Real-Time Exercise Form Analysis and Rep Counting

Muhammad Ali Haider

We present a study on pose-based activity recognition focused on yoga pose classification and on rule-based repetition counting for weight-room exercises. For **weight exercises** (bicep curls, shoulder press, push-ups), we implement a finite-state machine (FSM) using **OpenPose** (COCO/MPII models) to extract 2D joints; *we do not train deep learning models* on these exercises due to the lack of a suitable labeled dataset. For **yoga**, we train and evaluate ANN/CNN/VGG16/IRv2/MobileNetV3 (and a YOLOv7-as-classifier exploration) on the five-class **Kaggle Yoga Poses dataset**, comparing RGB images vs. **MediaPipe/BlazePose** skeletonized inputs.

Skeletonization consistently boosts compact backbones (e.g., MobileNetV3) and yields our top accuracy with VGG16+BlazePose. We release our evaluation protocol, confusion matrices, and the FSM implementation for rep counting.

I. Dataset

A. Yoga classification dataset

All deep-learning experiments are conducted on the public **Kaggle Yoga Poses dataset** (five classes: *downdog, goddess, plank, tree, warrior2*). We follow an 80/10/10 split and report results on the held-out test set. For the "skeleton" condition, we extract 2D pose landmarks from the yoga images using MediaPipe/BlazePose and feed either landmark vectors or stick-figure renderings to the classifiers. [1], [2]

B. Weight-exercise streams (no DL training)

For *bicep curls, shoulder press, push-ups*, we do **not** train deep models due to dataset unavailability. Instead, we apply **OpenPose** (COCO/MPII Caffe models) to each frame to obtain joint coordinates and run a finite-state machine for robust rep counting and posture prompts. These exercise streams are used to validate the FSM logic and on-device feasibility rather than to produce classification metrics.

***Index Terms*—Write up to three keywords about your work.**

II. Introduction

Vision-based exercise analysis has two complementary goals in practice: (i) *pose classification* to recognize postures such as yoga asanas, and (ii) *rep counting with form feedback* for repetitive weight-room movements. Direct RGB classification with convolutional networks is effective on large, well-curated datasets but often degrades under background, clothing, or illumination shifts. A widely used alternative is a *pose-first* pipeline: extract 2D landmarks with a pose estimator and classify or score using those landmarks. Landmark-based representations compress away appearance while retaining articulated geometry, which improves robustness in low-data, real-world settings.

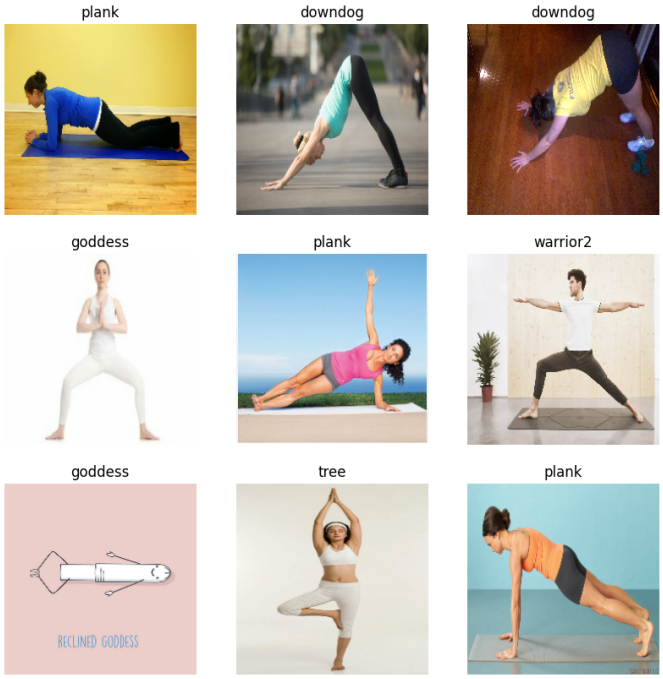


Fig. 1: YOGA Dataset Deep Learning Models

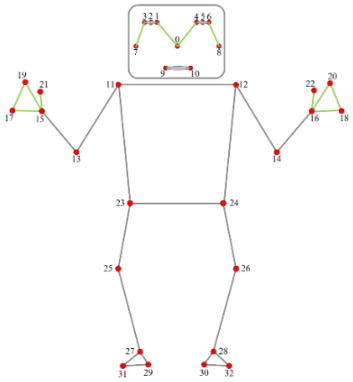
A. Motivation

Our application requires *real-time, on-device* feedback on commodity hardware. We therefore favor backbones designed for mobile CPUs (e.g., MobileNetV3) and single-person pose estimators optimized for speed (e.g., BlazePose), which reports around 30 FPS on a Pixel 2 while predicting 33 landmarks. Landmark ("skeleton") inputs are also attractive because they are more invariant to background and lighting than raw RGB, a property highlighted across recent surveys of skeleton-based action recognition. [**?**], [7]

