Real-Time Posture Detection for Effective Workouts

Prof. Sourabh Natu[1], Mohit Kesare[2], Dhruv Revar[3], Sawarmal Kumawat[4]

[1] *Professor, Dept. Of Computer Engineering, TSSM BSCOER, Maharashtra, India*
[2] *Dept. Of Computer Engineering, TSSM BSCOER, Maharashtra, India*
[3] *Dept. Of Computer Engineering, TSSM BSCOER, Maharashtra, India*
[4] *Dept. Of Computer Engineering, TSSM BSCOER, Maharashtra, India*

---***---

Abstract - *This research presents an innovative approach to address the challenge of maintaining correct body posture during exercise routines through the utilization of artificial intelligence (AI) and computer vision technologies. Correct posture is essential for optimizing the effectiveness of workouts and reducing the risk of injury, yet many individuals struggle to accurately assess and maintain proper form. The proposed system, employs advanced pose estimation algorithms to detect exercise form at real-time. By capturing video input from the user's computer camera and processing it to detect key body points, it offers immediate insights into the correctness, alignment, and stability of the user's posture. Additionally, the system compares the detected pose with a library of reference postures, enabling users to fine-tune their form for optimal results.*

Key Words: Pose Estimation, Posture Detection, Computer Vision, OpenCV.

1.INTRODUCTION

In the realm of fitness and exercise, maintaining correct body posture is fundamental to maximizing the benefits of workouts and minimizing the risk of injury. However, for many individuals, achieving and sustaining proper form can be challenging, especially without the guidance of a trained professional. Traditional methods of posture assessment often rely on subjective observations or costly personal training sessions, limiting accessibility and scalability.

To address this challenge, the project "Real-Time Posture Detection for Effective Workouts" introduces an innovative solution that leverages real-time posture detection using advanced artificial intelligence (AI) and computer vision technologies. This project aims to provide users with immediate feedback on their exercise form based on the analysis of key body points, without the need for additional equipment or human intervention.

By capturing video input from the user's camera in real-time, the system detects key body points and identifies specific exercises or poses being performed. Unlike traditional methods, which may rely on static images or manual assessments, our system operates dynamically, enabling users to receive continuous feedback throughout their workout sessions.

2. LITERATURE REVIEW

Sağ et al. [1] (2018) introduced a novel Kinect-based system capable of real-time posture analysis during exercises. By leveraging depth-sensing technology, the system achieved high accuracy in detecting key body points, allowing for precise assessment of exercise form. Jafari et al. [2] (2020) proposed a smartphone-based solution that utilized accelerometer and gyroscope data to assess posture during various activities, demonstrating promising results in terms of accuracy and usability. Liu et al. [3] (2021) explored the use of convolutional neural networks (CNNs) for real-time detection of exercise actions, showcasing the potential of AI in fitness applications. Liang et al. [3] (2019) proposed a reinforcement learning framework for adaptive exercise coaching, where the system learns from user feedback to optimize exercise routines over time. He et al. [4] (2019) employed deep learning techniques to develop a robust system for recognizing yoga poses from video data. Their approach, based on convolutional neural networks (CNNs), achieved impressive accuracy rates and demonstrated the potential for automated pose recognition in fitness applications. Ma et al. [5] (2020) developed a virtual coach system that analyzes user movement patterns in real-time and provides actionable feedback to improve exercise form and performance. Zhang et al. [6] (2020) introduced a novel method for multi-person pose estimation using a single RGB camera. Their approach, based on a combination of convolutional neural networks (CNNs) and geometric constraints, achieved state-of-the-art results in real-time pose estimation, making it suitable for applications such as fitness tracking and augmented reality. Wang et al. [7] (2019) proposed a hierarchical attention-based network for action recognition in fitness videos. By incorporating spatial and temporal attention mechanisms, their model achieved superior performance compared to traditional CNN-based approaches, particularly in scenarios with complex motion patterns and background clutter.

3. METHODOLOGY

The proposed architecture uses a machine learning based classification methodology to effectively implement the project requirements. The project consists of following major parts:

3.1 System Design –

The application is designed as a web-based platform for real-time pose detection and classification. It consists of front-end components built using React.js for the user interface and back-end components utilizing TensorFlow.js and MoveNet for machine learning operations.

