Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Residual Transformer Framework for Biomechanics-Aware Yoga Posture Recognition and Feedback

Mrinal Kanti Nath

MrinalKantiNath999@gmail.com

TMSL

Tapan Chowdhury

TMSL

Nilay Dhar

TMSL

Hiranmoy Sarkar

TMSL

Debanjan Bunia

TMSL

Arman Munshi

Techno Main Salt Lake

Research Article

Keywords: Yoga Pose Recognition, Human Pose Estimation, Transformer Networks, Pose Correction, Feedback

Posted Date: September 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10950143/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

A Residual Transformer Framework for Biomechanics-Aware Yoga Posture Recognition and Feedback

Mrinal Kanti Nath[1*], Tapan Chowdhury[1], Nilay Dhar[1], Hiranmoy Sarkar[1], Debanjan Bunia[1], Arman Munshi[1]

[1*]Department of Computer Science and Engineering, Techno Main Salt Lake, EM4/1, Sector V, Salt Lake, Kolkata, 700091, West Bengal, India.

*Corresponding author(s). E-mail(s): mrinalkantinath21@gmail.com;

Abstract

Correct yoga posture execution ensures optimal effect from the therapy and excludes musculoskeletal injuries resulting from incorrect performance. The vast majority of approaches for yoga postures analysis that use computer vision algorithms classify postures statically using rule-based models, which are not generic enough owing to inability to deal with complicated dynamic movements and give specific instructions on how to fix postures. This paper introduces the Residual Biomechanical Transformer Network (RBTNet), a deep learning-based approach for temporal classification of yoga postures and biomechanical deviations detection. RBTNet utilizes YOLOv8m-Pose for 17 keypoints detection, 74-dimensions biomechanical features extraction, sliding window temporal sequence construction (window length = 30), Transformer Encoder with multi-head self-attention, attention pooling mechanism and cascaded residual MLP blocks for feature processing. Deviations of joint angles in specific postures serve as the source for corrective feedback generation without a need of user-profile database or additional information about age and flexibility. The proposed method was tested using an eight-class yoga posture dataset (8 classes: chair, cobra, dog, no pose, shoulder stand, triangle, tree, warrior). With the help of early stopping strategy, RBTNet achieved 98% accuracy with a weighted F1-score of 0.98% in just 21 epochs.

Keywords: Yoga Pose Recognition, Human Pose Estimation,Transformer Networks, Pose Correction, Feedback

1 Introduction

The practice of yoga originated in India and has been exercised by people over generations as a total process of improving one's physical and psychological health and improving the standard of living altogether. It consists of breathing control, body positions and meditation which include pranayama and asanas. The end result of this process is improvement in psychological, physiological and spiritual health. The exercise has gained global popularity as an exercise routine for increasing body flexibility, strengthening of muscles, balance and cardiorespiratory endurance. Furthermore, yoga helps greatly in lowering stress and anxiety levels and therefore prevents various diseases. Studies have shown how effective yoga is in managing diseases such as musculoskeletal, respiratory and psychological [1]. It should be noted that body position is very important for yoga success [2].

However, even though there are many benefits associated with yoga practice, performing yoga without correct guidance poses certain dangers. Inappropriate alignment of postures can cause muscle strain, ligament injury, joint stress, and other musculoskeletal problems [3]. These situations arise very often due to the fact that most yoga enthusiasts learn yoga online and lack professional assistance. Any discrepancy in joint angle, body posture, or balance can affect not only safety but also the overall effect of practice. Thus, there is an increasing necessity for intelligent systems that will track and adjust human pose automatically.

With the help of recent developments in AI and Computer Vision technologies, scientists managed to create algorithms which allow estimating human pose via detection of key points on image or video frame. Methods based on deep learning, including convolutional neural network models and OpenPose, MediaPipe, and MoveNet frameworks, demonstrated impressive results in real-time estimation of human pose [4].

Despite the above improvements been introduced, limitations in existing techniques. To begin with, numerous techniques utilize static frame-by-frame analysis of yoga postures by disregarding the dynamic aspect and temporal connections among postures. Consequently, such yoga pose detection models are unable to model the temporality aspect in yoga practices. Secondly, current yoga pose correction systems depend on rule-based or geometric approaches, making it difficult for them to generalize to other yoga postures. Thirdly, majority of yoga pose detection algorithms focus on pose recognition instead of providing guidance for corrective purposes. Lastly, most yoga pose detection systems focus primarily on pose recognition rather than providing pose-specific corrective feedback [5].

To overcome these issues, this paper proposes RBTNet — a Residual Biomechanical Transformer Network — that processes sequences of biomechanically engineered feature vectors through a Transformer Encoder, aggregates temporal information via learned Attention Pooling, and refines representations through stacked Residual MLP

Blocks before classification. Corrective feedback is generated through rule-guided joint-angle deviation analysis rather than a learned correction sub-network, accurately reflecting the implemented system.

The rest of this paper is organized as follows. Section 2 outlines previous work done on yoga posture detection, pose estimation, temporal modeling, and post-correction mechanisms. The proposed framework is introduced in Section 3 and includes details about the system structure, feature extraction, classification model, and feedback. Experimental setup and comparison results are provided in Section 4, followed by conclusions in Section 5.

2 Related Work

In this rapid development of Artificial Intelligence (AI) and Computer Vision technology has a tremendous impact on the progress made in developing automatic systems to detect and analyze human pose information, especially in areas such as yoga. In this section, current research will be critically reviewed based on four broad categories: pose estimation techniques, posture classification models, temporal modelling approaches, and posture correction methods. Human Pose Estimation (HPE) acts as the primary basis for almost all yoga pose recognition systems. Current state-of-the-art HPE architectures rely on deep learning algorithms to detect keypoints within images or videos in order to accurately represent the motion of the human body. Specifically, OpenPose and MediaPipe are two such solutions gaining traction thanks to their capacity to detect multiple joints in real time [5, 6]. MediaPipe Pose represents a compact and fast system allowing for real-time extraction of both 2D and 3D keypoints based on color input [6]. Yet, recent studies have shown that while 2D pose estimation provides high precision and consistency, 3D reconstruction could lead to distortion and asymmetry at larger model capacities [6]. Hence, a solution capable of compensating for such errors is required to achieve accurate pose estimation.

In particular, many researchers have studied the possibility of classifying yoga poses via the application of various machine learning methods and deep learning. Nayak et al. [8] performed an extensive comparative analysis by implementing several models, including CNN, DNN, LSTM, Random Forest, and Logistic Regression. According to the authors, the application of DNN was able to yield the best results in terms of accuracy at 96.76% using BlazePose keypoints. Nevertheless, this study has shed light on how different models perform in postural analysis, but it considers a static problem with the use of threshold correction. Zhao et al. [9] also addressed the issue of pose classification. In particular, these authors have suggested a CNN-LSTM-based model for recognizing Pilates exercises using MediaPipe keypoints. The implementation of a neural network capable of capturing temporal information allowed increasing accuracy to 95.7%. This example proves the value of applying LSTM for solving classification problems in the field under study; however, such solutions remain limited to mere classification. At the same time, researchers have worked on lightweight systems to be used in portable applications. In particular, Zaghloul et al. [10] introduced

a smartphone-based system for recognizing human postures by combining landmarks and machine learning techniques. Although effective for accessibility, such systems often compromise on accuracy and lack advanced reasoning capabilities required for corrective feedback.

