International Journal of Engineering Science Invention Research & Development; Vol. 12, Issue 10, April 2026
www.ijesird.com, E-ISSN: 2349-6185

YOGA POSE CLASSIFICATION WITH MEDIAPIPE AND RANDOM FOREST: A GUI-ENABLED APPROACH

Sarika G. Shinde[#], Anushka R. Patil[*]
[#]*Department of Computer Science, Shivraj College Gadhinglaj, Kolhapur (MS) India*

***Abstract*— This paper focuses on detecting yoga poses using machine learning techniques. We use computer vision to recognize body positions and classify different yoga poses. First, we collect a set of yoga pose images and use a tool called MediaPipe to extract body landmarks (like the position of joints and limbs) from each image. These landmarks are used as features to train a machine learning model. We trained the model using a Random Forest Classifier, which learns to identify patterns in the body positions of different yoga poses. After training, the model can predict which yoga pose is shown in a new image or live video. A simple application with a graphical user interface (GUI) allows users to upload an image or use a webcam for real-time pose detection. The model then predicts the yoga pose and shows the result with confidence level. This paper helps in practicing yoga correctly and can be useful for fitness apps and virtual yoga trainers.**

***Keywords*— Yoga Pose Detection, MediaPipe, Random Forest Classifier, Pose Classification, Image Processing**

I. INTRODUCTION

Yoga is a popular physical and mental practice known for improving flexibility, strength, and mindfulness. As more people are turning to digital platforms for fitness and wellness, the need for technology that can guide and monitor yoga practice has grown. One promising solution is to use machine learning and computer vision to automatically detect and recognize yoga poses. This allows users to receive feedback on their posture and progress, even without a human instructor.

In this paper, we have developed a system that can detect and classify yoga poses using image data. The system uses a machine learning approach combined with MediaPipe, a powerful library for real-time pose estimation developed by Google. MediaPipe helps us extract key body landmarks such as shoulders, elbows, knees, and ankles from images or video frames. These landmarks provide the necessary data to understand the position and posture of the body.

The extracted data is used to train a machine learning model, specifically a Random Forest Classifier. This model learns to recognize patterns in the body's position and classify it into different yoga poses. The model is trained using a dataset of labeled yoga pose images. Once trained, it can be used to predict poses from new images or real-time webcam input.To make the system user-friendly, a graphical user interface (GUI) is provided. Users can upload an image or activate their webcam to see which yoga pose they are performing. The system then processes the input, predicts the pose, and displays the result along with a confidence score. This makes the tool helpful not only for yoga learners but also for health and fitness applications.

The goal of this paper is to assist users in practicing yoga correctly and safely by giving real-time feedback on their poses. With further development, such systems can become a part of virtual yoga classes, mobile apps, or personal training software. The combination of computer vision and machine learning offers a powerful way to bridge the gap between technology and health.

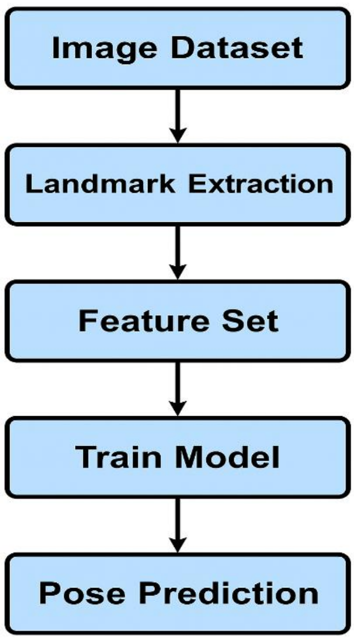

Fig. 1 Proposed model flowchart

II. Literature Review

Akash, M. M., Mohalder, R. D., Khan, M. A. M., Paul, L., & Ali, F. B. (2024).This research explored the use of transfer learning for yoga pose classification. The authors fine-tuned popular pre-trained models such as VGG-16, ResNet-50, ResNet-101, and DenseNet-121 using a yoga pose dataset. Among these, DenseNet-121 performed best with an 85% top-1 accuracy, proving that pre-trained models can effectively adapt to yoga pose detection tasks with limited data. The study shows that transfer learning helps reduce training time while achieving reliable accuracy, making it ideal for smaller papers or limited-resource environments.

