International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 02 | Feb - 2022 ISSN: 2582-3930

# Real-Time Yoga Pose Recognition

Radha Tawar[1], Sujata Jagtap[2], Darshan Hirve[3], Tejas Gundgal[4]

*Radha Tawar[1] (Department of Information Technology) MET BKC IOE Nashik*
*Sujata Jagtap[2] (Department of Information Technology) MET BKC IOE Nashik*
*Darshan Hirve[3](Department of Information Technology) MET BKC IOE Nashik*
*Tejas Gundgal[4](Department of Information Technology) MET BKC IOE Nashik*

--------------------------------------------------------------------- * * *---------------------------------------------------------------------

**Abstract:** Human pose estimation is a deep-rooted problem in computer vision that has exposed many challenges in the past. Analyzing human activities is beneficial in many fields like video surveillance, biometrics, assisted living, at-home health monitoring, etc. With our fast-paced lives these days, people usually prefer exercising at home but feel the need for an instructor to evaluate their exercise form. As these resources are not always available, human pose recognition can be used to build a self-instruction exercise system that allows people to learn and practice exercises correctly by themselves. This project lays the foundation for building such a system by discussing various machine learning and deep learning approaches to accurately classify yoga poses on real-time. Using the system, the user can select the pose that he/she wishes to practice. He/she can then upload a photo of themselves doing the pose. The pose of the user is compared with the pose of the expert and the difference in angles of various body joints is calculated. Based on this difference of angles feedback is provided to the user so that he/she can improve the pose.

***Key Words**:* Human pose estimation, mediapipe, OpenCV, machine learning, deep learning, yoga

## 1.INTRODUCTION

Yoga is an ancient science that originated in India. According to the Bhagavad Gita, it is the remover of misery and the destroyer of pain. Recently, Yoga is getting popular across the globe due to its physical, mental, and spiritual benefits. In 2014, the General Assembly of United Nations has declared 21st June as the 'International Day of Yoga. Over the last decade, Yoga is getting increasing importance in the medical research community, and numerous pieces of literature have been proposed for various medical applications including cardiac rehabilitation, positive body image intervention, mental illnesses, etc. Without the use of medicines, Yoga can completely cure many diseases. Yoga exercises boost physical health as well as help to cleanse the body, mind, and soul. It comprises of many asanas and each of them denotes the static physical postures. Yoga exercises boost physical health as well as help to cleanse the body, mind, and soul. It comprises many asanas and each of them denotes the static physical postures. Human pose estimation is a challenging problem in the discipline of computer vision. It deals with the localization of human joints in an image or video to form a skeletal representation. To automatically detect a person's pose in an image is a difficult task as it depends on several aspects such as scale and resolution of the image, illumination variation, background clutter, clothing variations, surroundings, and interaction of humans with the surroundings. The problem with yoga however is that, just like any other exercise, it is of most importance to practice it correctly as any incorrect posture during a yoga session can be unproductive and possibly detrimental. This leads to the necessity of having an instructor supervise the session and correct the individual's posture. Nowadays people use their mobile phones to learn how to do yoga poses and start doing that but while doing that they don't even know whether the yoga pose they are doing is in the right way or not. To overcome these limitations, many works have been done. Computer vision and data science techniques have been used to build AI software that works as a trainer. This software tells about the accuracy of the performance. This undertaking centers around investigating the various methodologies for yoga.

which gives information about the coordinates of the joint of the body. This method is good but the main disadvantage of this method is that the Kinect device is expensive and not user-friendly. To avoid this problem

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 02 | Feb - 2022 ISSN: 2582-3930

## 2.SYSTEM ARCHITECTURE:

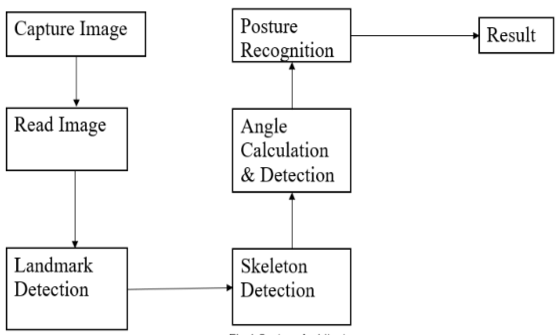


**Fig1**: System Architecture

## 3. SYSTEM IMPLEMENTATION PROCEDURE

### i. CAPTURE IMAGE:

In the first stage, Capture the image using an RGB camera.

