Real-Time Yoga Pose Classification and Dynamic Sequence Recommendations Using CNN-Based Computer Vision

Justin Mao Sophia Zou Saanvi Arora
Rithvik Srinivas William Brownstead

April 30, 2025

Abstract

This paper presents an AI-driven yoga assistant that combines real-time pose recognition with dynamic pose sequence recommendations through a lightweight computer vision pipeline. Utilizing the Yoga-82 dataset and fine-tuning a MobileNetV2 classifier, our system achieves 78.4% accuracy on the test set while ensuring balanced performance across different pose classes by applying data cleaning, class rebalancing, and augmentation techniques. To enable real-time feedback, we integrate the model with live webcam input, implementing logic for detecting pose correctness over time and suggesting appropriate next poses based on the selected difficulty level. The pipeline is optimized for CPU performance, making it accessible to users without the need for specialized hardware. The result is a responsive, pose-aware yoga assistant designed for home practice and personal use.

1 Introduction

With yoga growing in popularity as a form of exercise, stress relief, and mindfulness, there's a rising desire for technology that allows users to bring the yoga class into their living room. While options such as hiring a personal trainer, watching YouTube videos, or following a list of poses all currently exist, they are either expensive or offer no interactivity with the user. Thus, a free, easy-to-use method for doing yoga at home and getting real-time feedback on the accuracy of a given pose offers a great opportunity to apply artificial intelligence and machine learning.

In this project, we aim to develop an AI-powered system that can recognize yoga poses from a live video feed and recommend logical next poses to create a fluid, engaging experience. By leveraging efficient computer vision models and real-time inference techniques, we hope to deliver a more personalized and interactive yoga practice accessible to users at all skill levels.

2 Related work

Prior research on automated pose classification has laid the foundation for building intelligent yoga assistants. One of the most comprehensive datasets in this space is the Yoga-82 dataset introduced by Verma et al. [1], which consists of 82 distinct yoga pose classes collected from web images. Their work highlighted the challenges of fine-grained human pose classification, especially in datasets with high intra-class variance and inter-class similarity. We adopt a curated subset of this dataset to address its imbalance and noise for real-time use.

In addition to Yoga-82, smaller datasets like the Yoga-16 dataset on Kaggle [2] have emerged, offering simplified pose categories more suited to limited-resource training. However, these datasets often lack consistent labeling and sufficient diversity, limiting their applicability to general-purpose models.

Recent progress in deep learning has improved pose classification capabilities, especially through the use of lightweight convolutional architectures. Liu et al.[3] proposed ConvNeXt, a modern ConvNet architecture that competes with transformers in vision tasks. While powerful, ConvNeXt was computationally demanding in our experiments, which led us to favor more efficient architectures like MobileNetV2[5]. MobileNetV2's use of depthwise separable convolutions and inverted residuals make it well-suited for real-time inference on CPU-bound devices.

3 Methods

3.1 Dataset: Yoga-82

We train our classifier using a curated subset of the Yoga-82 dataset [1], which contains 82 yoga pose categories collected from web images. We select 12 of the most populated and distinct classes that also overlap with other datasets such as Yoga-16 [2].

To prepare the dataset, we discard broken or inaccessible image links and remove corrupt files. After cleaning, each pose class contains between 150 and 650 images. The dataset is split into 70% training, 15% validation, and 15% testing.

To promote angle invariance and increase diversity, we apply a horizontal flip augmentation with probability 0.5 during training. Many samples featured practitioners facing a specific direction, and this augmentation helps the model generalize to mirrored versions of poses. Qualitative results show improved robustness to camera perspective after applying this technique.

3.2 Pose classification with MobileNetV2

We fine-tune the MobileNetV2 architecture [5] for multi-class yoga pose classification. MobileNetV2 is chosen for its efficiency on CPU devices due to its use of depthwise separable convolutions and inverted residuals. We train the model for 15 epochs: 12 using a learning rate of 10^{-3}, then 3 with a reduced rate of 10^{-4}

for fine-tuning. Optimization is done using AdamW [4], following best practices demonstrated in ConvNeXt [3].

To mitigate class imbalance, we apply class-weighted cross-entropy loss. Each class is assigned a weight inversely proportional to its frequency in the training data, normalized by the total number of classes:

$$w_c = \frac{N}{n_c \cdot C}$$

where w_c is the weight for class c, N is the total number of training samples, n_c is the number of samples in class c, and C is the total number of classes. This normalization ensures that the weights remain consistent across different datasets with varying numbers of classes, and that underrepresented classes contribute more significantly to the loss and gradient updates without disproportionately affecting the model's training process.

