e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:05/May-2022 Impact Factor- 6.752 www.irjmets.com

REAL-TIME YOGA POSE DETECTION

Radha Tawar [*1], Sujata Jagtap[*2], Darshan Hirve[*3], Tejas Gundgal[*4], Dr. Namita Kale[*5]

[*1,2,3,4]Student, Information Technology, MET's Bhujbal Knowledge City, Nashik, Maharashtra, India.

[*5]Professor, Information Technology, MET's Bhujbal Knowledge City, Nashik, Maharashtra, India.

ABSTRACT

Human pose estimation is an arising problem in the computer vision areas that have exposed many challenges and consequences in the past, almost becoming a global issue now. Analyzing human activities is advantageous in many areas 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 detection can be used to build a self-instruction exercise system that allows people to learn and practice exercises comfortably by themselves. This project builds the foundation for building such a system by discussing various machine learning and deep learning approaches to accurately classify yoga poses in real-time. Using the system, the user is provided the choice to select the pose that he/she wishes to practice. The pose of the user is detected and the difference in angles of various body joints is calculated. Based on this difference in angles audio and text feedback are provided to the user so that he/she can improve the pose. Along with audio and text feedback, a live interaction session with an instructor is provided.

Keywords: Classification, Gesture Recognition, Voice Feedback.

I. INTRODUCTION

Yoga exercises boost physical health as well as help to clean the body, mind, and soul. It comprises many asanas and each of them denotes the static physical postures. In this system, the Mediapipe library is used. Using a detector, the pipeline first locates the person/pose region-of-interest (ROI) within the frame. The tracker subsequently predicts the pose keypoints and segmentation mask within the ROI using the ROI-cropped frame as input. Using this project, the user can select the pose that he/she wishes to practice. Based on this difference in angles feedback is provided to the user so that he/she can improve the pose. Thus, everyone can practice Yoga, no matter their age or health. Yoga poses are implemented using OpenCV and mediapipe with much more accuracy. The system is included 5 poses (Warrior 1, Warrior 2, Butterfly, Triangle, Cobra) which give output using audio feedback, and the rest of 4 poses (Lotus, T Pose, Mountain, Tree) with text-to-speech feedback.

II. METHODOLOGY

OpenCV:

OpenCV is a Python open-source library, which is used for computer vision in Artificial intelligence, Machine Learning, face recognition, etc. It works on every frame of the image. It contains library files which is used to calculate angles. It controls camera from which difference between real image and unknown image is known.

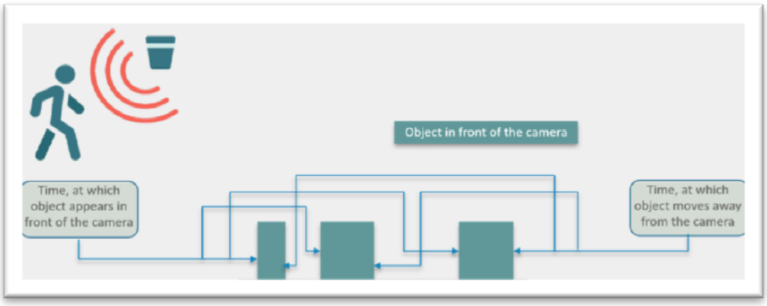

Figure 1: OpenCV Architecture

MediaPipe:

MediaPipe Pose is a 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 mask 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).

Figure 2: MediaPipe Architecture

SAPI(Voice Library):

The SpVoice object brings the text-to-speech (TTS) engine capabilities to applications using SAPI automation. An SpVoice object, usually referred to simply as a voice, is created with default property settings so that it is ready to speak immediately. An SpVoice object is created with its audio output set to the system default audio output. Use the Get Audio Outputs method to determine what other outputs are available to the voice.

III. PROPOSED SYSTEM

The previous yoga pose detection systems have been time-consuming and output given in the form of text displayed on screen which makes it difficult for user to execute and perform the yoga. This work proposes a System to detect correct yoga pose. For finding the trust between users and the system is considered. After evaluation, the output of the process is given in the form of text as well as audio feedback for better understanding and performance.