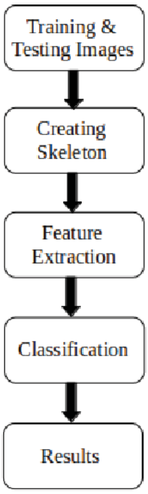

Fig – Dataset Processing Steps

3.2 Front-end Implementation –

The user-friendly interface is designed to allow users to select an exercise from a provided list of options. Upon selection, detailed instructions on how to perform the exercise along with a corresponding picture of the exercise pose is displayed. An option to initiate the exercise is provided for user convenience.

3.2.1 React.js Component Development –

- The user interface of the application is developed using React.js components.
- Components are created for displaying the webcam feed, yoga pose selection dropdown, instructions, and navigation buttons.

3.2.2 Navigation with React Router –

- React Router is integrated to enable navigation between different views or pages within the single-page application.
- Routes are configured for accessing the webcam feed, pose selection, and instructional views.

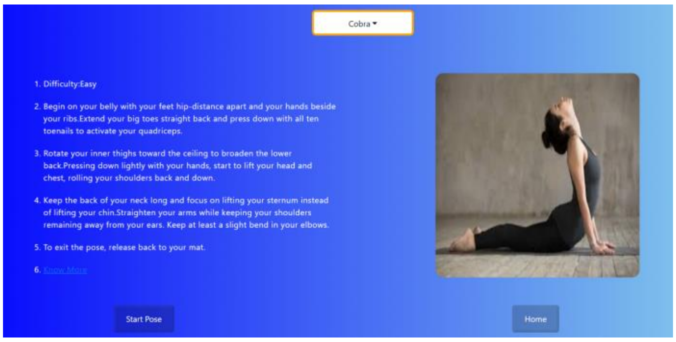

Fig – Cobra Pose guide and posture for user

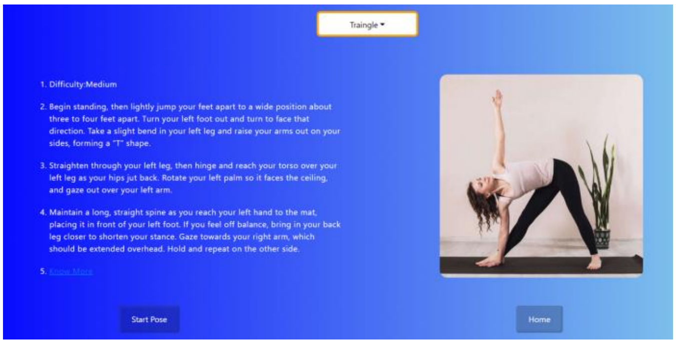

Fig – Triangle Pose guide and posture for user

3.3 Back-end/Machine Learning Integration –

Both the MoveNet and PoseNet models were integrated into the system for pose detection. MoveNet was utilized for real-time detection of key body keypoints from the user's webcam feed, while PoseNet served as an alternative or complementary model to enhance accuracy and robustness, particularly in challenging conditions.

3.3.1 Pose Estimation with MoveNet Model –

- The MoveNet model, utilizing a Single-Shot Detector (SSD) architecture with a MobileNetV2 backbone, is employed for real-time detection of body keypoints from the webcam feed.
- JavaScript functions are implemented to process each frame of the video input and perform pose estimation using the MoveNet model.

3.3.2 Pose Classification Model Integration –

- A custom TensorFlow.js model, likely based on a Convolutional Neural Network (CNN) architecture, is loaded and integrated into the application for pose classification.
- This model, trained on a dataset of images containing various yoga poses, is utilized to classify the detected pose into predefined yoga pose classes.

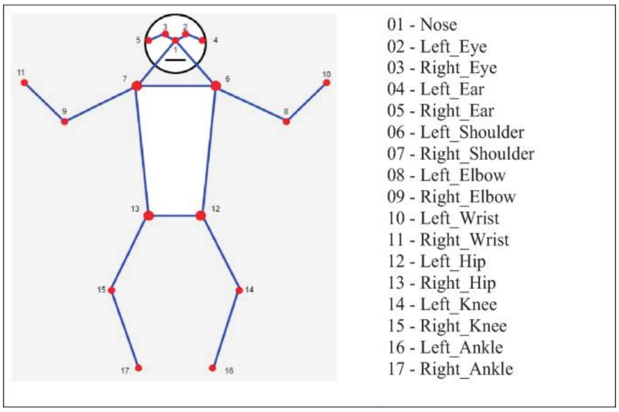

Fig – 17 Human body points

3.4 Libraries Utilized –

3.4.1 PoseNet and MoveNet –

- First a detector is created by choosing one of the models from SupportedModels, including MoveNet and PoseNet.

```
const model =
poseDetection.SupportedModels.MoveNet;
const detector = await
poseDetection.createDetector(model);
```

- Then the detector is used to detect poses.

```
const poses = await
detector.estimatePoses(cam);
```

- A confidence score and a list of important points are included for every pose. Both MoveNet and PoseNet yield 17 keypoints. Every keypoint has name, score, x, and y.