### RGB CAMERA:

RGB camera used to capture the color and depth images. The camera is mounted and adjusted on a tripod with an appropriate frame centering on the person performing the yoga poses. The distance is maintained around 4 to 5m between the camera and the person.

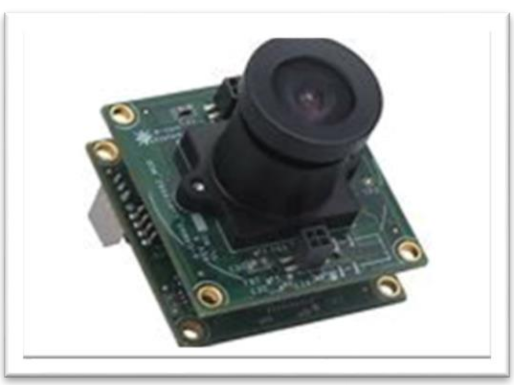

**Fig2**: RGB Camera.

### ii. READ IMAGE:

In the second stage, read a sample image using the function and then display that image using the matplotlib library. The image will be read using OpenCV.

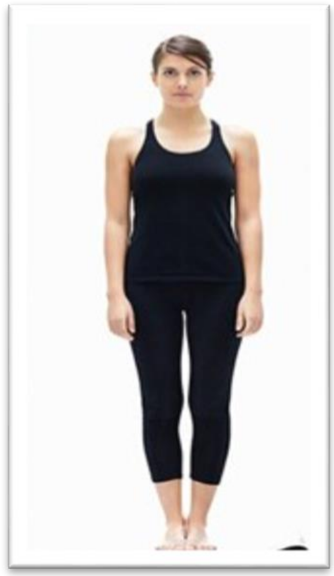

**Fig3**: Read Image

### OpenCV:

OpenCV is a Python open-source library, which is used for computer vision in Artificial intelligence, Machine Learning, etc. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java, C++. It works on every frame of the image. It contains library files that are used to calculate angles. It controls the camera from which the difference between a real image and an unknown image is known**.**

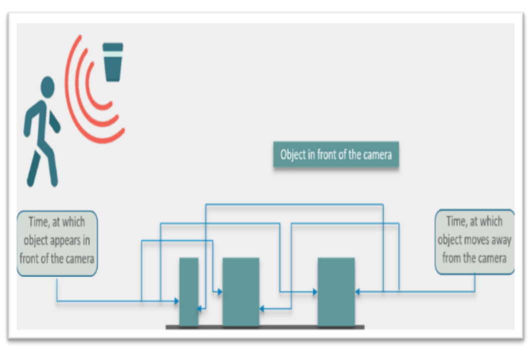


**Fig4**: OpenCV Architecture

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 02 | Feb - 2022 ISSN: 2582-3930

### iii. PERFORM LANDMARK DETECTION

In the third stage, mediapipe is used to create a skeleton of the person performing the yoga poses.
present the results as 33 simple key points:

Table 1: 33 keypoints

| | | | |
|---|---|---|---|
| 0. | Nose | 17. | Left_pinky |
| 1. | Left_eye_inner | 18. | Right_pinky |
| 2. | Left_eye | 19. | Left_index |
| 3. | Left_eye_outer | 20. | Right_index |
| 4. | Right_eye_inner | 21. | Left_thumb |
| 5. | Right_eye | 22. | Right_thumb |
| 6. | Right_eye_outer | 23. | Left_hip |
| 7. | Left_ear | 24. | Right_hip |
| 8. | Right_ear | 25. | Left_knee |
| 9. | Mouth_left | 26. | Right_knee |
| 10. | Mouth_right | 27. | Left_pinky |
| 11. | Left_Shoulder | 28. | Left_pinky |
| 12. | Right_Shoulder | 29. | Left_heel |
| 13. | Left_elbow | 30. | Right_heel |
| 14. | Right_elbow | 31. | Left_foot_index |
| 15. | Left_wrist | 32. | Right_foot_index |
| 16. | Right_wrist | | |