IV. SYSTEM ARCHITECTURE

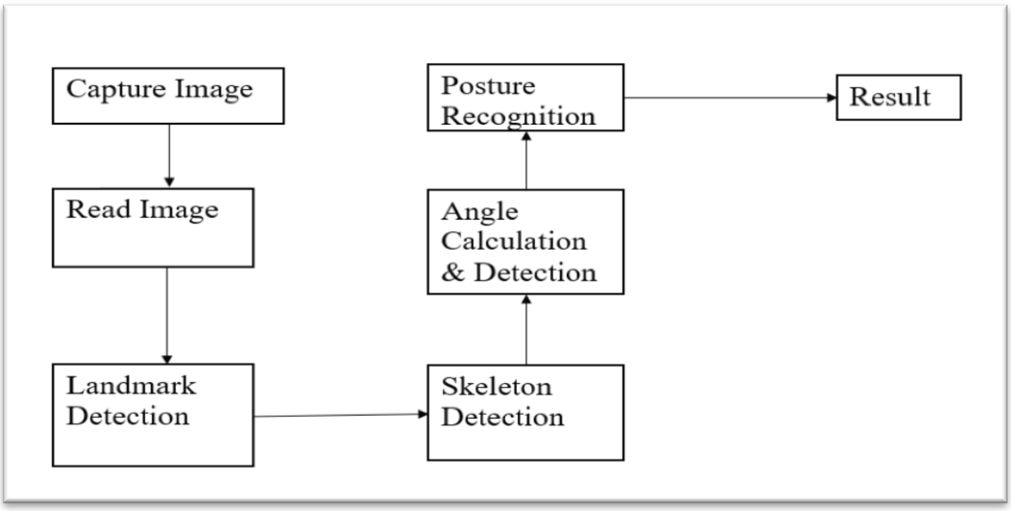

Figure 3: System Architecture.

V. SYSTEM WORKFLOW

- **User Registration**: Registration of new user for practicing yoga.
- **User Login**: Registered users are logged in.
- **Start Button**: Camera gets started.
- **Camera**: It captures images in real-time and reads the images using OpenCV.
- **Keypoints detection**: Keypoints are detected using mediapipe.
- **Skeleton formation**: landmarks are detected and skeleton formed.

IRJMETS
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:05/May-2022 Impact Factor- 6.752 www.irjmets.com

- **Pose Classification with Angle Heuristics**: Classifying different yoga pose using the 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.
- **Text Feedback**: Result is provided to user in text format on the screen It's showing the pose is correct or not.
- **Audio Feedback**: User get instructed with the help of audio feedback

VI. IMPLEMENTATION

1. 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 the person performing the yoga poses. The distance is maintained around 4 to 5m between the camera and the person.

2. 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.

3. 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:

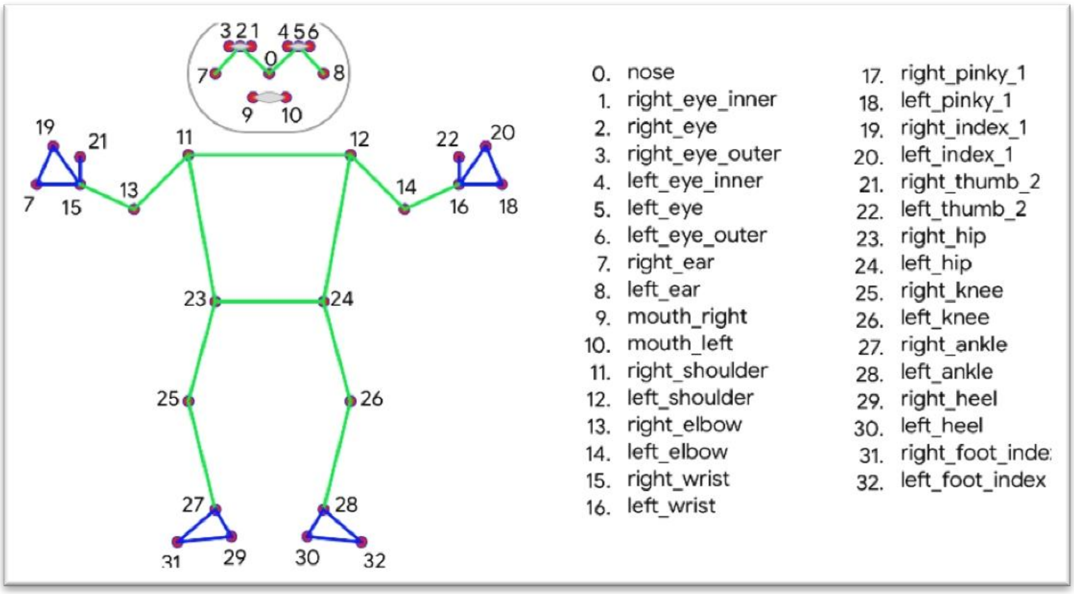

Figure 4: Standing pose skeleton using Mediapipe.