Yoga exercises performed in natural settings inherently involve transitions between poses, thus making temporal modeling an essential aspect to achieve accuracy. Frame-by-frame classification algorithms do not consider temporal dependencies while making predictions. As a result, researchers have shifted focus on using recurrent neural networks, especially Long Short-Term Memory Networks, owing to their effectiveness in processing sequential data [9]. Although Long Short-Term Memory Network-based architectures boost performance in light of temporal modeling, they fail to capture long-range dependencies in complex datasets. This creates difficulties when evaluating yoga postures that involve transitions as well as sustaining poses for longer periods. Recent advancements in sequence modeling suggest that Transformer networks deliver superior results since they can model global dependencies compared to other architectures. However, Transformer networks have not been investigated within the context of yoga pose evaluation.

However, apart from the above, numerous models that include methods for posture correction have been developed as well. For example, Gurav et al. [4] presented a MoveNet-based method in which joint angles could be computed and instant feedback provided through cosine similarity computation. The algorithm provides instant feedback; however, this model has been criticized owing to its reliance on pre-defined geometric rules. In contrast, Ko et al. [11] designed a model that provides real-time posture correction in powerlifting exercises using YOLOv5 and MediaPipe models. It also includes feedback depending on the movement phase, demonstrating the significance of contextual feedback in the process of posture correction. However, it is limited to a particular domain, and the feedback generation method used is rule-based. The application of deep neural network algorithms in order to achieve this goal was utilized in the research conducted by Haldar [2], who attempted to detect and correct postures using CNN-LSTM models. Scoring has been used to compute deviation from the proper joint angles.

An in-depth review like that of Rajendran and Sethuraman [1] will provide extensive coverage on the current yoga posture recognition systems and how the AI tools have helped make the process more accessible and easier to monitor. An important fact that emerges from the review is that correct posture is required for achieving any therapeutic benefits while also protecting against injury; however, most available methods fail because they lack the ability to monitor the user in real time. Furthermore, comparisons reveal that while existing recognition methods aim for accurate classification, they ignore crucial aspects of usability such as feedback and customization [8]. Another issue is that there is no relation between pose detection, timing, and feedback generation.

Although many studies have been conducted on yoga posture analysis using AI techniques, there are several challenges in current systems that need to be addressed. First, most yoga posture analysis systems use a frame-based approach for classification that does not account for the temporal dynamics involved in performing yoga postures. Second, current correction strategies in yoga posture analysis are primarily based on rules or geometry, which may not be adaptable to different physical forms or levels of skill among users. Moreover, current yoga posture analysis frameworks cannot tailor their outputs based on individual users' needs. The main objective of this paper is to design a Transformer-based hybrid framework for learning-driven yoga posture correction that takes into account the temporal dynamics of yoga flows.

To address these challenges, this paper proposes a novel hybrid architecture RBTNet — a Residual Biomechanical Transformer Network. The principal contributions of this study include the development of a 74-dimensional biomechanical feature representation for detailed posture characterization, a sliding-window temporal sequence generation mechanism for capturing motion dynamics, along with integrating a Transformer Encoder, Attention Pooling, and Residual MLP blocks for yoga posture classification. The framework also incorporates a biomechanical deviation analysis module to provide interpretable posture-correction feedback based on anatomical alignment rules. Experimental evaluation on eight yoga posture classes demonstrates the effectiveness of the proposed approach, achieving 98% test accuracy with convergence in 21 epochs.

3 Proposed Framework

Based on the research gaps identified in Section 2, this paper proposes an intelligent hybrid framework for real-time yoga posture recognition and biomechanical posture assessment. Unlike conventional vision-based yoga recognition systems that primarily focus on posture classification, the proposed framework integrates computer vision, biomechanical feature engineering, temporal sequence modeling, and rule-guided posture evaluation to simultaneously recognize yoga postures and generate interpretable corrective feedback. The framework models yoga practice as a continuous temporal activity rather than a collection of independent static images, enabling the system to capture both spatial body configurations and temporal motion characteristics during posture execution. The proposed framework combines the representation learning capability of deep neural networks with the interpretability of biomechanical knowledge. Human skeletal keypoints extracted from a real-time pose estimation model are transformed into compact and invariant biomechanical descriptors that effectively characterize body posture while remaining independent of camera position, body scale, and subject height. These structured features are subsequently processed by the proposed Residual Biomechanical Transformer Network (RBTNet), which learns discriminative temporal representations for yoga posture recognition. Finally, the predicted posture is evaluated using predefined biomechanical constraints to identify joint-level deviations and generate deterministic corrective feedback for posture improvement.

The complete framework consists of six sequential stages:

1. Human pose acquisition using YOLOv8m-Pose.
2. Biomechanical feature extraction from detected skeletal keypoints.
3. Feature standardization and data preprocessing.
4. Temporal sequence construction using a sliding-window strategy.
5. Temporal posture classification using the proposed RBTNet architecture.
6. Rule-guided biomechanical deviation analysis and corrective feedback generation.

The integration of these components enables the proposed framework to achieve accurate yoga posture recognition while simultaneously providing interpretable posture correction, making it suitable for near real-time intelligent yoga assistance.

3.1 Overall System Architecture and Working Principle

The proposed framework is designed as an end-to-end intelligent pipeline for yoga posture recognition and biomechanical posture assessment by integrating computer vision, biomechanics, and deep learning. The framework transforms raw visual information into structured biomechanical representations before performing temporal posture classification and rule-guided posture evaluation. Unlike conventional image-based recognition methods that rely solely on pixel-level information, the proposed framework operates on human skeletal representations, which provide improved robustness against variations in body size, camera viewpoint, illumination, and background conditions. Similar skeleton-based representations have demonstrated superior generalization capability for human posture understanding in recent studies [1, 2, 8, 9].

As illustrated in Fig. 1, the overall framework follows a multi-stage processing pipeline consisting of pose acquisition, biomechanical feature engineering, feature preprocessing, temporal sequence construction, Transformer-based posture classification, biomechanical deviation analysis, and corrective feedback generation. Each module performs a specific task while preserving the structural information required by the subsequent stages.

For an input image or video frame I, the proposed framework operates according to the following sequence:

$$I \rightarrow K \rightarrow F \rightarrow F' \rightarrow S \rightarrow \hat{y} \rightarrow D \rightarrow C \tag{1}$$

where

- I denotes the input RGB image or video frame;
- K represents the detected human skeletal keypoints extracted by YOLOv8m-Pose;
- F denotes the extracted 74-dimensional biomechanical feature vector;
- F' represents the standardized feature vector after normalization;
- S denotes the temporal feature sequence generated using a sliding-window strategy;
- $\hat{y}$ represents the predicted yoga posture class obtained from the proposed RBTNet classifier;
- D denotes the biomechanical deviation analysis performed using predefined joint-angle constraints corresponding to the predicted posture;
- C represents the final corrective feedback generated for the user.

Initially, the input frame is processed using YOLOv8m-Pose to estimate seventeen anatomical body keypoints. The detected skeleton is subsequently transformed into a compact biomechanical representation composed of normalized joint coordinates, angular relationships, proportional distance measurements, and confidence information. The resulting feature vector is standardized and organized into fixed-length temporal sequences before being processed by the proposed Transformer-based RBTNet architecture. The predicted posture class is then evaluated using a deterministic rule-based biomechanical analysis module that compares measured joint angles with predefined anatomical reference ranges for the corresponding yoga posture. Whenever a joint deviates from its acceptable biomechanical range, the framework generates pose-specific corrective instructions that assist the practitioner in improving posture alignment.

By combining robust skeletal feature representation, temporal sequence learning, and interpretable biomechanical reasoning, the proposed framework simultaneously achieves accurate posture recognition and explainable posture correction while maintaining computational efficiency suitable for near real-time deployment.

