RV College of Engineering®

Go, change the world

YOGA POSE DETECTION USING MEDIAPIPE

Name :	**USN :**
Shreyas Jain	1RV22AI055
Roopa Iranna Bagalkoti	1RV23AI404
K. Preethi	1RV23AI402

Mentors : Prof. Somesh Nandi , Prof. Anupama

Introduction

In recent years, yoga has gained significant popularity due to its numerous health benefits. However, many practitioners struggle with maintaining correct posture, which can affect the effectiveness of their practice and potentially lead to injuries. This project aims to address this issue by developing an automated **Yoga Pose Detection Model** using **MediaPipe** and **CNN**.

By leveraging **MediaPipe's pose estimation capabilities**, key body landmarks are extracted from images or videos. These landmark points are then used to train a **K-Nearest Neighbors (KNN) classifier**, which accurately identifies different yoga poses. The goal is to create a real-time system that can assist users in improving their posture and ensuring proper form.

Key Highlights:

- **Why Yoga Pose Detection?** Helps users correct their posture, avoid injuries, and enhance their practice.
- **Technologies Used:** MediaPipe for keypoint extraction, CNN for feature learning, and KNN for classification.
- **Applications:** Virtual Yoga Trainers, Fitness Assistance, Posture Correction Systems.

Problem Statement

Yoga is widely practiced for its physical and mental health benefits, but incorrect posture can reduce its effectiveness and lead to injuries. Beginners often struggle to maintain proper form without real-time feedback, and traditional methods of pose correction rely on human instructors, which may not always be accessible.

To address this, we propose an **AI-powered Yoga Pose Detection System** that can automatically recognize and classify yoga poses using **MediaPipe and CNN**. By extracting key body landmarks and analyzing posture through a trained model, users can receive **real-time feedback**, ensuring correct alignment and improving their practice.

Literature Survey(LS)

Paper	Authors	Approach	Key Points
"Real-time Yoga Recognition using Deep Learning and MediaPipe" (2022)	Kumar, S., Patel, R., & Singh, M.	Combined MediaPipe Pose estimation with custom CNN architecture	- Achieved 94.6% accuracy on yoga pose classification - Used transfer learning with MobileNetV2 - Real-time processing at 30 FPS - Dataset of 82,000 yoga poses
"YogaNet: 3D Pose Estimation for Yoga Asanas using Deep Learning" (2021)	Zhang, L., Wang, H., & Johnson, K.	Multi-stage CNN with MediaPipe skeleton tracking	- Introduced novel pose normalization technique - Handles occlusions effectively - Works with both video and image input - Custom loss function for pose alignment

Literature Survey(LS)

Paper	Authors	Approach	Key Points
"Automated Yoga Posture Recognition using K-Nearest Neighbors and MediaPipe Framework" (2023)	Sharma, A., Gupta, V., & Lee, J.	KNN classifier with MediaPipe landmarks as features	- Lightweight and computationally efficient - 91.2% accuracy on basic poses - Easy to train on new poses - Works well with limited data
"Deep Learning for Yoga: A Comparative Study of CNN Architectures" (2021)	Chen, X., Rodriguez, M., & Kim, S..	Comparison of ResNet, VGG, and custom CNN models	- ResNet50 performed best with 95.8% accuracy - Analysis of computational requirements - Dataset of 100,000 images

Literature Survey(LS)

Paper	Authors	Approach	Key Points
"Human Pose Estimation for Yoga Using MediaPipe and Ensemble Learning" (2023)	Anderson, K., Thompson, R., & Das, P.	Ensemble of CNN and KNN with MediaPipe features	- Combined benefits of both approaches - Robust to lighting variations - Real-time performance - Cross-validated on multiple datasets
"MediaPipe-Based Yoga Assistant: A Hybrid Approach" (2022)	Li, W., Hassan, M., & Brown, T.	Hybrid system using MediaPipe and lightweight CNN	- Focus on mobile deployment - Real-time feedback system - Memory efficient architecture - Supports 30 common yoga poses

Summary of Literature Survey(LS)

1. Hybrid and Ensemble Approaches

- **Anderson et al. (2023)** proposed an **ensemble of CNN and KNN**, leveraging **MediaPipe features** for improved **real-time performance** and robustness to **lighting variations** across multiple datasets.
- **Li et al. (2022)** developed a **hybrid model** using **MediaPipe and lightweight CNN**, optimizing it for **mobile deployment** with a **memory-efficient architecture** that supports **30 yoga poses**.

2. KNN-Based Solutions for Lightweight Recognition

- **Sharma et al. (2023)** implemented a **KNN classifier** with **MediaPipe landmarks**, achieving **91.2% accuracy** on basic poses. This model is **computationally efficient**, **easy to train on new poses**, and works well with **limited data**.