As shown in fig. 33 key points get detected

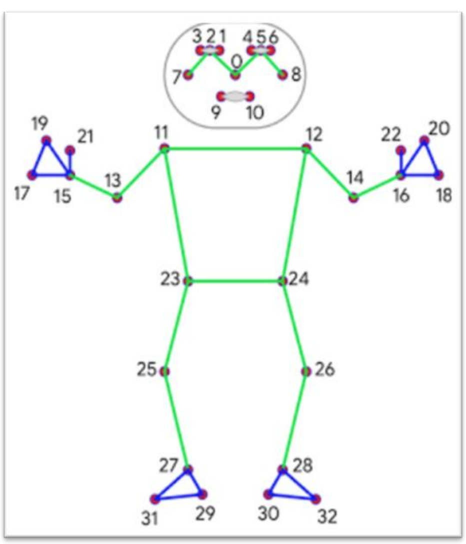

**Fig5**: Standing pose skeleton using Mediapipe

**MediaPipe**:

Mediapipe is a cross-platform library developed by Google that provides amazing ready-to-use ML solutions for computer vision tasks. MediaPipe Pose is an ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks, and background segmentation mask on the whole body from RGB video frames. Using a detector, the pipeline first locates the person region-of-interest (ROI) within the frame. The tracker subsequently masks within the ROI using the ROI-cropped frame as input. MediaPipe Pose can provide a full-body segmentation mask. Representation as a two-class segmentation (human or background)

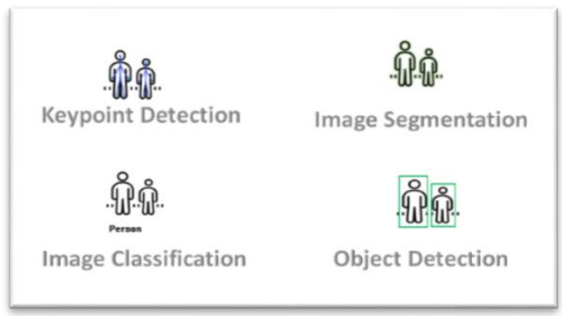


**Fig6**: Mediapipe Architecture

After performing the pose detection, a list of thirty-three landmarks representing the body joint locations of the prominent person in the image. Each landmark has **x**: It is the landmark x-coordinate normalized to [0.0, 1.0] by the image width.

**y**: It is the landmark y-coordinate normalized to [0.0, 1.0] by the image height.
**z**: It is the landmark z-coordinate normalized to roughly the same scale as x. It represents the landmark depth with a midpoint of hips being the origin, so the smaller the value of z, the closer the landmark is to the camera.
**Visibility**: It is a value with a range [0.0,1.0] representing the possibility of the landmark being visible (not occluded) in the image. This is a useful when variable wants to show a particular joint because it might be occluded or partially visible in the image.

After performing the pose detection on the sample image above, display the first two landmarks from the list, so that get a better idea of the output of the model. Now convert the two normalized landmarks displayed above into their original scale by using the width and height of the image.

### iv. POSE CLASSIFICATION WITH ANGLE HEURISTICS

In the fourth stage, classify different yoga poses using calculated angles of various joints. The first point (landmark) is considered as the starting point of the first line, the second point (landmark) is considered as the ending point of the first line and the starting point of the second line as well, and the third point (landmark) is considered as the ending point of the second line.