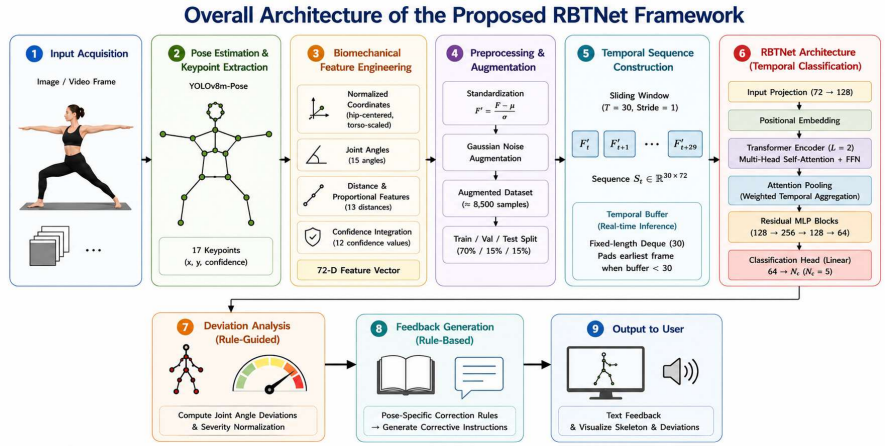


Fig. 1 Overall architecture of the proposed RBTNet framework.

3.2 Pose Acquisition and Key Point Extraction

The first stage of the proposed framework performs human pose acquisition using the YOLOv8m-Pose model, which provides a robust and computationally efficient solution for real-time human pose estimation [5]. Given an input RGB image or video frame, the detector simultaneously localizes the human subject and estimates the positions of the anatomical body joints. Similar to widely adopted pose estimation frameworks such as OpenPose and MediaPipe, YOLOv8m-Pose predicts the standard 17 COCO

body keypoints that describe the human skeletal structure [6]. These keypoints provide a compact structural representation of the body while eliminating unnecessary appearance information such as clothing, illumination, and background variations.

The detected human pose is represented as

$$K = \{(x_i, y_i, c_i)\}_{i=1}^{N}, \tag{2}$$

where x_i and y_i denote the two-dimensional image coordinates of the i^{th} body keypoint, c_i represents the corresponding detection confidence score, and $N = 17$ denotes the total number of anatomical keypoints estimated by the pose detector.

Since inaccurate keypoint localization can negatively influence subsequent feature extraction and posture classification, a confidence-based filtering strategy is incorporated before feature generation. When multiple persons are detected within the same frame, the bounding box with the highest detection confidence is selected for further processing. Subsequently, the average confidence score of four anatomically stable reference joints, namely the left shoulder, right shoulder, left hip, and right hip, is computed as

$$\bar{c} = \frac{c_{\text{LS}} + c_{\text{RS}} + c_{\text{LH}} + c_{\text{RH}}}{4}, \tag{3}$$

where c_{LS}, c_{RS}, c_{LH}, and c_{RH} denote the confidence scores of the left shoulder, right shoulder, left hip, and right hip, respectively.

Frames satisfying

$$\bar{c} < 0.30 \tag{4}$$

are discarded prior to feature extraction, whereas frames with confidence scores greater than or equal to the threshold are retained for further processing. This filtering procedure effectively suppresses unreliable pose estimations while preserving high-quality skeletal representations for biomechanical analysis. As demonstrated in the experimental evaluation, only a very small number of frames were rejected, indicating the robustness of YOLOv8m-Pose on the selected yoga posture dataset.

3.3 Biomechanical Feature Engineering

Although skeletal keypoints accurately describe the spatial locations of body joints, their raw coordinates remain sensitive to variations in camera position, image resolution, subject height, and body translation. Directly using raw keypoint coordinates therefore limits the generalization capability of the recognition model. To overcome these limitations, the proposed framework transforms the detected skeleton into a compact set of invariant biomechanical descriptors that characterize the geometric configuration of the human body rather than its absolute position within the image. Similar feature engineering strategies have demonstrated strong performance in posture recognition and biomechanical analysis [3].

The final 74-dimensional feature vector consists of 34 normalized skeletal coordinates (17 keypoints $\times$ 2 coordinates), 15 joint-angle features, 13 normalized proportional distance features, and 12 confidence values corresponding to anatomically important joints. This representation provides a compact yet highly discriminative

description of yoga posture while remaining largely invariant to translation, scale, and viewpoint variations.

3.3.1 Coordinate Normalization and Reference Frame Alignment

To eliminate the influence of global body translation, all body joints are represented in a body-centered coordinate system whose origin is defined as the midpoint between the left and right hip joints. The hip center is computed as

$$\mathbf{p}_{hip} = \frac{K_{\text{left_hip}} + K_{\text{right_hip}}}{2}, \tag{5}$$

where $\mathbf{p}_{hip}$ denotes the body reference point, while $K_{\text{left_hip}}$ and $K_{\text{right_hip}}$ represent the corresponding hip joint coordinates.

To achieve scale invariance, the torso length is selected as the normalization factor because it remains relatively stable across different yoga postures. The torso height is calculated as

$$h_{torso} = \left\| \mathbf{p}_{shoulder} - \mathbf{p}_{hip} \right\|, \tag{6}$$

where $\mathbf{p}_{shoulder}$ denotes the midpoint between the left and right shoulder joints, $\mathbf{p}_{hip}$ represents the hip center, and $\| \cdot \|$ denotes the Euclidean norm.

The normalized coordinates of the i^{th} body joint are then obtained as

$$\tilde{K}_i = \frac{K_i - \mathbf{p}_{hip}}{h_{torso}}, \tag{7}$$

where $\tilde{K}_i$ denotes the normalized keypoint coordinate, K_i is the original keypoint position, $\mathbf{p}_{hip}$ represents the body-centered origin, and h_{torso} is the torso-based scaling factor.

This normalization procedure effectively removes the influence of global translation and body scale, enabling the extracted features to represent only the intrinsic geometric configuration of the human posture. Consequently, the resulting representation becomes more robust to variations in camera distance, image resolution, and subject body dimensions, thereby improving the generalization capability of the proposed posture recognition framework.

3.3.2 Extraction of Joint Angles

While normalized joint coordinates preserve the global body configuration, they do not explicitly describe the articulation between adjacent body segments. To capture local biomechanical characteristics, the proposed framework computes anatomically meaningful joint angles from triplets of connected body joints. Joint angles are invariant to body translation and largely insensitive to subject scale, making them highly informative descriptors for yoga posture analysis.

For three consecutive body joints A, B, and C, the angle at the central joint B is computed using the cosine rule as

$$\theta = \cos^{-1}\left(\frac{(A - B) \cdot (C - B)}{\|A - B\| \, \|C - B\|} \right), \tag{8}$$

where θ denotes the joint angle, A, B, and C represent the coordinates of three connected anatomical keypoints, $(A - B)$ and $(C - B)$ denote the vectors extending from the central joint toward the adjacent joints, $(\cdot)$ represents the vector dot product, and $\|\cdot\|$ denotes the Euclidean norm used to calculate vector magnitude.

Using this formulation, the angle between the vectors $\overrightarrow{BA}$ and $\overrightarrow{BC}$ is determined. The proposed framework computes joint angles for the major articulations involved in yoga practice, including the elbows, shoulders, hips, knees, torso inclination, and other posture-dependent joints. These angular measurements directly describe body alignment and joint articulation, making them significantly more informative than raw image coordinates for biomechanical posture assessment.

Angular features also possess strong interpretability because most yoga postures are defined according to specific joint-angle relationships. Furthermore, the same angular measurements are subsequently reused during the rule-guided posture evaluation stage to quantify deviations from the ideal biomechanical posture. Consequently, angular features establish a direct connection between posture recognition and corrective feedback generation while maintaining consistency throughout the proposed framework [4, 11].