An initial attempt to train a ConvNeXt-based model from scratch resulted in poor real-time performance (10 FPS on Apple M1 CPU) and generalization issues due to dataset noise. In contrast, MobileNetV2 achieved 30 FPS and significantly better transfer learning results on our dataset.

3.3 Live pose detection and next-pose recommendation

We integrate the trained model into a real-time pipeline using live webcam input. The system instructs the user to perform a pose, detects when it is correctly held, and then recommends the next pose based on a chosen mode.

At startup, users select one of three modes: • **Standing-Only**: upright poses only • **Hard**: more physically demanding poses • **Class**: a fixed, sequential progression of all 12 poses

Once a mode is selected, a target pose is chosen (either randomly or in sequence), and the system begins classification using the live video feed. To reduce flicker and increase reliability, we consider a pose as "achieved" if it appears in 5 out of the last 8 model predictions and is among the top-2 predicted classes.

When the target pose is confirmed, a green border is shown to the user. After a hold duration, the next pose is selected based on the current mode. The process then repeats, creating a responsive, guided yoga session.

4 Results

We evaluated our model using four key metrics: accuracy, precision, recall, and F1-score. On the test set, our classifier achieved 78.4% accuracy, 79.0% precision, 78.4% recall, and a 78.4% F1-score. These metrics are summarized in Table 1, along with training performance. The similarity between training and testing metrics suggests the model generalizes well to unseen data.

Training and validation loss curves over 15 epochs are shown in Figure 1. While both curves decrease overall, validation loss exhibits more fluctuation, and

	Accuracy	**Precision**	**Recall**	**F1-score**
Training	78.7%	79.1%	78.7%	78.7%
Testing	78.4%	79.0%	78.4%	78.4%

Table 1: Training and testing results at the final epoch.

begins diverging from training loss after around epoch 8—likely due to noise in the dataset. After reducing the learning rate at epoch 12, both curves stabilize.

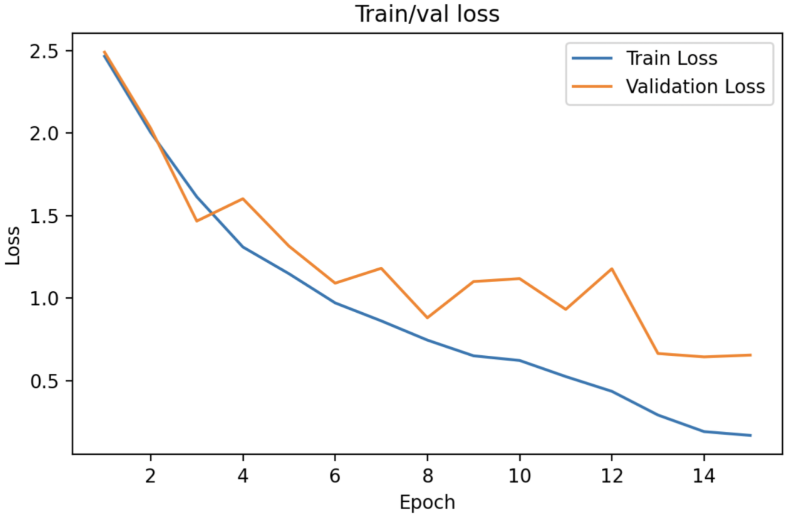

Figure 1: Training and validation loss over 15 epochs.

5 Conclusion

In this project, we successfully developed an AI-powered system capable of identifying yoga poses, providing real-time feedback, and suggesting subsequent poses to guide users through personalized yoga routines. By optimizing MobileNetV2 for CPU performance and incorporating techniques such as class weighting and data augmentation, our classifier achieved a test accuracy of 78.4% while maintaining balanced performance across classes. Integration with a live video feed enabled real-time pose recognition and dynamic sequencing based on user-selected difficulty modes.

This pipeline also provides a foundation for future extensions, such as user-specific pose correction using skeletal angle analysis. Overall, the system demonstrates an effective and accessible approach to delivering intelligent yoga instruction outside of traditional studio settings.

References

[1] Verma, M., Kumawat, S., Nakashima, Y., & Raman, S. (2020). Yoga-82: A new dataset for fine-grained classification of human poses. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 4472–4479.

[2] Kaggle user mohiuddin2531. (n.d.). Yoga-16 dataset. Retrieved from `https://www.kaggle.com/datasets/mohiuddin2531/yoga-16`

[3] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. *arXiv preprint arXiv:2201.03545*. Retrieved from `https://arxiv.org/pdf/2201.03545`

[4] Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

[5] Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *arXiv preprint arXiv:1801.04381*.

Appendix A: Code Availability

The source code used in this study is publicly available at:

`https://github.com/JM5064/yoga-pose`