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 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 range [0.0, 1.0] representing the possibility of the landmark being visible (not occluded) in the image. This is a useful when variable want 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.

4. 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:

$$\tan(x + y) = \frac{\tan x + \tan y}{(1 - \tan x * \tan y)}$$

Table 1. Angles Range of Poses

Sr No.	Keypoints	Warrior1	Warrior2	Butterfly
1	Left shoulder -Left elbow -Left wrist.	100°-220°	165°-180°	160°-200°
2	Right shoulder -Right elbow -Right wrist.	100°-220°	165°-180°	160°-200°
3	Left hip -Left knee -Left ankle.	165°-180°	165°-180°	69°-81°
4	Right hip -Right knee -Right ankle.	80°-90°	80°-90°	280°-298°
5	Left ankle -Left heel -Left foot index.	80°-90°	80°-90°	10°-30°
6	Right ankle -Right heel -Right foot index.	165°-180°	165°-180°	10°-30°

5. 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.

6. Feedback

1. Audio Feedback:

The system gives audio feedback to the user to correct their pose accordingly. In the proposed system as mentioned in table 2 for these yoga poses system gives feedback in audio using a manually recorded audio clips.

Table 2. Audio Feedback Poses

Sr No.	Asanas Name
1	Bhujangasana (Cobra Pose)
2	Trikonasana (Triangle Pose)
3	Virabhadrasana1(Warrior1 Pose)
4	Virabhadrasana2(Warrior2 Pose)
5	Bhadrasana (Butterfly Pose)

2. Text to Speech Feedback:

The system gives text to speech feedback to the user to correct their pose accordingly. In the proposed system as mentioned in table 3 for these yoga poses system gives feedback in text to speech using SAPI library.

Table 3. Text-to-speech Feedback Pose

Sr No.	Asanas Name
1	Padmasana (Lotus Pose)

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:05/May-2022 Impact Factor- 6.752 www.irjmets.com

2	Reference Pose (T Pose)
3	Tadasana (Mountain Pose)
4	Vrikshasana (Tree Pose)

VII. RESULTS

Our proposed system provides auditory feedback and instruct the use accordingly. This proposed system architecture precisely detects whether the yoga pose is correct or not. The system includes 5 poses (Warrior 1, Warrior 2, Butterfly, Triangle, Cobra) which give output using audio feedback, and the rest of 4 poses (Lotus, T Pose, Mountain, Tree) with text-to-speech feedback. This web-based application is expected to give much better insights in the future.