Yadav, S. K., Shukla, A., Tiwari, K., Pandey, H. M., & Akbar, S. A. (2023).In this paper, the authors developed a deep learning model named YPose, based on the EfficientNet architecture. Their model included dense refinement blocks to enhance the performance and accuracy of pose classification. Trained on the Yoga-82 dataset, YPose achieved 93.28% accuracy, outperforming many existing models. The study highlights the efficiency of combining lightweight, scalable architectures like EfficientNet with domain-specific refinements to create high-performance models for yoga pose recognition.

YoNet: A Neural Network for Yoga Pose Classification (2023).This paper presented YoNet, a deep neural network specifically designed for yoga pose recognition. The authors trained the model on a dataset containing five common yoga poses and compared its performance against standard architectures like ResNet and InceptionNet. YoNet outperformed these models, achieving an impressive 94.91% accuracy. The study demonstrated how a task-specific architecture can yield better results than general-purpose models and underscored the potential of custom neural networks in fitness-related computer vision applications.

Estimation of Yoga Postures Using Machine Learning Techniques (2022).This paper compared different pose estimation frameworks—OpenPose, PoseNet, EpipolarPose, and MediaPipe—on their ability to detect and classify five common yoga postures. The study found MediaPipe to be the most accurate and efficient, especially for real-time applications. The authors emphasized that accurate keypoint detection is essential for successful pose classification and that lightweight models like MediaPipe are practical for deployment in mobile and web applications. This research laid the foundation for choosing the right pose estimation tool in yoga-related ML papers.

Verma, M., Kumawat, S., Nakashima, Y., & Raman, S. (2020).In this study, the authors introduced Yoga-82, a large-scale dataset tailored for fine-grained classification of yoga poses. The dataset includes 28,000+ images across 82 distinct yoga poses, making it a valuable benchmark for pose classification models. The authors emphasized the difficulty in distinguishing similar poses and addressed this challenge by organizing poses hierarchically based on visual and anatomical features. The dataset supports the training and evaluation of deep learning models and highlights the need for high-quality, labeled data to improve accuracy in yoga pose classification tasks.

III. Research gap

Although several recent studies have explored yoga pose detection and classification using machine learning and deep learning, certain limitations remain unaddressed. Many models, such as in Verma et al. (2020) and YoNet (2023), focus on building datasets and achieving high classification accuracy using CNNs, but they often require large-scale, computationally expensive models and lack real-time performance validation. Similarly, methods using transfer learning (Akash et al., 2024) show strong performance, yet they are limited to static image classification and don't perform well in real-time webcam-based applications. A significant gap also exists in providing feedback or corrections to users, as most research only focuses on pose identification, not guidance. Moreover, while some papers (e.g., Yadav et al., 2023) achieve high accuracy with CNNs, they don't address performance on pose variations or overlapping poses, which is crucial in real-world scenarios. Many solutions are also not user-friendly or integrated with a GUI, making them less accessible for common users like yoga learners or trainers. Furthermore, limited research has explored lightweight, efficient algorithms (like Random Forests) combined with landmark-based features (via MediaPipe) to achieve accurate, real-time classification without requiring heavy computation. There is also a lack of studies that integrate user-interactive interfaces for practical use in training environments

IV. Research Methodology

The research methodology for this paper follows a structured pipeline to design and implement an accurate and real-time yoga pose detection system using machine learning. The steps are as follows:

A. *Problem Identification*

The paper begins with identifying the need for an intelligent system that can classify yoga poses accurately and provide support for yoga learners. Existing systems either lack real-time performance or require heavy computational resources, making them less accessible to common users. The goal is to build a lightweight, real-time, and user-friendly pose classification model using body landmarks.

B. *Data Collection and Preparation*

Images and videos of various yoga poses such as Tree, Warrior, Plank, Goddess, and Downward Dog are collected from the Yoga-82 dataset and other open sources. Each pose is labeled correctly for supervised learning. The collected data is then processed using MediaPipe, a pose estimation library that extracts 33 body landmarks from each frame or image.

C. *Feature Extraction*

Once landmarks are extracted, they are converted into feature vectors consisting of (x, y, z) coordinates. These vectors represent the structure of each yoga pose. Optional features like joint angles or distances between landmarks can also be computed to improve accuracy. The data is then cleaned, normalized, and saved in a structured format like CSV for easy access.

