Science Engineering and Information Technology.

[3]. M. C. Thar, K. Z. N. Winn and N. Funabiki, "A Proposal of Yoga Pose Assessment Method Using Pose Detection for Self-Learning," 2019 International Conference on Advanced Information Technologies (ICAIT), Yangon, Myanmar, 2019, pp. 137-142, doi: 10.1109/AITC.2019.8920892.

[4]. Swain, D., Satapathy, S., Acharya, B., Shukla, M., Gerogiannis, V. C., Kanavos, A., & Giakovis, D. (2022). “Deep Learning Models for Yoga Pose Monitoring. Algorithms”, 15(11), 403. https://doi.org/10.3390/a15110403

[5]. Sharma, A., Sharma, P., Pincha, D., & Jain, P. (2022). “Surya Namaskar: Real-time advanced yoga pose recognition and correction for smart healthcare”. *ArXiv*. /abs/2209.02492

[6]. Satyam Goyal, Animesh Jain “Yoga Pose Perfection using Deep Learning” (2021).

[7]. Kishore, D. M., Bindu, S., & Manjunath, N. K. (2022). “Estimation of Yoga Postures Using Machine Learning Techniques. International Journal of Yoga”, 15(2), 137-143. https://doi.org/10.4103/ijoy.ijoy_97_22

[8]. W. Wu, W. Yin and F. Guo, "Learning and Self-Instruction Expert System for Yoga," 2010 2nd International Workshop on Intelligent Systems and Applications

[9]. A. Mohanty, A. Ahmed, T. Goswami, A. Das, P. Vaishnavi, and R. R. Sahay, “Robust pose recognition using deep learning,” in Proceedings of the International Conference on Computer Vision and Image Processing, pp. 93–105, Springer, Singapore, December 2017.

[10]. W. Gong, X. Zhang, J. Gonzàlez et al., “Human pose estimation from monocular images: a comprehensive survey,” Sensors, vol. 16, no. 12, p. 1966, 2016.

[11]. Chaudhari, A., Dalvi, O., Ramade, O., & Ambawade, D. (2021). Yog-Guru: Real-Time Yoga Pose Correction System Using Deep Learning Methods. In 2021 International Conference on Communication information and Computing Technology (ICCICT) IEEE.

[12]. U. Rafi, B. Leibe, J.Gall, and I. Kostrikov, “An efficient convolutional network for human pose estimation”, British Mach. Vision Conf., 2016.

[13]. Y. Agrawal, Y. Shah, and A. Sharma, “Implementation of Machine Learning Technique for Identification of Yoga Poses,”IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT). Apr. 2020.

[14]. Holte MB et al (2012) Human pose estimation and activity recognition from multi-view videos: comparative explorations of recent developments. IEEE J Selec Topics Signal Process 6(5):538–552

[15]. Borthakur, D., Paul, A., Kapil, D., & Saikia, M. J. (2022). Yoga Pose Estimation Using Angle-Based Feature Extraction. Healthcare, 11(24), 3133. https://doi.org/10.3390/healthcare11243133