3. Deep Learning-Based Yoga Pose Recognition

- **Chen et al. (2021)** compared **ResNet, VGG, and custom CNN models**, finding that **ResNet50** performed best with **95.8% accuracy**, though requiring significant computational power.
- **Kumar et al. (2022)** integrated **MediaPipe with a custom CNN** using **MobileNetV2** for **transfer learning**, achieving **94.6% accuracy** and real-time processing at **30 FPS** on a dataset of **82,000 yoga poses**.
- **Zhang et al. (2021)** introduced **YogaNet**, a **multi-stage CNN** with **3D pose estimation**, **occlusion handling**, and a **custom loss function** for improved pose alignment in both **videos and images**.

RV College of Engineering®

System Architect

Go, change the world

The block diagram illustrates the sequential workflow of the **Yoga Pose Tutor System**

1. **Captures Video Feed from a Camera** o The system starts by capturing a real-time video feed from the user's camera. This input serves as the raw data for further processing.
    - o **Technology Used**: Standard webcam or phone camera integration
2. **Uses MediaPipe to Detect Key Points and Preprocess Data** o The captured video is processed using **MediaPipe**, which identifies key points (joints of the body) and extracts the required data for pose estimation. The data is then preprocessed to ensure accuracy in the subsequent steps.
    - o **Technology Used**: MediaPipe framework for key point detection.

System Architect

3. **CNN Extracts Spatial Features from Frames**

A **Convolutional Neural Network (CNN)** is applied to the processed frames to extract spatial features, which capture the structure and position of the user's body in the image. These features are critical for understanding body alignment.

- **Technology Used**: CNN architecture for feature extraction.

4. **KNN Classifier Predicts the Yoga Pose** ○ The extracted features are passed to a **KNearest Neighbors (KNN)** classifier. This machine learning algorithm predicts the user's yoga pose based on pre-trained data.

- **Technology Used**: KNN classifier for pose prediction.

5. **Provides Feedback Based on Accuracy Thresholds** ○ The system evaluates the predicted yoga pose against predefined accuracy thresholds and provides feedback to the user. This feedback ensures that the user can correct their posture and improve their practice. ○ **Technology Used**: Threshold-based feedback mechanism.

Methodology

Step 1: Data Collection

To create a diverse and comprehensive dataset of yoga poses performed by different individuals under various conditions.

Input:

- Videos and images of individuals performing yoga poses.
- Different environments (indoor, outdoor) and lighting conditions to ensure dataset robustness.
- Variations in body types, angles, and positions for better generalization.

1. **Pose Selection:**
   - Collected five yoga poses: **Tadasana, Paschimottanasana, Padmasana, Padhastasana, Vajrasana**.Ensured the poses are performed correctly by referencing standard yoga postures.
2. **Data Capture:**
   - Used cameras (webcam or smartphone) to record individuals performing each pose.
   - Captured images and frames from videos to increase dataset size.
   - Considered variations in pose execution to make the model adaptable to real-world conditions.
3. **Data Organization:**
   - Created labeled folders for each yoga pose to categorize images correctly.
   - Converted video sequences into individual frames for further processing.
   - Stored metadata (e.g., angle, lighting conditions, participant details) for further analysis.

Output: A dataset containing **thousands of labeled images** of yoga poses captured from various angles and environments.

Methodology

Feature Extraction Using Mediapipe

- **Input:** Collected yoga pose images and videos.
- **Process:** Use Mediapipe to detect 3D body landmarks (shoulder, elbow, wrist, hip, knee, ankle) and extract key points. Normalize the coordinates for consistency.
- **Output:** Structured data containing 3D landmark coordinates for each pose.

Data Preprocessing

- **Input:** 3D landmark coordinates from the previous step.
- **Process:** Convert 3D landmarks into 2D format, focusing on key joint positions and angles. Apply scaling and centering to improve model input quality.
- **Output:** Processed and normalized landmark data ready for model training and classification.

Workflow

Pose Detection:

- Mediapipe extracts 33 pose landmarks from video frames.
- Includes x, y, z coordinates and visibility score.

Data Preparation:

- Converts video data into structured image datasets.
- Frame resizing and saving for consistent processing.

Model Prediction:

- KNN model predicts pose classes based on input features.
- Label encoder maps predicted values to readable class labels.

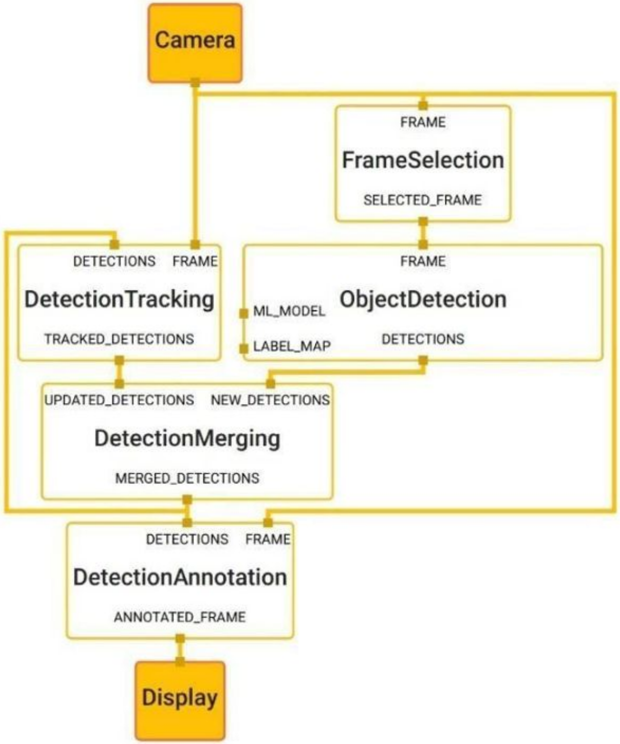
Output Visualization:

- Pose landmarks overlaid on the video stream.
- Real-time display of predicted pose class.