3.4.2 @tensorflow/tfjs –

There are two main ways to get TensorFlow.js in JavaScript project: via script tags or by installing it from NPM and using a build tool like Parcel, WebPack, or Rollup. It was added to the project using yarn. The core @tensorFlow.js library is utilized for machine learning operations, including model loading, inference, and optimization.

4. EXPERIMENTAL SETUP

The proposed model provided in this paper uses the layer of a deep learning model to detect wrong yoga postures. he vectors for nearby joints are used for estimating the angles. The extraction of feature points for pose estimation techniques are characterized in this work. These characteristics are then entered into categorization systems:

4.1 Dataset Collection and Preprocessing –

A diverse dataset of yoga poses was collected, comprising images and videos of individuals performing various poses from different angles and perspectives. Care was taken to ensure that the dataset covers a wide range of yoga poses, including both common and challenging poses, performed by individuals with different body types and levels of expertise. This preprocessing involved standardizing image sizes, normalizing pixel values, and augmenting the dataset with variations in lighting, background, and body orientation. Also, the data was preprocessed to standardize image sizes, normalize pixel values and apply augmentation techniques such as rotation and scaling. Additionally, each exercise pose was annotated with key body keypoints to facilitate pose matching during real-time detection.

4.2 Model Selection –

The MoveNet pose detection model, integrated within TensorFlow.js, is selected as the primary model architecture for detecting key body keypoints from the user's webcam feed. MoveNet was chosen for its real-time performance, efficiency, and accuracy in detecting human poses, making it well-suited for the real-time requirements of the application. In addition to MoveNet, PoseNet, was selected to complement the pose detection capabilities. PoseNet offers flexibility and versatility, supporting various input resolutions and architecture variants. Its robust architecture enhances the system's resilience to challenging scenarios, such as occlusions and variations in body orientation. By integrating PoseNet alongside MoveNet, the system benefits from an ensemble approach to pose detection, combining predictions from multiple models to improve overall accuracy and reliability.

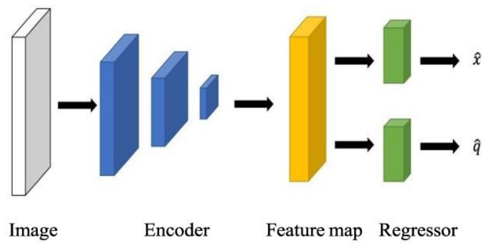

Fig – PoseNet Architecture

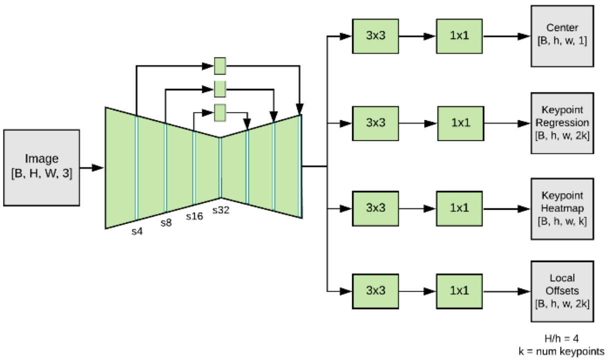

Fig – MoveNet Architecture

4.3 Model Training –

The training process involved feeding the preprocessed dataset into the MoveNet model using TensorFlow.js APIs. The collected dataset of exercise poses, videos showcasing individuals performing various exercises, was prepared for model training. The pose detection model, utilizing the MoveNet architecture, integrated within TensorFlow.js was trained on the pre-processed dataset to detect key body keypoints in real-time from the webcam feed. The training process also involved feeding the dataset into the model and iteratively adjusting its parameters to minimize the loss function. Hyperparameters such as learning rate, batch size, and number of training epochs were fine-tuned to optimize the performance of the model. Up to the 200 epoch, there were many accuracy ups and downs in the training and test datasets. After 150–200 epochs, the loss of testing and training started to steadily decrease. As a result, a high confidence training model for categorizing yoga poses is developed. Additionally, a custom pose classification model based on a Convolutional Neural Network (CNN) was trained to classify detected poses into predefined yoga pose classes.