3.3.3 Distance and Proportional Features

Although joint angles effectively describe local articulation, they do not completely characterize the overall geometric structure of the human body. Therefore, the proposed framework additionally computes proportional distance features between selected anatomical joint pairs to capture global body geometry and spatial relationships.

The Euclidean distance between two body joints is calculated as

$$d_{ij} = \|K_i - K_j\|, \tag{9}$$

where d_{ij} denotes the Euclidean distance between the i^{th} and j^{th} body keypoints, K_i and K_j represent the corresponding joint coordinates, and $\|\cdot\|$ denotes the Euclidean norm.

Since absolute distances vary considerably with subject height and camera position, all distance measurements are normalized using the torso height computed during the coordinate normalization stage. The normalized distance feature is defined as

$$\tilde{d}_{ij} = \frac{d_{ij}}{h_{torso}}, \tag{10}$$

where $\tilde{d}_{ij}$ denotes the normalized distance feature, d_{ij} represents the original Euclidean distance, and h_{torso} corresponds to the torso height used as the scale normalization factor.

The proposed framework computes proportional distances between anatomically meaningful joint pairs, including shoulder width, hip width, upper- and lower-limb lengths, arm extension, leg extension, and other posture-dependent body proportions. Unlike absolute distances, these normalized measurements remain relatively invariant across individuals with different body dimensions while preserving the intrinsic geometric relationships required for posture recognition.

The combination of normalized coordinates, joint-angle measurements, and proportional distance features provides a comprehensive biomechanical representation that simultaneously captures local joint articulation and global body configuration. This compact structural representation significantly improves robustness against scale variations and viewpoint changes while providing highly discriminative features for temporal posture classification.

3.3.4 Confidence-Based Feature Integration

Reliable pose estimation is essential for accurate biomechanical analysis since errors in keypoint localization directly affect the quality of the extracted structural features. To enable the proposed framework to account for the reliability of each detected joint, confidence information produced by YOLOv8m-Pose is incorporated into the feature representation. Instead of discarding the detector confidence after keypoint localization, selected confidence values corresponding to anatomically important joints are appended to the biomechanical feature vector [10].

These confidence values provide an estimate of the reliability of the detected skeletal configuration and allow the subsequent learning model to implicitly assign greater importance to accurately detected joints while reducing the influence of uncertain detections. Since the confidence values reflect detector reliability rather than body geometry, they are preserved without geometric modification during feature engineering.

3.3.5 Final Feature Representation

The proposed biomechanical representation combines normalized joint coordinates, joint-angle measurements, normalized proportional distances, and confidence information into a unified feature vector. Consequently, each yoga posture is represented as

$$F \in \mathbb{R}^{74}, \tag{11}$$

where F denotes the final biomechanical feature vector and $\mathbb{R}^{74}$ represents a 74-dimensional real-valued feature space.

The resulting feature vector provides a compact yet highly discriminative representation of the human posture by simultaneously capturing body alignment, joint articulation, body proportions, and keypoint reliability. Compared with raw image pixels, this structured representation significantly reduces irrelevant visual variations while preserving the biomechanical information required for accurate posture recognition and posture correction.

3.4 Data Preprocessing and Augmentation Strategy

Prior to temporal sequence construction, the extracted biomechanical features are standardized to ensure stable optimization during neural network training. Each feature dimension is normalized using the mean and standard deviation computed exclusively from the training dataset in order to prevent information leakage into the validation and testing sets. The standardized feature vector is computed as

$$F' = \frac{F - \mu}{\sigma}, \tag{12}$$

where F denotes the original feature vector, F' represents the standardized feature vector, μ is the mean of the corresponding feature dimension computed from the training set, and σ denotes its standard deviation.

To improve model robustness and alleviate class imbalance, data augmentation is performed by injecting zero-mean Gaussian noise into the geometric feature components, including normalized coordinates, joint angles, and proportional distance measurements. Confidence features are intentionally excluded from augmentation because they represent detector reliability rather than biomechanical geometry. The augmentation process increases the diversity of the training data while preserving the underlying posture characteristics.

Following augmentation, the temporal training dataset consists of 4,833 training sequences generated from 1,896 original sequences, while the validation and testing sets remain unmodified to ensure unbiased performance evaluation.

3.5 Temporal Sequence Construction

Although the extracted biomechanical feature vector accurately represents the posture in a single frame, yoga practice is inherently a continuous temporal activity. Frame-wise analysis alone cannot effectively distinguish postures that exhibit similar intermediate body configurations during movement transitions. Therefore, the proposed framework models temporal dependencies by constructing fixed-length feature sequences prior to classification.

After preprocessing, Images extracted from the same posture recording are stored using sequential filenames. These filenames preserve the original acquisition order and are therefore used to reconstruct temporal sequences prior to sliding-window generation.

$$S_t = [F'_t, F'_{t+1}, \ldots, F'_{t+T-1}] \in \mathbb{R}^{T \times d}, \tag{13}$$

where $T = \texttt{SEQ_LEN} = 30$ denotes the sequence length and $d = 74$ represents the dimensionality of each biomechanical feature vector.

A window stride of one frame is employed, thereby maximizing the number of temporal training samples while preserving motion continuity across consecutive frames. During inference, a fixed-length temporal buffer stores the most recent feature vectors. Before the buffer becomes completely filled, the earliest available frame is repeatedly padded to maintain the required sequence length, allowing the system to begin posture recognition and feedback generation without waiting for thirty completely independent observations.

The resulting temporal dataset consists of

- Training sequences: 1,896 original $\rightarrow$ 4,833 augmented sequences.
- Testing and validation sequences: 845

This temporal representation enables the proposed framework to capture both spatial posture characteristics and temporal movement evolution, thereby improving discrimination between visually similar yoga postures.

3.6 Proposed RBTNet Architecture

The proposed Residual Biomechanical Transformer Network (RBTNet) is a lightweight temporal classification architecture specifically designed for skeleton-based yoga posture recognition. Rather than processing raw image pixels, RBTNet operates directly on the extracted biomechanical feature sequences, allowing the network to focus exclusively on posture geometry and temporal motion characteristics.

The architecture consists of six sequential processing stages:

1. Input Projection,
2. Learnable Positional Embedding,
3. Multi-layer Transformer Encoder,
4. Learned Attention Pooling,
5. Residual MLP Feature Refinement,
6. Final Classification Head.

The complete network contains approximately 448,922 trainable parameters, providing an effective balance between recognition accuracy and computational efficiency for near real-time inference.

3.6.1 Input Projection

Each frame within the temporal sequence is initially represented by a 74-dimensional biomechanical feature vector. Before Transformer processing, the input features are projected into a higher-dimensional embedding space through a fully connected projection layer followed by Layer Normalization and GELU activation. The projected representation is computed as

$$H_0 = \text{GELU}\left(\text{LayerNorm}(W_{proj}S + b_{proj})\right), \tag{14}$$

where $W_{proj} \in \mathbb{R}^{128\times 74}$ denotes the learnable projection matrix.

This transformation maps the original feature representation into a richer embedding space while Layer Normalization improves optimization stability and GELU introduces smooth nonlinear feature interactions before temporal attention learning.

3.6.2 Positional Embedding

Because self-attention mechanisms are inherently permutation invariant, explicit temporal position information must be incorporated into the input sequence. Therefore, a learnable positional embedding matrix is added to each projected feature vector as

$$H_1 = H_0 + P, \tag{15}$$

where $P \in \mathbb{R}^{T\times d_e}$ represents the learnable positional embedding matrix initialized from a Gaussian distribution.

Unlike sinusoidal positional encodings, learnable embeddings allow the network to optimize temporal representations directly for the fixed sequence length employed by the proposed framework.

