A Mini Project Report

On

AsanMLze

DECODING YOGA PRECISION WITH THE POWER OF VISION AND ML

By

Saniya Satish Bhosale (05)

Tejas Kailas Deshmukh (11)

Rajeshwari Vilas Jadhav (22)

Swadesh Nitin Jadhav (23)

Under the guidance of

Prof. Rupali M. Bora

Department of Information Technology

K. K. WAGH EDUCATION SOCIETY'S

K. K. WAGH INSTITUTE OF ENGINEERING EDUCATION AND RESEARCH

Hirabai Haridas Vidyanagari, Amrutdham, Panchavati, Nashik,
Maharashtra 422003

(2024-25)

ABSTRACT

This mini-project introduces a cutting-edge deep learning-based system designed for the classification and evaluation of yoga poses using computer vision and artificial intelligence. With the increasing demand for accessible, at-home wellness and fitness solutions, this project addresses a significant gap by enabling real-time, accurate pose assessment through a webcam. Leveraging TensorFlow's MoveNet pose estimation model and a custom-trained neural network classifier, the system is capable of identifying five fundamental yoga poses and providing detailed feedback on posture correctness based on specific joint angle ranges.

The model is trained on a curated dataset and achieves an impressive classification accuracy of **95.32%**. The real-time system provides feedback to guide practitioners toward safer and more accurate poses, making yoga more accessible and reducing the reliance on physical instructors. Additionally, the project features a rules-based evaluation mechanism that checks for the correctness of specific joint alignments, improving user learning and performance.

CONTENTS

PROBLEM STATEMENT

The global shift towards digital fitness solutions has made it essential to create tools that support personal fitness practices like yoga without requiring constant supervision. While many fitness applications focus on routine tracking or calorie estimation, very few focus on the correctness of form—especially in yoga, where improper posture can lead to strain or injury. Most existing solutions either require wearable sensors or expensive depth cameras, limiting accessibility.

This project aims to fill this void by using a vision-based, deep learning-powered yoga pose evaluation tool that eliminates the need for any additional hardware beyond a webcam. It provides accurate pose classification and granular feedback on form, empowering users to self-correct and enhance their practice with minimal cost and high effectiveness.

METHODOLOGY

The proposed yoga pose evaluation system integrates a deep learning-based classifier for pose recognition and a rule-based evaluation mechanism for assessing pose correctness. This hybrid methodology enables both accurate classification and real-time feedback on posture alignment. The methodology adopted is structured into the following major components:

2.1 Pose Estimation using MoveNet

The **MoveNet SinglePose Lightning** model from TensorFlow Hub was selected for its exceptional real-time inference speed and precision. This model detects 17 anatomical keypoints corresponding to crucial human joints:

- Keypoints include: nose, eyes, ears, shoulders, elbows, wrists, hips, knees, and ankles.
- Each keypoint output contains three elements: **x-coordinate**, **y-coordinate**, and a **confidence score**, indicating the model's certainty in its prediction.

This real-time pose estimation forms the foundation for feature extraction and subsequent classification tasks.

2.2 Feature Extraction and Engineering

From each frame or static image processed by MoveNet, a feature vector of **51 elements** is derived:

- 17 keypoints × (x, y, confidence) = 51 features per sample.
- The coordinates are normalized with respect to the image dimensions for consistency across varying resolutions.
- The features encapsulate both the position and confidence of the detected keypoints, preserving critical spatial relationships for pose classification.

2.3 Dataset Preparation and Label Encoding

To train and validate the classification model, a **custom dataset** was constructed. The dataset comprises labeled yoga pose images across **five distinct categories**:

- **Downward Dog**
- **Goddess**
- **Plank**
- **Tree**
- **Warrior II**

Steps taken during dataset preparation:

- Data collection was done using a combination of online resources and self-curated image frames.
- Each image was manually annotated with the correct pose label.
- A LabelEncoder was employed to convert categorical labels into numerical format for training, and the encoder was serialized into a .pkl file for consistent inference.
- **Data augmentation techniques** (such as horizontal flipping, scaling, rotation) were applied to increase dataset variability, prevent overfitting, and enhance model robustness across body types and orientations.

2.4 Deep Learning Model Architecture

A fully connected feed-forward neural network was implemented to classify the input feature vectors. The architecture includes:

- **Input Layer**: 51 nodes corresponding to the extracted features.
- **Hidden Layers**:
  - First dense layer with 128 neurons and **ReLU activation**.
  - Second dense layer with 64 neurons and **ReLU activation**.
  - Dropout regularization (rate = 0.3) applied to prevent overfitting.
- **Output Layer**: 5 neurons with **softmax activation**, corresponding to the five yoga poses.
- **Optimizer**: Adam optimizer, chosen for its adaptive learning rate and efficiency.
- **Loss Function**: Categorical Cross-Entropy, appropriate for multi-class classification.

The model was compiled using TensorFlow and trained with early stopping to avoid overfitting.

2.5 Training Procedure

The training process involved the following:

- **Data split**: 80% of the dataset was used for training, and 20% for validation.
- **Batch size and epochs**: Optimized through experimentation; early stopping was used to halt training upon convergence.

- **Model Checkpointing**: The best-performing model (based on validation accuracy) was saved in .keras format for deployment.
- The final trained model achieved a **validation accuracy of 95.32%**, demonstrating its high reliability in real-time classification.