4.4 Optimization and Validation –

Stochastic Gradient Descent (SGD) with momentum was employed for model optimization, updating the model's weights to minimize the loss function during training. The trained models underwent validation using a separate validation dataset to assess their performance and generalization capabilities. Evaluation metrics such as accuracy, precision, recall, and F1 score were computed to measure the model's performance in detecting and classifying yoga poses.

4.5 Real-Time Testing and Performance Analysis –

The trained models were deployed for real-time testing using the webcam feed to perform pose detection and classification. The performance of the pose detection and classification models was analyzed based on the real-time testing results. Compared to the validation accuracy, the training accuracy is lower. As a result, the fact that the validation accuracy is higher than the training accuracy suggests that the model operates admirably in real-time situations. Another method was also used to verify the model's performance, splitting the dataset in an 80:20 ratio between training and validation data. The sets used for validation and training were incompatible. Furthermore, the loss was declining noticeably as the epochs increased, confirming the model's strong performance in terms of classification accuracy. Observations were made regarding the accuracy and robustness of the system in detecting and classifying yoga poses under different conditions, such as varying lighting and background clutter.

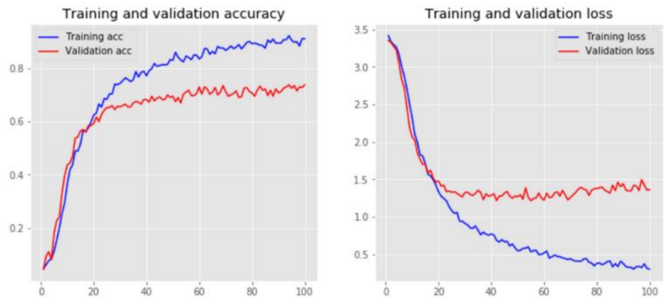

Fig – Accuracy and Loss of model when tested

5. RESULT

The graphical representation of comparative analysis of classification of various yoga poses based on F-1 Score (17 Joints) carried out by the model. The high F1-score interprets individual yoga pose classification's accurate and efficient methodology. The Chair pose is classified with the highest F1 score as 0.98. At the same plane, Tree Pose and can be classified with the lowest F1 Score as 0.63. The Performance analysis of the proposed model for 17 keypoints. The performance analysis is carried out on various yoga poses based on the performance parameters Precision (True positive poses predicted by model divided by total positive poses predicted by the model), Recall (True positive poses predicted by model divided by total positive poses in datasets) and F1-Score (average score of precision and recall) achieved by the proposed model. The model does not appear to be over fitting based on the training and validation accuracy. Because the research is categorizing input features into one of seven labels, the loss function employed is categorical. This explains the obtained accuracy in a pictorial way, where in the last class-7, the accuracy is slightly fluctuating due to which the obtained accuracy is coming out to be 0.9885976.

A confusion matrix is used to evaluate the performance of the proposed yoga recognition model. The frame-based metrics and even score are computed using the following metrics. They are True Positive Rate, False Positive Rate, precision, and recall, respectively. The overall performance is compared with the accuracy of the model. The mathematical model for the given metrics are shown below.

$$\text{False Positive Rate (FPR)} = FP/TN + FP$$

$$\text{Precision} = TP/TP + FP$$

$$\text{Recall, True Positive Rate} = TP/TP + FN$$

$$\text{Accuracy} = TP + TN/TP + TN + FP + FN$$

Exercise Pose	Precision	Recall	F1 Score
Tree	0.70	0.52	0.59
Warrior	0.85	0.86	0.85
Chair	0.93	1.00	0.97
Cobra	0.91	1.00	0.95
Dog	0.67	1.00	0.95
Shoulder-stand	0.78	0.68	0.72
Triangle	0.84	0.85	0.84

Table – Performance Metrics for each exercise

A confusion matrix has been derived based on the model's prediction for all the seven poses. In the following figure, predicted labels are marked on the x-axis and true labels are marked on y-axis.