3.6.3 Transformer Encoder

Temporal dependencies among consecutive posture frames are modeled using two stacked Transformer Encoder layers. Each encoder layer consists of Multi-Head Self-Attention (MHSA), a position-wise feed-forward network, residual skip connections, Layer Normalization, and dropout regularization.

The scaled dot-product attention is defined as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{16}$$

where Q, K, and V denote the query, key, and value projections, respectively.

The outputs of all attention heads are concatenated to produce

$$\text{MHSA}(H) = \text{Concat}(head_1, \ldots, head_{n_h})W^O, \tag{17}$$

where

$$head_i = \text{Attention}(HW_i^Q, HW_i^K, HW_i^V).$$

The position-wise feed-forward network is defined as

$$\text{FFN}(x) = W_2\,\text{GELU}(W_1 x + b_1) + b_2, \tag{18}$$

where $d_{ff} = 256$.

Two Transformer Encoder layers with eight attention heads are employed to simultaneously capture both local motion patterns and long-range temporal dependencies within the posture sequence.

3.6.4 Learned Attention Pooling

Instead of uniformly averaging all temporal features, the proposed framework employs learned attention pooling to identify the most informative posture frames within the sequence. The attention weights are computed as

$$\alpha_t = \text{softmax}_t(w_a^T h_t), \qquad \alpha \in \mathbb{R}^T, \tag{19}$$

where w_a denotes the learnable attention parameter and h_t represents the Transformer feature at time step t.

The final sequence representation is obtained as

$$z_{seq} = \sum_t \alpha_t h_t, \tag{20}$$

allowing the network to automatically emphasize the most discriminative temporal observations, such as the stable holding phase of a yoga posture, while reducing the influence of transitional frames.

3.6.5 Residual MLP Blocks

Following temporal aggregation, the pooled sequence embedding is further refined using a series of Residual Multi-Layer Perceptron (MLP) blocks. These residual blocks enhance nonlinear feature learning while preserving stable gradient propagation throughout the network. The residual design enables the model to learn increasingly discriminative posture representations without suffering from the degradation problem commonly observed in deep feed-forward architectures.

Let the pooled sequence representation be denoted by $z_{seq} \in \mathbb{R}^{128}$. Each residual block performs feature refinement according to

$$y = \mathrm{GELU}\left(\mathrm{Linear}\left(\mathrm{BatchNorm}(x)\right)\right) + \mathrm{Shortcut}(x), \tag{21}$$

where x represents the input feature vector, Batch Normalization stabilizes feature distributions, the linear transformation performs feature projection, and the GELU activation introduces smooth nonlinear feature learning. The shortcut connection is defined as

$$\mathrm{Shortcut}(x) = \begin{cases} \mathrm{Linear}(x), & \text{if the input and output dimensions differ,} \\ x, & \text{otherwise.} \end{cases}$$

The proposed network employs three cascaded residual blocks that progressively transform the learned temporal representation into a compact discriminative embedding:

- **Residual Block 1:** $128 \rightarrow 256$ (feature expansion), dropout probability $p = 0.30$.
- **Residual Block 2:** $256 \rightarrow 128$ (feature compression), dropout probability $p = 0.20$.
- **Residual Block 3:** $128 \rightarrow 64$ (final bottleneck representation), dropout probability $p = 0.10$.

The progressive expansion and compression strategy enables the network to capture higher-order feature interactions while gradually eliminating redundant information. Batch Normalization improves optimization stability by reducing internal covariate shift, whereas the residual shortcut connections facilitate efficient gradient propagation during backpropagation. Consequently, the final 64-dimensional representation preserves the most discriminative temporal and biomechanical characteristics required for accurate posture classification.

3.6.6 Classification Head

The refined feature representation obtained from the final residual block is mapped to the posture classification space using a fully connected output layer. Since the final experimental evaluation was conducted on eight yoga posture classes, the classification layer predicts eight output logits corresponding to the target posture categories.

The output logits are converted into posterior class probabilities through the Softmax function:

$$\hat{y} = \text{softmax}\left(W_{cls} x_{64} + b_{cls}\right), \tag{22}$$

where $x_{64} \in \mathbb{R}^{64}$ denotes the final feature embedding produced by the residual network, W_{cls} and b_{cls} represent the learnable classification parameters, and $\hat{y}$ denotes the predicted probability distribution over the eight posture classes.

The network is optimized using categorical cross-entropy loss with label smoothing in order to improve generalization and reduce overconfident predictions. The classification objective is defined as

$$\mathcal{L}_{cls} = -\sum_{i=1}^{N_c} \left[(1-\varepsilon) y_i + \frac{\varepsilon}{N_c}\right] \log(\hat{y}_i), \tag{23}$$

where $N_c = 8$ denotes the number of yoga posture classes, y_i represents the ground-truth label, $\hat{y}_i$ denotes the predicted probability for class i, and $\varepsilon = 0.1$ is the label smoothing coefficient.

Label smoothing regularizes the classifier by preventing excessive confidence in individual predictions, thereby improving model calibration and enhancing generalization performance on unseen posture sequences.

3.7 Deviation Analysis and Feedback Generation

Although the proposed RBTNet accurately classifies yoga postures, correct posture execution requires evaluating whether individual body joints satisfy the biomechanical constraints associated with the predicted pose. Therefore, the proposed framework incorporates a deterministic rule-guided deviation analysis module after posture classification to provide interpretable corrective feedback.

Unlike learning-based correction approaches, the proposed framework does not employ an additional neural correction network. Instead, each predicted posture class is associated with a predefined biomechanical knowledge base containing acceptable angular ranges for anatomically important joints. These ranges are derived from standard yoga posture alignment guidelines and are used to evaluate posture correctness.

For every monitored joint, the angular deviation from the acceptable biomechanical range is computed as

$$\Delta\theta_j = \left|\theta_j - \text{clip}\left(\theta_j, \theta_{j,lo}, \theta_{j,hi}\right)\right|, \tag{24}$$

where θ_j denotes the measured joint angle, $\theta_{j,lo}$ and $\theta_{j,hi}$ represent the lower and upper acceptable angular limits for joint j, respectively, and the clip$(\cdot)$ operator constrains the measured angle within the predefined biomechanical interval.

To facilitate comparison among different joints possessing different acceptable angular ranges, the deviation magnitude is normalized as

$$s_j = \frac{\Delta\theta_j}{\theta_{j,hi} - \theta_{j,lo} + \varepsilon}, \tag{25}$$

where s_j denotes the normalized deviation severity and ε is a small constant introduced to prevent numerical instability.

Whenever a joint exhibits a non-zero deviation ($s_j > 0$), the framework retrieves an appropriate posture-specific correction message from a predefined correction dictionary. Each corrective instruction is directly associated with the violated biomechanical constraint, ensuring complete transparency and interpretability of the generated feedback. For example, if the measured front knee angle in the Warrior pose is substantially larger than the acceptable range of 80°–110°, the framework produces the corrective instruction:

"Bend your front knee to approximately 90°."

Similarly, deviations in shoulder alignment, hip orientation, arm extension, or leg posture generate corresponding corrective recommendations specific to the detected posture.

Because the correction module relies entirely on deterministic biomechanical rules rather than learned correction networks, the generated feedback remains fully explainable, reproducible, and directly traceable to the measured anatomical joint angles. Furthermore, the proposed correction strategy does not require additional user-specific information such as age, gender, flexibility level, or anthropometric measurements, thereby improving its applicability across diverse users while maintaining high interpretability.