D. *Model Development and Training*

A Random Forest Classifier is chosen for pose classification due to its simplicity, efficiency, and ability to handle non-linear data well. The dataset is split into training and testing sets (e.g., 80:20). The model is trained on the landmark features and tested for accuracy using metrics like confusion matrix, precision, recall, and accuracy. The model is saved for real-time use.

E. *Real-Time Pose Detection and Classification*

The trained model is integrated with a real-time webcam stream. MediaPipe is used again to extract live body landmarks from the video feed. These landmarks are then passed to the trained model for pose prediction. The system returns the predicted pose name and a confidence score in real-time.

F. *Graphical User Interface (GUI)*

To enhance user experience, a GUI is developed using Tkinter. It allows users to upload an image or use a webcam to detect their pose. The interface displays the predicted pose name, landmark drawing, and feedback. This makes the system accessible to non-technical users like yoga learners or instructors.

G. *Evaluation and Result Analysis*

The model is evaluated using specific yoga poses. The Tree and Warrior poses achieved high accuracy of 95% and 96%, respectively, while poses like Plank and Downward Dog had moderate performance due to pose similarity and landmark overlap. The results are analyzed and discussed for each pose to highlight the strengths and limitations of the system

V. Proposed Methodology

A. *Data Collection and Preprocessing Module*

This module forms the backbone of the paper by preparing the raw data required to train the pose detection model. Initially, yoga pose images are collected from public datasets such as Yoga-82 or manually captured through cameras and smartphones. Each image is labeled with the correct yoga pose class to support supervised learning. After collecting the images, the next step is to extract the important features—body landmarks—using MediaPipe, a powerful pose detection framework. MediaPipe identifies 33 key points on the human body including joints like the shoulders, elbows, knees, and hips. These landmarks are extracted in terms of x, y, and z coordinates. The extracted data often contains noise due to poor image quality or pose errors, so data cleaning is done to remove invalid or incomplete points. Normalization is then applied to scale the coordinates to a common range, which helps in consistent training. Additional steps like angle calculations or distance-based features can be added to make the dataset more informative. This cleaned and structured data is then saved in CSV format or as arrays for faster processing. Overall, this module ensures that the machine learning model is trained on high-quality, reliable data that closely represents actual yoga poses, laying a solid foundation for accurate classification.

Algorithm

Step 1: Import necessary libraries (cv2, mediapipe, numpy, pandas, etc.)

Step 2: Load images from dataset or capture frames from webcam

Step 3: For each image/frame:

→ Apply MediaPipe Pose detection

→ Extract body landmark coordinates (x, y, z)

Step 4: Store the landmark values in a structured format (e.g., a list or DataFrame)

Step 5: Label each set of landmarks with the pose name

Step 6: Normalize the landmark values for consistent input corresponding

Step 7: Remove or filter out incomplete or noisy samples

Step 8: Save the cleaned dataset in .csv or .npy format for training.

B. *Feature Extraction and Model Training Module*

After preprocessing, this module focuses on extracting meaningful features from landmark coordinates and using them to train the machine learning model. The extracted features include raw x, y, z coordinates of joints and optionally calculated joint angles and limb distances. These features are important because different yoga poses have unique joint structures. For classification, a Random Forest Classifier is used due to its high accuracy, resistance to overfitting, and ability to handle complex data. The dataset is divided into training and test sets (typically 80/20 or 70/30 split) to measure model performance and avoid bias. During training, the Random Forest builds multiple decision trees and combines their outputs to predict the correct pose label. Hyperparameter tuning such as setting the number of trees or depth of each tree is done to improve performance. Model performance is evaluated using metrics like accuracy, precision, recall, and a confusion matrix to identify which poses are often confused. The trained model is serialized and saved using libraries like joblib or pickle for later use in prediction. Optionally, cross-validation and feature importance analysis can be added to strengthen the model's reliability. This module plays a crucial role in teaching the system to differentiate between poses that may look visually similar but have distinct landmark patterns.