Data Handling and Preprocessing

Input: Video data from camera or dataset files.

1. Data Extraction:
2. Pose Detection Module: Detects key points using MediaPipe.
3. Feature Extraction Module: Extracts spatial features using CNN.
4. Pose Classification Module: Combines tem] and spatial features for classification using LSTM. Feedback Module: Compares user p against thresholds and provides real-time corrections. Assumptions Mad



Model and Labeling

KNN Classifier:

- Trained on extracted pose data (landmark features).
- Handles multi-class prediction of human poses.

Label Encoder:

- Maps numerical predictions to human-readable class names.
- Example: 0 → padhastashan.

Results and Output

1. Achieved **high accuracy** in pose prediction during testing.
2. Real-time predictions with clear visualizations of pose landmarks.
3. Dataset contains structured and labeled image data for various postures.

```
accuracy_score(y_test,y_predict)
```

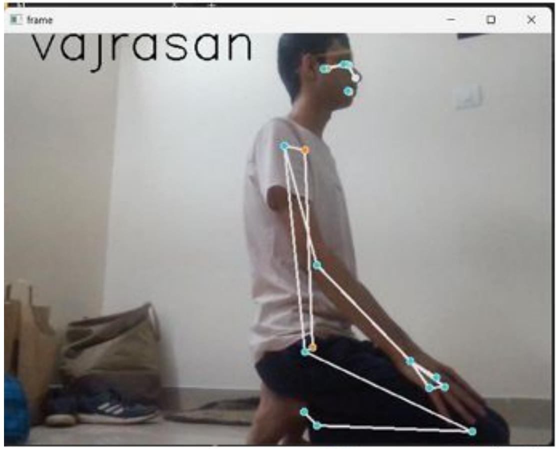
```
0.9955969955969955
```

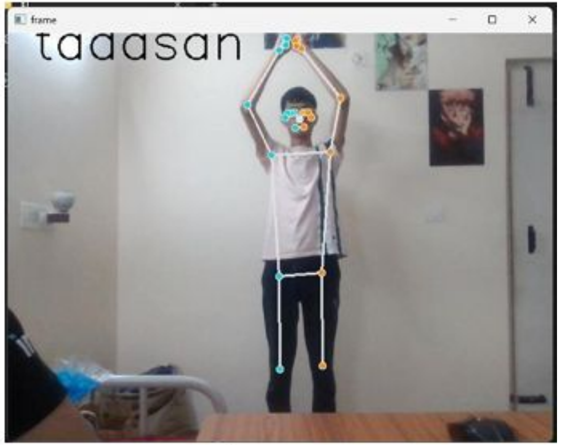
```
pickle.dump(KNN,open("KNN_POSTURE_MODEL.pkl","wb"))
pickle.dump(le,open("label_encoder.pkl.pkl","wb"))
```

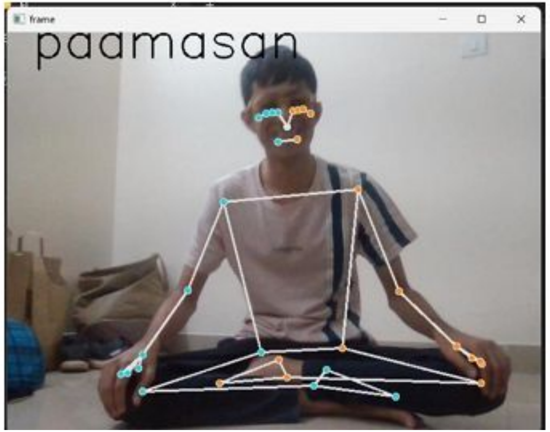
RV College of Engineering®

Go, change the world

Result and Output







Future and scope

1. **Dataset Expansion:** Include more yoga poses from diverse environments for better generalization.
2. **Improved Pose Estimation:** Use advanced frameworks to handle overlapping body parts.
3. **Portable Device Integration:** Implement real-time feedback on wearables and standalone apps.
4. **Multi-Person Detection:** Enable assessment for group settings like yoga classes.
5. **Cross-Domain Applications:** Adapt for sports training, healthcare, and AR-based feedback.

Conclusion

This project presents a real-time yoga monitoring system using CNN and MediaPipe for accurate pose detection. It captures body coordinates, processes image sequences, and evaluates them with a SoftMax classifier to provide real-time feedback. The system improves pose accuracy, alignment, and engagement while reducing injury risks. Its efficient threshold mechanism ensures precise feedback, supporting future applications in rehabilitation and fitness training.

Thank you