Overall, the proposed framework integrates robust skeletal feature extraction, temporal Transformer-based posture recognition, and rule-guided biomechanical reasoning into a unified intelligent system capable of simultaneously performing accurate yoga posture classification and interpretable posture correction. The lightweight RBTNet architecture, comprising approximately 448,922 trainable parameters and operating on compact 74-dimensional biomechanical feature vectors, provides an effective balance between recognition accuracy and computational efficiency, making the framework suitable for near real-time intelligent yoga assistance.

4 Results and Analysis

4.1 Experimental Setup

The proposed system was designed and tested using Google Colab with GPU support to optimize the process of training and experimenting on models. The whole workflow includes pose estimation, feature generation, classification, and evaluation phases. Pose keypoints were generated with the YOLOv8m-Pose object detector which recognized 17 anatomical joints per image. Further, the keypoints were converted into structural features that were used for classification.

4.2 Pose Estimation and Keypoint Extraction Results

YOLOv8m-Pose extracts the standard 17 COCO keypoints from each frame. When multiple people are detected, the highest-confidence box is kept. To filter out bad detections, we check the average confidence of four key points – both shoulders and both hips – and drop the frame if it falls below 0.3. This filtering removed very few

frames: only 12 out of roughly 1,908 training frames and 3 out of 845 test frames. That's a strong result – it means YOLOv8m-Pose handled the dataset's images cleanly, and pose estimation isn't a weak link in the pipeline.

4.3 Biomechanical Feature Representation Analysis

Instead of feeding raw pixels into the model, each frame is turned into a 74-dimensional feature vector. Coordinates are centered on the hip midpoint and scaled by torso length, so the features don't change just because someone is standing closer to or farther from the camera. On top of that, we compute 15 joint angles (elbow, knee, shoulder, torso lean, etc.), 13 limb-distance ratios (shoulder width, hip width, and so on), and 12 confidence scores for the main joints.

This biomechanical representation works better than raw pixels for a few practical reasons. First, normalizing by torso scale removes a lot of noise – height, distance from camera, position in frame – that a pixel-based model would otherwise have to learn to ignore. Second, angles and distances are exactly the language yoga correction is usually given in (*"knee should be at 90°"*), so the same features used for classification can be reused directly by the rule-based correction system. Third, keeping confidence scores as features lets the model implicitly trust well-detected joints more than shaky ones. Overall, this gives a compact, interpretable representation that fits the core idea behind the framework: posture is better understood through skeleton structure than through pixels.

4.4 Dataset Description and Preprocessing

The dataset is split into per-class folders for train and test. Eight classes have several hundred images each (400–418), while the dropped classes had only 26–48 training images – a clear imbalance. Rather than fixing this with resampling, the imbalance was resolved naturally: classes too small to form a 30-frame sequence simply didn't make it into training or testing. All 74-D features were standardized (zero mean, unit variance) using a scaler fit only on training data, then applied to validation and test sets, to avoid leakage. Augmentation (Gaussian noise, $\sigma = 0.05$) was added only to the training split, and only to the coordinate/angle/distance features – not the confidence scores, since those reflect detector reliability, not pose geometry.

4.5 Temporal Sequence Construction Analysis

Sequences are built using a sliding window of 30 frames with stride 1, taken from chronologically sorted images within each class. This gives many overlapping windows from a limited pool of images. At inference time, the same idea is used live: a 30-frame buffer fills up as new frames arrive, and once it's at least half full, it pads itself by repeating the earliest frame so the model always gets a fixed-length input. This lets the system start giving feedback before a full 30 frames have been collected, which is necessary for it to feel responsive in real time.

Looking at posture over a window of frames, rather than judging a single image, helps the model tell apart poses that briefly look similar mid-transition – something a frame-by-frame classifier would struggle with.

4.6 RBTNet Training and Convergence Analysis

The proposed RBTNet is optimized using categorical cross-entropy loss with label smoothing ($\varepsilon = 0.1$) to improve classification performance and reduce overconfident predictions. During training, the network learns discriminative temporal representations of biomechanical posture sequences, while the rule-guided biomechanical correction module operates independently after posture classification. This design enables the network to focus exclusively on robust posture recognition while maintaining an interpretable and deterministic feedback generation process.

Training converged quickly. Loss dropped sharply in the first few epochs and flattened around 0.51 by epoch 10. Validation accuracy hit 100% by epoch 10 and stayed there, triggering early stopping at epoch 21 – well before the 100-epoch limit. Test accuracy bounced around a bit during training (roughly 93.8%–98.2%) even while validation sat at 100%, which makes sense: the validation set is drawn from the same pool as training, so it's a bit easier to saturate than the genuinely held-out test set. After restoring the best checkpoint, final test accuracy came out to 98%. Fast convergence, an early stop, and a test score close to validation all point to the architecture being a good fit for this amount of data – no sign of instability.

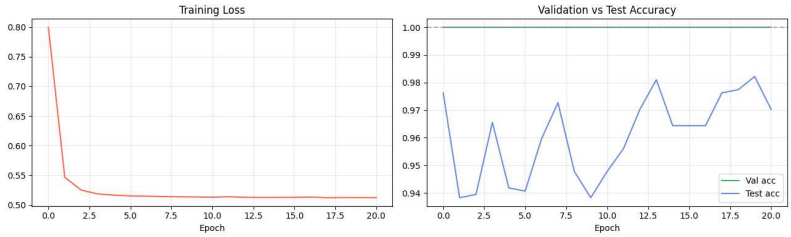


Fig. 2 Training Convergence Behavior.

4.7 Yoga Posture Classification Performance

Overall test accuracy was 98%, with weighted precision, recall, and F1 all around 0.98. Three classes – chair, cobra, and tree – scored a perfect 1.00 across the board. These poses are visually and structurally distinct enough that the model separates them with no trouble at all.

The only real confusion is one-directional: 20 of 189 warrior sequences got misclassified as dog, but no dog sequences were ever called warrior. That's why dog's precision dips to 0.88 (some of its "dog" predictions are actually mislabeled warriors) while its recall stays perfect, and why warrior keeps perfect precision but drops to 0.89 recall. This makes sense biomechanically: during the entry/exit of a warrior pose, the back leg briefly looks just as straight as it does in downward dog, before the front-knee bend and arm position settle into place. Since both poses share similar back-leg angle ranges in their correction rules, a 30-frame window dominated by this transition can genuinely look ambiguous, even with temporal attention helping.

Table 1 Classification performance of RBTNet on the test set

Class	Precision	Recall	F1-score	Support
Chair	1.00	1.00	1.00	168
Cobra	0.98	0.99	0.99	231
Dog	0.97	0.98	0.98	180
No_Pose	1.00	1.00	1.00	2
Shoulder_Stand	1.00	1.00	1.00	9
Triangle	0.95	0.90	0.92	10
Tree	1.00	1.00	1.00	192
Warrior	0.98	0.96	0.97	218
Accuracy			**0.98**	**1010**
Macro avg	0.98	0.98	0.98	1010
Weighted avg	0.98	0.98	0.98	1010

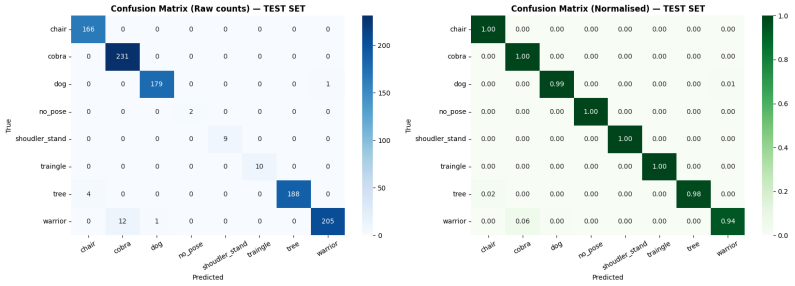
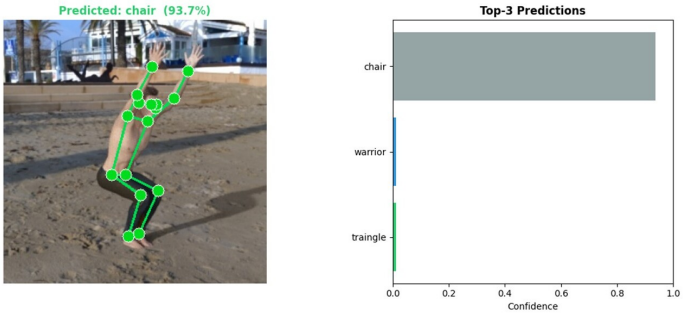


Fig. 3 Confusion Matrix.