The formula for calculating the angle is shown below

Here's an equation: $\tan(x+y) = \frac{\tan x + \tan y}{1 - \tan x * \tan y}$

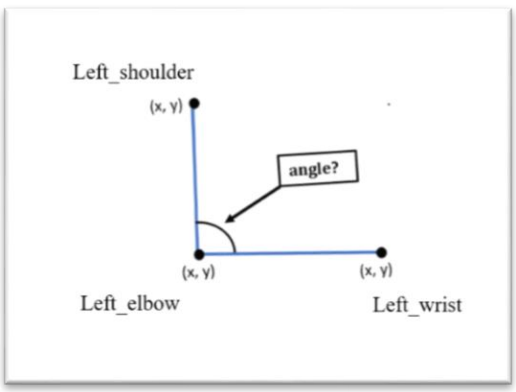


**Fig7**: Angle Heuristics

Table 2: Angles between keypoints

| Sr No. | Keypoints |
|---|---|
| 1. | Left shoulder –Left elbow – Left wrist. |
| 2. | Right shoulder –Right elbow – Right wrist. |
| 3. | Left hip –Left knee –Left ankle |
| 4. | Right hip –Right knee –Right ankle. |
| 5. | Left ankle – Left heel – Left foot index. |
| 6. | Right ankle – Right heel – Right foot index. |

calculates the angle between three different landmarks
landmark1: The first landmark containing the x, y, and z coordinates.
landmark2: The second landmark containing the x, y, and z coordinates.
landmark3: The third landmark containing the x, y, and z coordinates.
Get the coordinates of the required landmarks. Calculate the angle between the three points, if the angle is less than zero add 360 to the found angle. Return the calculated angle.

### v. POSE CLASSIFICATION

In the final stage, the pose can be classified using the combination of body part angles Initialize the label of the pose. It is not known at this stage as 'Unknown Pose'. Specify the color (Red) with which the label will be written on the image. Calculate the required angles, if the pose is classified successfully update the color (to green) with which the label will be written on the image. Setup Pose function for video. Initialize the Video. Capture object to read from the webcam. Initialize a resizable window. Iterate until the webcam is accessed successfully. Read a frame. Check if the frame is not read properly. Continue to the next iteration to read the next frame and ignore the empty camera frame. Flip the frame horizontally for natural (selfie-view) visualization. Get the width and height of the frame. Resize the frame while keeping the aspect ratio. Perform Pose landmark detection. Check if the landmarks are detected. Perform the Pose Classification. Display the frame. Wait until a key is pressed. Retrieve the ASCII code of the key pressed. Check if 'ESC' is pressed.

**Fig8**: Correct Pose


**Fig9:** Incorrect Pose

**Yoga Poses**

Table 3: Asanas

| | |
|---|---|
| 1. | Bhujangasana (Cobra Pose) |
| 2. | Padmasana (Lotus Pose) |
| 3. | Vrikshasana (Tree Pose) |
| 4. | Trikonasana (Triangle Pose) |
| 5. | T Pose |
| 6. | Tadasana (Mountain Pose) |
| 7. | Virabhadrasana 1(Warrior1 Pose) |
| 8. | Virabhadrasana 2(Warrior2 Pose) |
| 9. | Bhadrasana (Butterfly Pose) |

## vi. VOICE FEEDBACK

Feedback is provided to the user so that he/she can improve the pose. For voice feedback pyttsx3 library was used.

**pyttsx3**:
pyttsx3 is a cross-platform text to speech library which is platform-independent. The pyttsx3 is a text-to-speech conversion library. The pyttsx3 module supports two voices first is female and the second is male

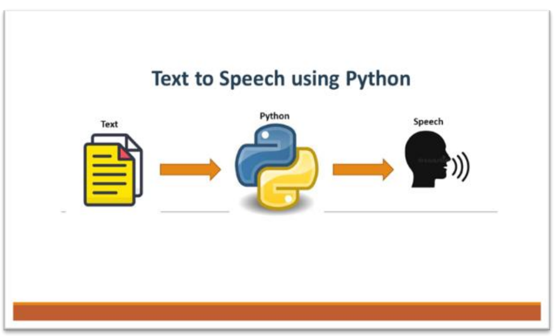


**Fig10**: Text to Speech

## vii. CONCLUSION

In this paper, we proposed a Yoga identification system using a traditional RGB camera. Yoga poses are implemented using OpenCV and mediapipe with much more accuracy. A system is suggested that classify nine yoga poses without any dataset as 6 yoga poses already performed by previous research papers, we are adding 3 more poses to the system. An audio guidance system is also included.