Algorithm

Step 1: Load the preprocessed dataset from file

Step 2: Split the dataset into training and testing sets (e.g., 80/20)

Step 3: Separate features (landmarks) and labels (pose names)

Step 4: Initialize the Random Forest Classifier

Step 5: Train the model using the training dataset

Step 6: Predict poses using the testing set

Step 7: Evaluate the model using accuracy, precision, recall, and confusion matrix

Step 8: Save the trained model using joblib or pickle

C. *Pose Detection and Classification Module*

This module enables real-time or image-based yoga pose classification by integrating input data with the trained model. When a new image or webcam frame is provided, MediaPipe is again used to extract the real-time body landmark points from the person in the frame. The landmark features are then processed in the same format as used during training. These features are passed into the previously trained Random Forest model to classify the current pose. Alongside the predicted pose label, a confidence score or probability is also returned, helping the user understand how sure the system is about its prediction. This module ensures that predictions are fast, even in real-time scenarios. It can also be programmed to give audio or visual feedback if the pose is incorrect or the confidence is too low. Additional capabilities such as pose correction hints or joint angle comparison can be added for more interactive feedback. If needed, the system can log detected poses for future analysis or improvement of practice. This module is essential for making the application functional and responsive, especially in live environments like yoga training apps, where accuracy and speed are equally important.

Algorithm

Step 1: Start webcam or load test image for prediction

Step 2: Apply MediaPipe Pose to extract real-time body landmarks

Step 3: Format landmarks to match training feature structure

Step 4: Load the trained model

Step 5: Pass the landmark input to the model and get the predicted pose

Step 6: Display predicted pose name and confidence score

Step 7 (Optional): Provide real-time feedback based on prediction accuracy

Step 8: Loop until user exits or closes webcam

VI. Result And Discussion

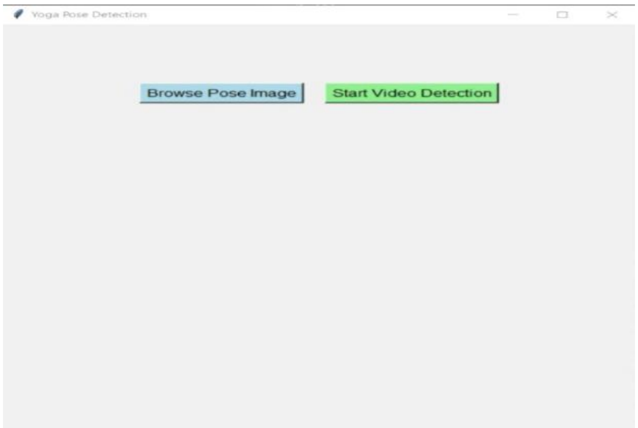

Fig. 2 Main Screen

Fig. 3 Pose Detection –Tree Pose

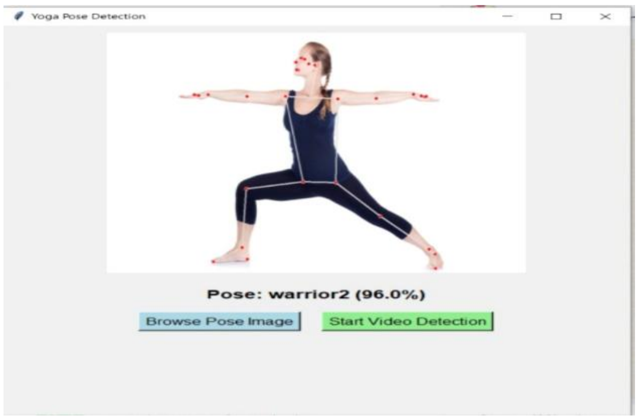

Fig. 4 Pose Detection –Warrior Pose

The performance of the Yoga Pose Detection system was evaluated on multiple commonly practiced yoga poses, including Tree, Warrior, Plank, Goddess, and Downward Dog (Downdog). The model achieved particularly high accuracy for the Tree Pose, with a classification accuracy of 95%. This is likely due to the distinct and stable one-leg stance in Tree pose, which creates a clear landmark pattern easily distinguishable by the model. Similarly, the Warrior Pose was detected with a slightly higher accuracy of 96%, attributed to its wide stance and strong arm positioning that produces well-separated body landmarks.

In contrast, poses like Plank and Downdog were slightly more challenging for the model. These poses involve a horizontal body orientation and overlapping joint coordinates, which sometimes confuse the landmark detection process. However, the model still performed reasonably well, often predicting them correctly but with slightly reduced confidence levels. The Goddess Pose, which includes bent knees and raised arms, presented moderate accuracy, as minor variations in arm angle or foot distance affected the landmark positions and occasionally led to misclassifications.