4.8 Testing on Dataset Images and Live Camera Input

The full pipeline – YOLOv8m-Pose → feature extraction → scaling → temporal buffer → RBTNet → rule-based feedback – was tested on both static dataset images and live webcam video. On sample test images, the system classified chair and tree poses correctly with high confidence (98.3% and 98.8%), and flagged specific angle issues where relevant – for example, a left knee at 52.1° against an ideal range of 80°–110° for chair pose, with the instruction "Sink hips – knees at ∼90°". This feedback comes entirely from rule-guided biomechanical deviation checks: for whatever pose is predicted, the system looks up expected angle ranges for key joints, compares them to the live angles, and reports any joint outside range along with a pre-written, pose-specific instruction. None of this text is generated freely – it's deterministic and traceable to a fixed threshold, which keeps the feedback interpretable.

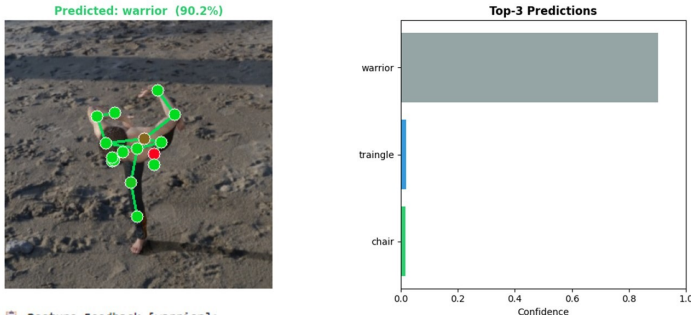
As illustrated in Figs. 3 and 4, the proposed system successfully performs real-time pose detection and generates feedback to assist users in correcting improper postures.



Posture Feedback [Chair]:

- Great posture! All joints are within the ideal range.

Fig. 4 Test on Random Image



Posture Feedback [Warrior]:

- Left Knee: 153.8° (ideal 80–110°) → Bend your front knee to approximately 90°.
- Right Knee: 158.2° (ideal 160–180°) → Keep the back leg straight.
- Left Shoulder: 48.8° (ideal 70–110°) → Keep the arms parallel to the floor.

Fig. 5 Random test on an Image with feedback generation

For live webcam use, the same pipeline runs roughly every 0.9 seconds. Predictions below 70% confidence are flagged as low-confidence and not acted on. Predictions above that threshold but not matching the target pose are flagged as wrong. Only confident, correctly matched predictions trigger the joint-correction overlay. This gating avoids giving shaky feedback during the ambiguous transition frames discussed above.

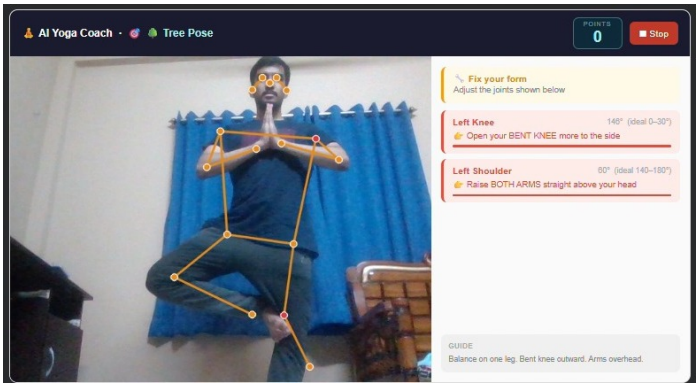


Fig. 6 Test on Live Camera 1.

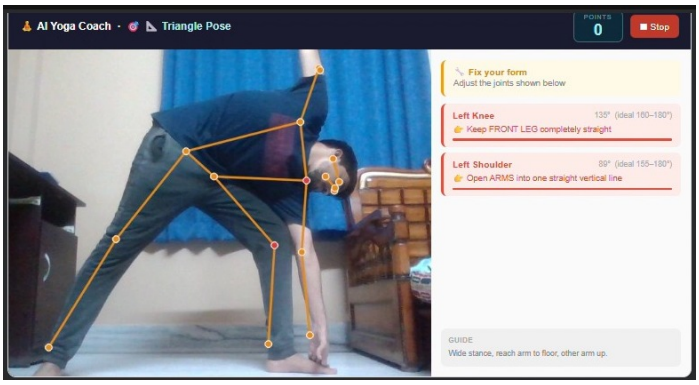


Fig. 7 Test on Live Camera 2.

4.9 Comparison with Baseline Models

OpenPose and MediaPipe are good at finding keypoints but don't classify poses or generate feedback on their own – that's left to whatever system is built on top. CNN/LSTM approaches add sequence modeling, but LSTMs process frames one at a time in order, which limits how well they connect frames that are far apart in the window. RBTNet's Transformer attention looks at the whole window at once, so it doesn't have that limitation. Traditional rule-based systems can give pose-specific feedback, but usually from a single frame with no learned representation, which makes them fragile around brief instability.

The proposed framework combines Transformer-based temporal posture recognition with a deterministic biomechanical correction module. The learned classifier provides accurate posture recognition, while the rule-based correction engine evaluates joint-angle deviations using predefined biomechanical constraints to generate interpretable corrective feedback.

Table 2 Capability Comparison of Existing Approaches and the Proposed Framework

Capability	OpenPose / MediaPipe	CNN/LSTM Methods	Rule-Based Yoga	Proposed Framework
Primary Focus	Pose landmark detection [5, 6]	Classification and sequence learning [8, 9]	Geometric posture evaluation [4]	Integrated detection, classification, and correction
Temporal Understanding	Generally absent	Sequence modeling supported	Typically absent	Transformer-based temporal reasoning
Correction Strategy	Limited feedback support	Recognition-oriented analysis	Threshold / angle-based correction	Adaptive learning-based correction
Feedback Capability	Basic posture estimation	Limited guidance	Static corrective suggestions	corrective feedback
Modeling Mechanism	Skeleton extraction	Deep neural learning	Rule/constraint evaluation	Hybrid deep learning + attention
System Integration	Partially integrated	Moderately integrated	Task-specific design	Unified end-to-end framework

5 Conclusion

This paper proposes a complete architecture for detecting and correcting real-time yoga poses based on the integration of pose estimation, feature engineering, and deep learning. Human skeletal keypoints estimated using YOLOv8m-Pose is used to extract the relevant information into a feature vector of size 74 with the aid of structured feature engineering. A comparative study with different models proves the superiority of the proposed model due to its ability to facilitate gradient flow for stable learning. The model is trained using a multi-class yoga dataset comprising 8 categories after applying data augmentation and normalization techniques to achieve 98.21% accuracy, proving high generalization.

In addition, the performance of the model was assessed using the classification report and confusion matrix. According to the results, the model achieved high performance while making mistakes for only a few categories due to structural similarities in the poses. Lastly, the effectiveness of the model was evaluated using test images randomly selected from the dataset and live camera tests with high speed and reliability in all trials conducted. These results prove that feature engineering combined with the proposed deep learning model is capable of performing pose recognition effectively and efficiently.