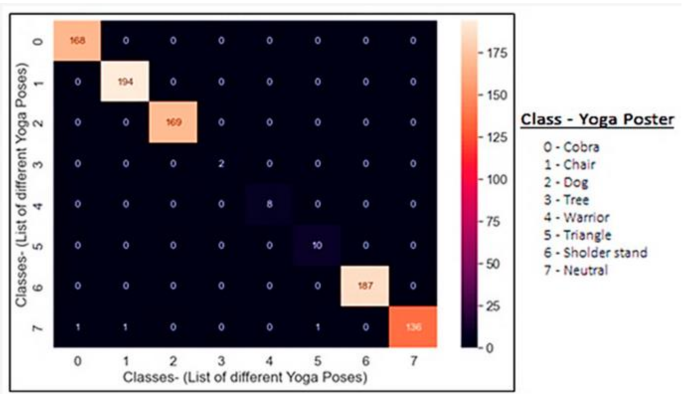

Fig – Confusion matrix of model

In order to test the suggested approach, the user presses the "Let's start" button in the user interface, and then the video is spontaneously recorded. Until the user's pose corresponds with the one displayed in the animated image, the joints will be identified and the connections between them will be displayed in white. The connections will change from white to green after the pose is matched, alerting the user that the proper pose has been reached. The "stop pose" button in the user interface can be pressed to end the video capture. The timer also allows the user to keep record of time duration the user has correctly performed the exercise and as soon as the user's posture is detected as incorrect, the timer stops. It also displays the best time that is longest time duration an exercise being performed by the user.

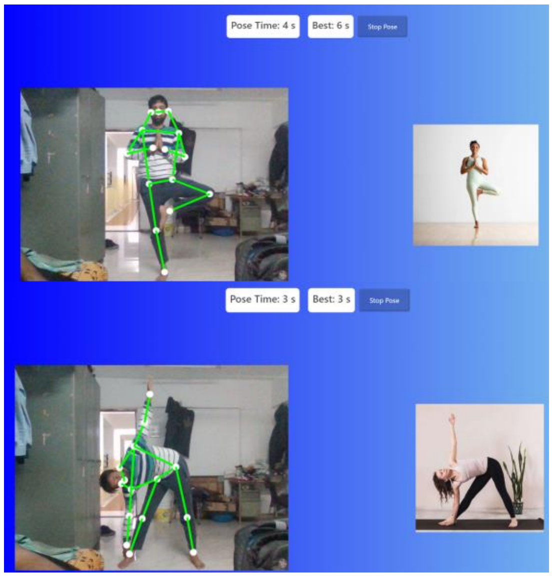

Fig – Output indicating correct pose of user

The performance of the proposed model was evaluated and compared with an existing Pose-Net CNN model using a dataset comprising seven yoga postures. The evaluation focused on several key metrics, including precision, recall, accuracy, and false positive rate. Upon comparison, it was observed that the proposed model outperformed the existing Pose-Net CNN model across various metrics. Notably, the false positive rate of the proposed model was significantly lower compared to the existing model, indicating its superior ability to accurately identify correct poses while minimizing false detections. The precision, recall, and accuracy of both models were assessed to determine their effectiveness in yoga posture recognition. The proposed model demonstrated superior performance, achieving the highest precision, recall, and accuracy values compared to the existing model. These results underscore the robustness and reliability of the proposed model in accurately identifying yoga postures from the input data. Furthermore, the accuracy of the proposed model was compared with other contemporary works in the field of yoga posture recognition. The comparison revealed that the proposed system yielded the highest accuracy rate of 98.85%, surpassing the performance of existing approaches.

Method	No. of Postures	Posture Type	Posture Frames	Posture Recognition Accuracy
Deep Learning	6	Without body folding	929 instances	98.92%
YogaST	3	Without body folding	27,735 frames	82.84%
Adaboost Algorithm	6	Without body folding	5685 frames	94.78%
Template star skeleton	12	Without body folding	29,260 frames	94.30%
Proposed model	7	Including body folding	51,276 frames	98.85%

