BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2026
http://excel.fit.vutbr.cz

Personalized Pose Detector for Yoga and Similar Sports

Bc. Adriana Buchmei*

Abstract
Standard human pose estimation models trained on pristine datasets often fail on real-world users due to individual flexibility limits and varying camera angles. To address this, this project introduces an edge-ready, hybrid architecture combining a lightweight PyTorch neural network with a personalized 1-Nearest Neighbor database, utilizing robust scale-invariant mathematical normalization. The resulting system accurately classifies complex yoga poses in real-time, allowing users to register their unique bodily variations instantly without retraining the core network. This approach proves that smart geometric preprocessing and training-free personalization can effectively replace heavy, power-hungry CNNs, enabling seamless and offline fitness tracking on mobile devices.

*xbuchm03@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Standard pose detection models are heavily biased by the pristine datasets they are trained on [1], which primarily feature highly flexible individuals. As illustrated in Figure 1, when applied to real-world users with physical constraints or awkward camera angles, these models fail, resulting in unrecognized poses and demotivating feedback.

To resolve this rigid and demotivating behavior, this project introduces a personalized pose detection system that adapts to the user's specific physical limits. Current state-of-the-art methods typically rely on heavy Convolutional Neural Networks (CNNs) to extract spatial features directly from image pixels. While highly accurate, these models are computationally expensive for mobile environments, and accommodating a user's bodily constraint requires a lengthy retraining process.

Instead of a deep CNN, this solution employs a hybrid pipeline relying on strict geometric preprocessing. Key contributions include: 1) A hybrid pipeline utilizing local database storage to create a personalized detector via training-free updates, 2) An AI Inference Visualizer for continuous analysis of class probabilities, and 3) An integrated Active Learning tool designed to capture edge cases and systematically improve the dataset.

2. Architecture and Core Components

The poster visually traces the data lifecycle from a raw video feed to a stable, personalized AI prediction.

2.1 Data Extraction and Normalization

The process begins with the raw input video, illustrated in Figure 3. Frame by frame, the video is processed by the MediaPipe Pose Landmarker [3], which extracts 33 3D joint coordinates alongside their visibility scores. As shown in Figure 4, this results in a video frame overlaid with the extracted skeleton landmarks.

Before any classification, the system uses basic math to normalize the skeleton's size. As detailed in Equation 1, L_{torso} is calculated by finding the Euclidean distance between the shoulder and hip centers. Dividing all joint coordinates by L_{torso} makes the skeleton entirely scale-invariant. Inspired by the topological alignment of BlazePose [4], this ensures reliable tracking regardless of the user's distance from the camera.

2.2 The Hybrid Pipeline and Prediction Output

The normalized skeleton first enters the primary engine: the Personal Pose DB. Using Fast Distance Matching (Equation 2), a 1-Nearest Neighbor algorithm [5] calculates the Euclidean distance between the current pose and the user's locally saved variations. If the distance is below a strict threshold ($d < 0.08$), the system instantly recognizes the pose.

If there is no match, the system falls back to the Py-

Torch Neural Network [6]. As illustrated in the green diagram (Figure 5), this is a lightweight 3-layer Fully Connected Classifier ($132 \rightarrow 128 \rightarrow 64 \rightarrow 8$ neurons) terminating with a Softmax activation. Because the input geometry is already normalized, and the network utilizes batch normalization alongside dropout layers, this shallow architecture is perfectly sufficient to learn complex joint angles without overfitting.

The final prediction output consists of the detected pose label, the confidence score, and the body status (stable or moving).

2.3 AI Inference Visualizer

To demonstrate the system's performance, an AI Inference Visualizer processes the video input and displays the pose detection alongside a skeleton overlay. It features a dynamic timeline graph (Figure 6) that tracks fluctuating class probabilities over time, providing transparent visual feedback of the model's decision-making process.

2.4 Active Learning Tool

No dataset covers all real-world scenarios. Following Uncertainty Sampling principles [2], the system includes a Visual Active Learning Tool (Figure 2). Whenever the PyTorch model's confidence drops below 75% while the pose remains stable for at least 3 seconds, the frame is flagged. The pipeline automatically exports the raw image, the skeleton overlay, and JSON metadata into a review folder. This creates an effortless feedback loop, enabling a human reviewer to easily verify the pose and incorporate it into the dataset and local database via an automated script for future training iterations.

3. Implementation for the Edge

Although the current pipeline operates as a Python prototype, the architecture was purposefully designed for fully offline execution within an Android application[1] where users will seamlessly track their fitness progress by logging exercise photographs during their sessions, without interrupting their workout flow. The transition to this native mobile environment is actively underway, with the heavy OpenCV video capture already successfully replaced by the native CameraX API[2]. To guarantee instant pose matching, the prototype's JSON storage is migrating to a low-latency Room Database [7]; the core database structure already exists and simply requires an additional table for skeletal landmarks. Finally, the lightweight PyTorch fallback classifier will be exported to LiteRT [8] to unlock hardware-accelerated mobile inference.

[1] Poseify App
[2] Android CameraX docs

4. Conclusions

This project highlights a practical alternative to heavy, resource-draining AI models. It proves that by using smart geometric alignment alongside a simple neural network and a local database, a pose detector can be both lightning-fast on mobile phones and truly empathetic to the physical constraints of everyday users.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, prof. Ing. Adam Herout, Ph.D., for his expert guidance, valuable feedback, and continuous support throughout the development of this project.

References

[1] Verma, M., Kumawat, S., Nakashima, Y., and Raman, S. *Yoga-82: A New Dataset for Fine-grained Classification of Human Poses*. CVPR Workshops, 2020.

[2] Settles, B. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009.

[3] Google. *MediaPipe Pose Landmarker*. Available at: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker

[4] Bazarevsky, V., et al. *BlazePose: On-device Real-time Body Pose tracking*. CVPR Workshop on Computer Vision for Augmented and Virtual Reality, 2020.

[5] Cover, T., and Hart, P. *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory, 13(1), 21-27, 1967.

[6] Paszke, A., et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. NeurIPS, 2019.

[7] Google. *Room Persistence Library*. 2018. Available at: https://developer.android.com/training/data-storage/room

[8] Google. *LiteRT: High Performance On-Device ML & GenAI Deployment on Edge Platforms*. 2026. Available at: https://ai.google.dev/edge/litert