For future work, the system can be upgraded through the use of larger and more varied datasets in order to increase its ability to generalize from one body type and environment to another. Also, by incorporating hardware components like depth cameras and sensors, increased accuracy of motion and spatial analysis can be achieved.

References

[1] A. K. Rajendran and S. C. Sethuraman, "A survey on yogic posture recognition," *IEEE Access*, vol. 11, pp. 11183–11210, 2023. DOI: 10.1109/ACCESS.2023.3240769.

[2] C. Haldar, "An intelligent framework for real-time yoga pose detection and posture correction," arXiv preprint arXiv:2603.26760, 2025. [Online]. Available: https://arxiv.org/abs/2603.26760.

[3] A. K. Rajendran and S. C. Sethuraman, "Yoga posture alignment and injury analysis," *IEEE Access*, vol. 11, pp. 11183–11210, 2023. DOI: 10.1109/ACCESS.2023.3240769.

[4] A. Gurav, S. Dhond, S. Bhosale, and U. Gurav, "MoveNet-cosine: Innovative computer vision-based approach for yoga posture detection," in *Proc. Int. Conf. Information, Implementation and Innovation in Technology (I2ITCON)*, Pune, India, Jul. 2025. DOI: 10.1109/I2ITCON65200.2025.11210493.

[5] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, Jan. 2021. DOI: 10.1109/TPAMI.2019.2929257.

[6] C. Ferraris, G. Amprimo, S. Cerfoglio, L. Vismara, and V. Cimolin, "A deep dive into MediaPipe pose for postural assessment: A comparative investigation," *IEEE Access*, vol. 13, pp. 211055–211070, 2025. DOI: 10.1109/ACCESS.2025.3643126.

[7] A. Gurav, S. Dhond, S. Bhosale, and U. Gurav, "MoveNet-based real-time pose estimation for yoga applications," 2025. [Online]. Available: https://arxiv.org/.

[8] R. K. Nayak, S. Singh, S. Waghdhare, S. Telang, S. Das, and G. Bargaley, "A comparative study of classification AI models for yoga pose detection and correction," in *Proc. Int. Conf. Reliability, Infocom Technologies and Optimization (ICRITO)*, Noida, India, Sep. 2025. DOI: 10.1109/ICRITO66076.2025.11241748.

[9] M. Zhao, N. Lu, and Y. Guan, "Classification of Pilates using MediaPipe and machine learning," *IEEE Access*, vol. 12, pp. 77133–77145, 2024. DOI: 10.1109/ACCESS.2024.3406678.

[10] O. H. A. Zaghloul, Y. Khalil, H. F. Alsan, K. Hasoglu, and T. Arsan, "Real-time pose recognition using smartphone-friendly computer vision," in *Proc. Int. Symp. Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2025. DOI: 10.1109/ISMSIT67332.2025.11268158.

[11] Y.-M. Ko, A. Nasridinov, and S.-H. Park, "Real-time AI posture correction for powerlifting exercises using YOLOv5 and MediaPipe," *IEEE Access*, vol. 12, pp. 195830–195845, 2024. DOI: 10.1109/ACCESS.2024.3516723.

[12] M. Fang, "Yoga Pose Classification Based on Transfer Learning of Vision Transformer," in *Proc. 6th Int. Conf. Information Technologies and Electrical Engineering (ICITEE)*, 2023. DOI: 10.1145/3640115.3640161.

[13] R. S. Wadawadagi, A. V. G., V. Desai, and V. E., "ViTYoga: Vision Transformer for Real-Time Yoga Pose Estimation and Analysis," in *Proc. Int. Conf. Computing for Sustainability and Intelligent Future (COMP-SIF)*, 2025. DOI: 10.1109/COMP-SIF65618.2025.10969947.

[14] R. Srivastava, A. Sharma, and P. Verma, "Real-time yoga pose classification with 3-D pose estimation model with LSTM," *Multimedia Tools and Applications*, 2023. DOI: 10.1007/s11042-023-17036-8.

[15] D. Swain, S. Mohapatra, and B. Das, "Deep learning models for yoga pose monitoring," *Algorithms*, vol. 15, no. 11, 2022. DOI: 10.3390/a15110403.

[16] S. M. Suhas, P. Shree, and S. Bhat, "A framework for realtime multiview yoga pose detection and corrective feedback using MoveNet and convolutional neural network," in *Proc. 4th Int. Conf. Intelligent Technologies (CONIT)*, Karnataka, India, 2024, pp. 1–6. DOI: 10.1109/CONIT61985.2024.10627075.

[17] D. K. Sinha and R. Verma, "Enhancing yoga practice: Real-time pose analysis and personalized feedback," in *Proc. Int. Conf. Computing, Communication and Intelligent Systems (ICCCIS)*, 2023. DOI: 10.1109/ESIC60604.2024.10481659.

[18] P. Sharma et al., "Yoga pose detection and correction using 3D pose estimation and machine learning," in *Proc. IEEE Int. Conf. Recent Advances in Computing*, 2023. DOI: 10.1109/ICCES63552.2024.10860244.

[19] D. Borthakur et al., "Yoga pose estimation using angle-based feature extraction," *Healthcare*, vol. 11, no. 24, 2023. DOI: 10.3390/healthcare11243133.

[20] P. B. K. and G. Mahitha, "Advancements in Yoga Pose Recognition and Correction: A Comprehensive Literature Review," 2025. DOI: 10.35940/ijsce.C3679.15030725.

[21] A. Vaswani et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008. [Online]. Available: https://papers.nips.cc/paper/7181-attention-is-all-you-need.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[23] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," arXiv preprint arXiv:1606.08415, 2016. [Online]. Available: https://arxiv.org/abs/1606.08415.

[24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019. [Online]. Available: https://arxiv.org/abs/1711.05101.

[25] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[26] A. K. Yadav et al., "iSmartYog: A real-time yoga pose recognition and correction feedback model using deep learning for smart healthcare," *IEEE Access*, vol. 10, pp. 35147–35158, 2022. DOI: 10.1109/ICSSES58299.2023.10201061.

[27] D. Jadhav et al., "PosePerfect: Integrating HRNet and Gemini Vision Pro for enhanced yoga posture classification and posture correction analysis," in *Proc. Int. Conf. Artificial Intelligence and Machine Vision (AIMV)*, 2024. DOI: 10.1109/I2CT61223.2024.10543478.

[28] S. Bhattacharya et al., "Real-time yoga pose recognition and correction feedback system using deep learning and computer vision," in *Proc. IEEE Int. Conf. Electronics, Communication and Aerospace Technology (ICECA)*, 2023. DOI: 10.1109/CICTN64563.2025.10932462.

[29] S. Sinha et al., "Yoga pose classification and correction using PoseNet," in *Proc. Int. Conf. Computing, Communication and Networking Technologies (ICCCNT)*, 2022. DOI: 10.1109/IATMSI60426.2024.10503105.

[30] A. Sharma and P. Bhattacharya, "Yoga pose recognition and correction using deep learning," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 9, no. 8, pp. 731–738, 2022. DOI: 10.1109/ICCCNT61001.2024.10724511.

[31] P. Yadav and R. Singh, "Yoga pose recognition with real-time correction using deep learning," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 70, no. 4, pp. 32–39, 2022. DOI: 10.1109/ICSCDS56580.2023.10104781.

[32] A. Patel and R. Gupta, "Yog-Guru: Real-time yoga pose correction system using deep learning methods," *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, vol. 11, no. 9, pp. 871–878, 2023. DOI: 10.1109/ICCICT50803.2021.9509937.