The following figs. (5,6,7,8,9,10 & 11) shows the GUI of the proposed system

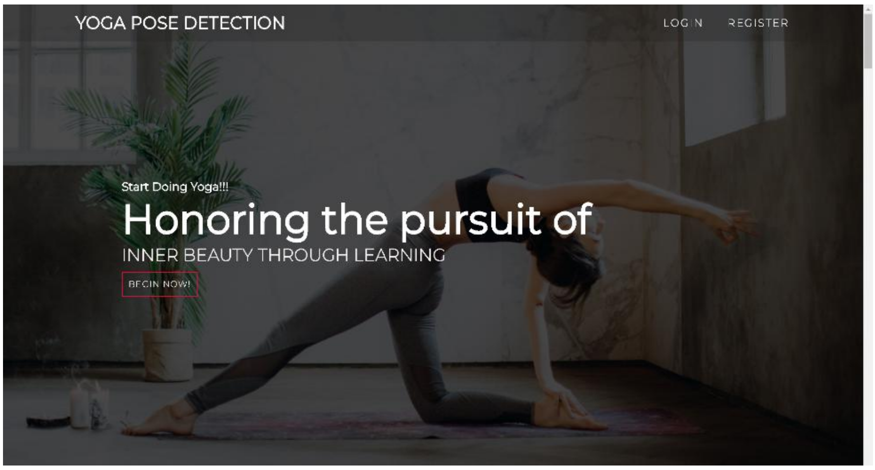

Figure 5: Home Page

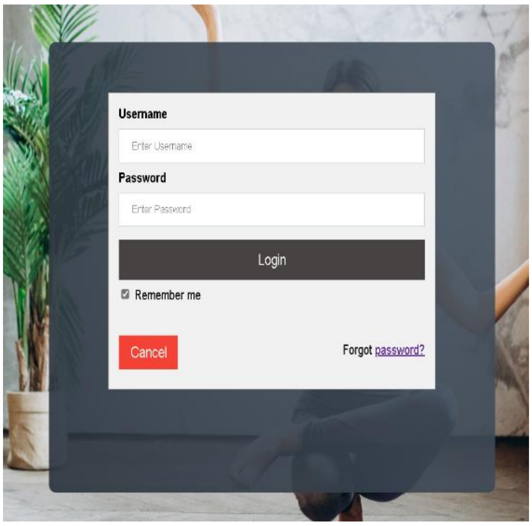

Figure 6: Login Page

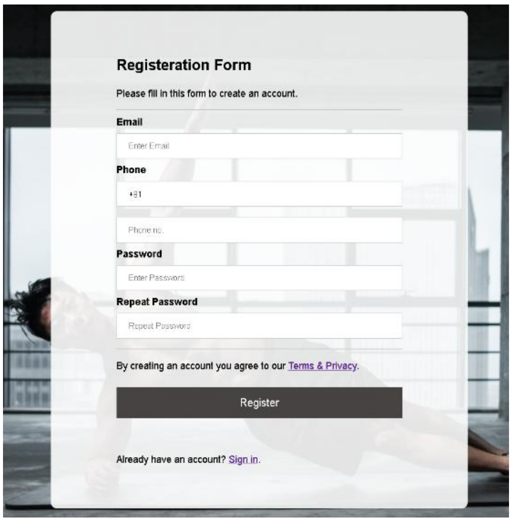

Figure 7: Registration Page

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:05/May-2022 Impact Factor- 6.752 www.irjmets.com

Figure 8: Yoga Information Page

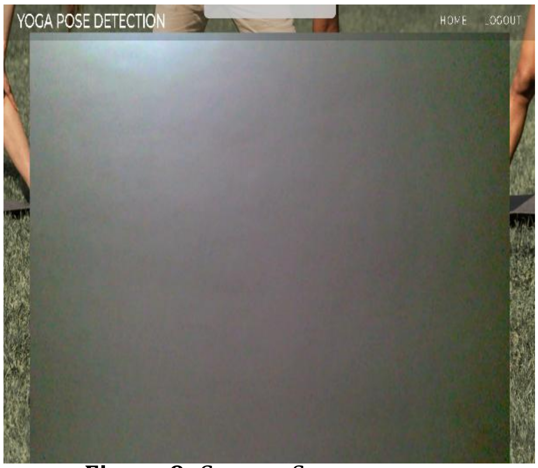

Figure 9: Camera Screen

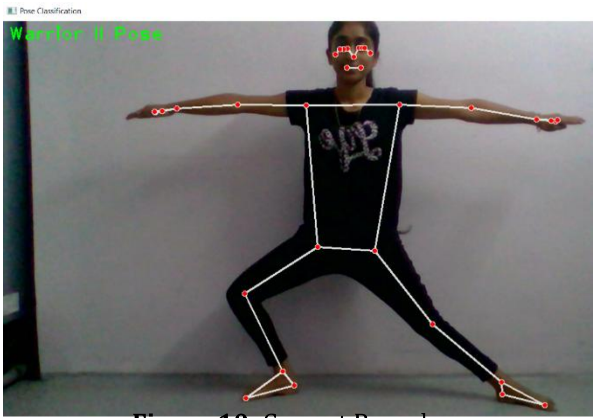

Figure 10: Correct Pose demo

Figure 11: Incorrect Pose demo

VIII. CONCLUSION

In this paper, we proposed a Yoga Pose Detection system using a traditional RGB camera. Yoga poses are implemented using OpenCV and mediapipe with much more accuracy. The system includes 5 poses (Warrior 1, Warrior 2, Butterfly, Triangle, Cobra) which give output using audio feedback, and the rest of 4 poses (Lotus, T Pose, Mountain, Tree) with text-to-speech feedback.

IX. FUTURE WORK

Extra new poses will be added in the system. Personal session will be improvised. Time-complexity of pose detection and audio feedback will be reduced.

X. REFERENCES

[1] Qifei Wang; Gregorij Kurillo; Ferda Ofli; Ruzena Bajcsy; "Evaluation of Pose Tracking Accuracy in the First and Second Generations of Microsoft Kinect" International conference on healthcare informatics,pp.1-11, 2015.

[2] 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.

[3] 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.

[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] Santosh Kumar Yadav, Amitojdeep Singh, Abhishek Gupta, Jagdish Raheja . "Real-time Yoga recognition using deep learning,"Published : 20 May 2019.

[6] Deepak Kumar, Anurag Sinha. "Yoga Pose Detection and Classification Using Deep Learning " Published : 28 Nov 2020. Jharkhand, India
Spinger.

[7] 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. Published: 12 June 2020.