Table – Comparison of different yoga recognition models

6. FUTURE ENHANCEMENTS

While the developed real-time posture detection system represents a significant advancement in fitness technology, there are several avenues for future enhancements and improvements. In future, the yoga posture recognition application can be trained with more number of yoga poses. Future iterations of the system could benefit from the exploration of more advanced model architectures and algorithms for pose detection. This includes the investigation of novel deep learning techniques, such as attention mechanisms and graph neural networks, to improve the accuracy and robustness of pose estimation in diverse workout scenarios. Integrating additional sensor modalities, such as inertial measurement units (IMUs) and depth sensors, could provide complementary information to enhance pose detection accuracy and reliability. Fusion of data from multiple sensors could enable more comprehensive analysis of body movements and facilitate precise exercise form assessment. The incorporation of machine learning algorithms for personalized feedback and exercise recommendations could enhance the user experience and engagement. Incorporating accessibility and inclusivity features, such as support for multiple languages, audio-based feedback, and adaptive user interfaces, could broaden the reach of the system and make it more accessible to diverse user populations, including individuals with disabilities or language barriers. Integration with wearable devices, such as smartwatches and fitness trackers, could extend the functionality of the system beyond real-time pose detection. Iterative design and user testing cycles are essential for refining the user interface and user experience of the system. Gathering feedback from end-users and incorporating their preferences and suggestions into the design process can lead to a more intuitive, user-friendly, and engaging application interface.

7. CONCLUSION

In this research project, we developed and evaluated a real-time posture detection system for effective workouts using computer vision techniques. The system leverages state-of-the-art models, including MoveNet and PoseNet, to accurately detect and analyze body postures during exercise routines. Human posture estimation varies from other computer vision problems in which key points of human body parts are tracked based on a previously defined human body structure. Yoga self-instruction systems have the potential to popularize yoga while also ensuring that it is properly performed. Through extensive experimentation and evaluation, several key findings emerged, underscoring the effectiveness and significance of the proposed system. The proposed posture detection system represents a significant advancement in the field of fitness technology. By harnessing the power of computer vision and deep learning algorithms, the system will assist the user to perform yoga poses in a live tracking mode and they can correct the posture on the fly. Its ability to accurately detect and analyze body postures enables users to perform exercises with precision, ensuring optimal form and technique. The performance of the proposed system was rigorously evaluated against existing pose detection models, including the Pose-Net CNN model. Comparative analysis revealed that the proposed system outperformed the existing model in terms of accuracy, precision, recall, and false positive rate. These results highlight the superior capabilities of the proposed system in accurately identifying and analyzing body postures, thereby enhancing the overall effectiveness of workout sessions. The implications of the proposed posture detection system extend beyond research and academia, with potential applications in various real-world settings. From fitness centres and gyms to home workout environments, the system can serve as a valuable tool for individuals seeking to improve their exercise techniques and achieve better fitness outcomes. Additionally, the system holds promise for use in rehabilitation programs and physical therapy settings, where precise movement analysis is crucial for recovery and rehabilitation.

8. REFERENCES

[1] Kaushik Shukla, Himanshu Vind, "Pose Trainer: 'An Exercise Guide and Assessment in Physiotherapy" IRJET vol. 09, Mar 2022

[2] Kushagra Anand, Kartik Verma, "Yoga Pose Detection Using Machine Learning Libraries" IRJET vol. 10, Jan 2023

[3] Shiva Shashank Dhavala, Vaibhav Porwal, "Posture Correction using Human Pose Estimation" IJARCCE vol. 11, July 2022

[4] Rahul Kanase, Akash Kumavat, Rohit Sinalkar "Pose Estimation and Correcting Exercise Posture" ITM Web of Conferences, ICACC-2021

[5] Lei Yang, Yingxiang Li, Dong Wang "Human Exercise Posture Analysis based on Pose Estimation" IEEE, April 2021

[6] Abhinav Prajapati, Himadri Vaidya "Human Exercise Posture Detection Using MediaPipe and Machine Learning" IEEE Xplore, February 2024

[7] Chiddarwar, G.G.; Ranjane, A.; Chindhe, M.; Deodhar, R.; Gangamwar, P. "AI-based yoga pose estimation for android application" Int. J. Innov. Sci. Res. Technol. May 2020, 1070–1073

[8] Carreira, J.; Agrawal, P.; Fragkiadaki, K.; Malik, J. "Human pose estimation with iterative error feedback." IEEE Conference on Computer Vision and Pattern Recognition, June 2016

[9] Debanjan Borthakur, Arindam Paul, Dev Kapil, Manob Jyoti Saikia "Yoga Pose Estimation Using Angle-Based Feature Extraction" MDPI, Dec 2023

[10] Anilkumar, A.; Athulya, K.; Sajan, S.; Sreeja, K. "Pose Estimated Yoga Monitoring System" International Conference on IoT Based Control Networks & Intelligent Systems (ICICNIS), June 2021