This trained model, combined with the saved label encoder, forms the backbone of the yoga pose recognition pipeline.

2.6 Rule-Based Evaluation Logic for Pose Correctness

To assess not just whether a pose is identified correctly, but also whether it is performed correctly, a rule-based evaluation layer was introduced. This component uses **trigonometric angle calculations** between critical joint triplets:

- Angles are computed using the cosine rule from three connected keypoints.
- Each pose has a **set of predefined rules** specifying:
    - Joint triplets involved (e.g., shoulder-elbow-wrist, hip-knee-ankle).
    - Expected **angle ranges** that represent correct posture alignment.
    - **Textual feedback** (e.g., “Arms should be straight”) if the angle deviates from the expected range.

Examples of rules include:

- **Downward Dog**: Arms should be straight (160°–180°), body forming an inverted “V” (90°–130° at hips).
- **Goddess Pose**: Knees bent at ~90°, arms horizontal at shoulder level.
- **Tree Pose**: Raised leg angle between 10°–60°, standing leg near 180°.
- **Warrior II**: Front knee bent at 85°–115°, back leg straight.
- **Plank**: Torso aligned (160°–180° from shoulders to feet), arms perpendicular (75°–105° at elbow).

Each user attempt is validated in real-time against these conditions and provided with feedback, facilitating correction and improvement.

2.7 Deployment and Integration

While the current system is implemented in Python using TensorFlow and OpenCV, the architecture is modular and ready for integration into mobile or web platforms. The .keras model and .pkl label encoder support efficient inference pipelines and can be deployed using frameworks like TensorFlow Lite or Flask-based APIs.

2.8 Innovations Over Existing Approaches

Compared to existing pose recognition tools, our system offers:

- **Combined classification and evaluation**, not just recognition.
- **Real-time personalized feedback** using angle-based logic.
- **High accuracy (95.32%)** due to domain-specific training and augmentation.
- **Custom rule engine** adaptable for other physical activities (e.g., physiotherapy, workouts).
- **Scalability**: Easy to extend to more poses with rule definition and training samples.

MODEL IMPLEMENTATION

The proposed yoga pose evaluation system is implemented as a full-stack application combining a trained deep learning model with a Flask-based web interface. This implementation allows users to upload images or videos or use a live webcam to receive real-time feedback on their yoga pose performance.

3.1 Flask-Based Frontend Integration

The user interface offers three core functionalities:

1. **Live Camera Feed**:
    - Captures frames using a webcam.
    - Streams the video to the backend in real time.
    - Each frame is processed, classified, and evaluated for correctness.
    - Results are displayed live with labels and joint correctness feedback.
2. **Image Upload**:
    - Users can upload a single image of themselves in a yoga pose.
    - The backend processes the image, performs pose classification and angle evaluation, and returns a detailed report.
3. **Video Upload**:
    - Short yoga pose videos can be uploaded for batch processing.
    - The system extracts frames at fixed intervals, evaluates each frame, and generates a cumulative feedback report.

3.2 Model Execution and Feedback Pipeline

The core functionality operates as follows:

- **Pose Estimation**:
    - The uploaded media is passed through **MoveNet**, which outputs 17 keypoints per human pose.
- **Pose Classification**:
    - The extracted keypoints are normalized and flattened into a 51-feature vector.
    - This vector is passed into the trained neural network model loaded from a .keras file.
    - The predicted label is decoded using the .pkl label encoder to map to pose names like *Tree*, *Plank*, etc.
- **Pose Evaluation**:
    - Based on the predicted pose, corresponding **evaluation rules** are triggered.
    - These rules calculate angles (using the cosine rule) between joint triplets (e.g., shoulder-elbow-wrist).

    - Angle ranges are checked to determine correctness, and descriptive feedback is generated in real time.
- **Visualization**:
    - Pose landmarks and skeleton are drawn on the media.
    - Errors or misalignments are highlighted in red, and correct joints are marked green.
    - Feedback is shown directly below the video or image pane.

3.3 Technical Achievements

- **Model Accuracy**:
    - The classifier achieved a **validation accuracy of 95.32%**, highlighting the robustness of the dataset, augmentation strategy, and model design.
    - The model was saved in .keras format for lightweight and portable deployment.
- **Reusable Components**:
    - The **LabelEncoder** file (.pkl) ensures consistent label decoding during both training and inference.
    - Rule-based logic is modular and can be extended to new poses or corrective conditions.
- **Real-Time Response**:
    - Optimized processing ensures real-time feedback on live streams.
    - Parallel frame capturing and processing enhance performance.

CONCLUSION

This yoga pose evaluation system delivers a powerful combination of deep learning classification and rule-based feedback to assist users in improving their yoga practice. Unlike traditional pose estimation tools which stop at detecting body landmarks, this system pushes further by offering personalized evaluations based on anatomical correctness.

Key achievements of the project include:

- Implementation of a custom neural network with **95.32% accuracy** for pose classification
- Robust integration with MoveNet for precise landmark detection
- Development of a rule-based angle evaluator for detailed form correction
- Lightweight and deployable solution without needing additional hardware

Compared to existing projects, this system uniquely offers **both classification and real-time feedback**, bridging the gap between mere detection and meaningful, actionable correction. Its modular design also allows future expansion to additional poses or integration into larger fitness platforms. This project lays a strong foundation for AI-assisted personal fitness applications focused on correctness, wellness, and injury prevention.