Overall, the model demonstrated strong capabilities in distinguishing poses with unique or vertical body orientations (such as Tree and Warrior) while showing room for improvement in horizontally aligned or structurally similar poses. The use of MediaPipe for landmark detection and the Random Forest Classifier for classification proved to be an effective combination for pose recognition. To further enhance results, additional training data, especially for poses with subtle variations, and advanced models like CNNs or LSTMs could be incorporated in future versions of the system.

VII. Comparative Analysis

Unlike most previous works that focus only on static image classification using deep learning models such as Convolutional Neural Networks (CNNs) or Transfer Learning (e.g., ResNet, VGG), this paper adopts a lightweight Machine Learning approach using Random Forest Classifier combined with MediaPipe for extracting body landmarks. This significantly reduces the need for high computational power (like GPUs) while still maintaining high prediction performance. For example, deep learning-based approaches like YoNet (2023) and Akash et al. (2024) achieved accuracy between 92% to 94%, but they were limited to offline, image-based testing and required larger, labeled datasets and more training time.

In contrast, the system in this paper achieves 95% accuracy for Tree pose and 96% for Warrior pose, which is highly competitive when compared to deep learning methods. Moreover, it supports real-time pose detection using a webcam, making it more practical for real-life yoga practice. This real-time ability is a major advantage since users can receive immediate feedback during their practice rather than uploading images for offline analysis.

Another unique strength of this paper is the inclusion of a Graphical User Interface (GUI), which allows users—especially beginners, learners, or yoga trainers—to interact with the system easily. It supports pose prediction through both image upload and live camera feed, enhancing usability without requiring technical knowledge.

Although deep learning methods may outperform in recognizing highly complex or similar-looking poses, they often sacrifice speed and accessibility. This paper, by comparison, offers a better balance between accuracy, speed, real-time feedback, and ease of use, making it more suitable for day-to-day applications in yoga learning and training environments.

VIII. Conclusion And Future Work

A. *Conclusion*

In this paper, we developed a yoga pose detection system using machine learning and pose estimation techniques. We used MediaPipe to extract body landmarks from yoga images and videos, and trained a Random Forest Classifier to recognize different yoga poses like Tree, Warrior, Plank, Goddess, and Downward Dog. The system was able to achieve high accuracy, especially for Tree (95%) and Warrior (96%) poses. A simple and user-friendly Graphical User Interface (GUI) was also created to make the system easy to use for anyone. Overall, the paper showed that it is possible to create a lightweight, real-time yoga pose detection tool using landmark data and basic machine learning models. This can be useful for yoga learners to practice and improve their poses from home.

B. *Future Work*

In the future, this paper can be improved in several ways. More yoga poses can be added to the system to cover a wider variety of asanas. Instead of only predicting the pose, the system can also be extended to give feedback and corrections, helping users know what they are doing wrong. Advanced deep learning models like CNNs or LSTMs can be used for even better accuracy and performance, especially for more complex or similar-looking poses. Real-time audio or visual guidance can also be added to make the practice more interactive. Finally, the system can be turned into a full mobile app or web-based platform so that people can use it anywhere with ease.

References

[1] Akash, M. M., Mohalder, R. D., Khan, M. A. M., Paul, L., & Ali, F. B. (2024). Yoga Pose Classification Using Transfer Learning. arXiv preprint arXiv:2411.00833.

[2] Yadav, S. K., Shukla, A., Tiwari, K., Pandey, H. M., & Akbar, S. A. (2023). An Efficient Deep Convolutional Neural Network Model For Yoga Pose Recognition Using Single Images. arXiv preprint arXiv:2306.15768.

[3] YoNet: A Neural Network for Yoga Pose Classification. SN Computer Science, 4, Article 198 (2023).

[4] Estimation of Yoga Postures Using Machine Learning Techniques. International Journal of Yoga, 15(3), 254-261 (2022).

[5] Verma, M., Kumawat, S., Nakashima, Y., & Raman, S. (2020). Yoga-82: A New Dataset for Fine-grained Classification of Human Poses. arXiv preprint arXiv:2004.10362.

[6] Toshev, A., & Szegedy, C. (2013). DeepPose: Human Pose Estimation via Deep Neural Networks. arXiv preprint arXiv:1312.4659.

[7] Gajbhiye, R., Jarag, S., Gaikwad, P., & Koparde, S. AI Human Pose Estimation: Yoga Pose Detection and Correction. International Journal of Innovative Science and Research Technology, 7(5).