## ACKNOWLEDGEMENT

We would like to thank our HOD, project guide and project coordinator for the guidance. And also, we would like to our colleagues.

## REFERENCES

[1]Muhammad Usama Islam ; Hasan Mahmud ; Faisal Bin Ashraf ; Iqbal Hossain ; Md. Kamrul Hasan ”Yoga posture recognition by detecting human joint points in real-time using Microsoft Kinect.” IEEE Region 10 Humanitarian Technology Conference (R10-HTC). pp.1-5, 2017.

[2]Hua-Tsung Chen, Yu-Zhen He, Chunself-trainingChien-Li Chou, SuhYin Lee, Bao-Shuh P. Lin, "Yoga posture recognition for selftraining.” International Conference on Multimedia Modeling. Springer, pp.496-505, 2014.

[3] Xin Jin; Yuan Yao; Qiliang Jiang; Xingying Kinect; Jianyi Zhang; Xiaokun Zhang; Kejun Zhang, “Virtual personal trainer via the kinect sensor” IEEE 16th International Conference on Communication Technology (ICCT). pp.1-6, 2015.

[4] Pullen, Paula, and William Seffens. “Machine learning gesture analysis of yoga for exergame Development.” IET Cyber-Physical Systems: Theory Applications, vol.3, no.2, pp.106-110, 2018.

[5] Trejo, Edwin W., and Peijiang Yuan. ”Recognition of Yoga poses through an interactive system with Kinect device." 2nd International Conference on Robotics and Automation Sciences (ICRAS), 2018.

[6] Kodama, Tomoya, Tomoki Koyama, and Takeshi Saitoh, “Kinect sensor based sign language word recognition by multi-stream HMM.” 201756th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), pp.1-4, 2017.

[7] Caˇlin, Alina Delia.” Variation of pose and gesture recognition accuracy using two Kinect versions.” 2016 International Symposium on Innovations in Intelligent Systems and Applications, pp.1-4, 2016.

[8] Qifei Wang; Gregorij Kurillo; Ferda Ofili; Ruzena Bajcsy; “Evaluation of Pose Tracking Accuracy in the First and Second

Generations of Microsoft Kinect" International conference on healthcare informatics, pp.1-11, 20 15.

[9] Gonzalez-Jorge, H., Diaz Vilarino L., "Metrological Comparison Between Kinect I and Kinect II Sensors." Elsevier Measurement, Vol.70, pp.21=26, 2015.

[10] Khoshelham, Kourosh, and Sander Oude Elberink. "Accuracy and Resolution of Kinect Depth Data for Indoor mapping applications", Sensors, Vol. 12, No.2, pp.1437-1454, 2012.

[11] Yang Lin, Zhang Longyu; Dong Haiwei, "Evaluating and improving the depth accuracy of Kinect for Windows v2." IEEE Sensors Journal, Vol.15, No.8, pp. 4275-4285, 2015

[12] S.K. Yadav, Singh, A. Gupta, J. L. Raheja, "Real-time Yoga Recognition using Deep Learning" Neural Computing and Applications, Vol.31, pp. 9349-9361, 2019.

[13] https://en.wikipedia.org/wiki/Law of cosines.

[14] https://en.wikipedia.org/wiki/Euclidean distance

[15] Yash Agrawal, Yash Shah, Abhishek Sharma," Implementation of Machine Learning Technique for Identification of Yoga Poses" 9th IEEE International Conference on Communication Systems and Network Technologies.