B. Problem Statement

We study two concrete tasks:

- **Yoga pose classification** over five classes (downward dog, goddess, plank, tree, warrior II) using the public Kaggle dataset with standard train/val/test splits. [1]
- **Weight-exercise rep counting** (bicep curls, shoulder press, push-ups) with stable state transitions and posture prompts. Due to the lack of large, high-quality, labeled



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Fig. 2: Skeletal Points by Blazepose (Mediapipe)

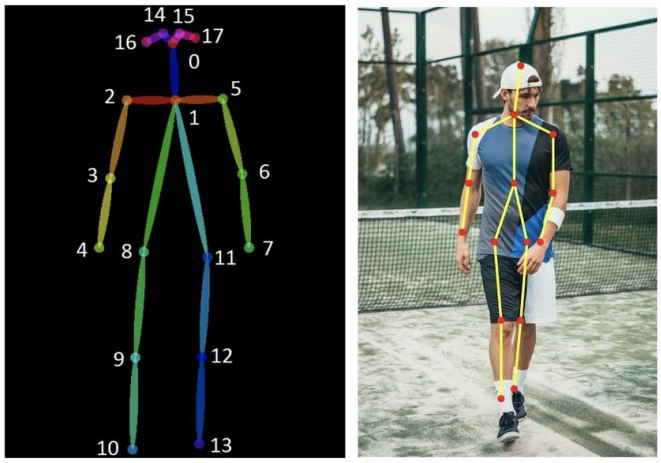


Fig. 3: Openpose Body Points

form datasets within the project window, we avoid deep model training for these and instead use a deterministic pipeline.

C. Proposed Approach

For yoga, we compare baselines (ANN/CNN) with transfer-learning backbones (VGG16, Inception–ResNet–v2, MobileNetV3), training on RGB and on *skeletonized* inputs derived from BlazePose. For weight exercises, we compute joint angles from OpenPose keypoints and drive a two-state finite-state machine (DOWN↔UP) with hysteresis for robust rep counting and simple form cues. [**?**], [3]–[5], [17]

III. Related Work

Pose estimation. OpenPose introduced Part Affinity Fields (PAFs), a bottom-up representation that detects body parts and associates them into multi-person skeletons in real time; the journal version provides extended analysis and benchmarks. For single-person tracking on mobile devices, BlazePose predicts 33 landmarks and demonstrates on-device, real-time performance suitable for fitness feedback. [**?**], [5]

Backbones for classification. VGG16 showed that increasing depth with 3×3 convolutions yields strong ImageNet features transferable to downstream tasks. Inception–ResNet–v2 combined Inception modules with residual connections, improving optimization and accuracy. MobileNetV3 used hardware-aware neural architecture search and architectural tweaks (e.g., h-swish, SE) to deliver better latency–accuracy trade-offs for mobile deployment. [3], [4], [6]

Skeleton-based action recognition. Surveys report that skeleton representations increase robustness to appearance and background variations, and enable efficient models for recognition and temporal reasoning—properties desirable for real-world exercise analysis. [7]

Yoga datasets. The Kaggle *Yoga Poses* dataset provides five canonical asanas (downward dog, goddess, plank, tree, warrior II) used widely for benchmarking pose classification; we adopt it for reproducibility and comparability across backbones. [1]

IV. Background

Vision-based human pose analysis commonly follows two paradigms. **(i) Direct RGB classification:** a convolutional network predicts a pose label from pixels. **(ii) Pose-first ("skeletonization"):** a pose estimator extracts 2D/3D landmarks which a downstream model (or rules) uses for recognition and feedback. Skeleton representations are attractive on edge devices because they compress away appearance and background while retaining body geometry; the literature repeatedly notes their *robustness to illumination, viewpoint, and background clutter* and their computational efficiency compared with raw RGB/video. [8]–[10]

Pose estimators used in this work.: OpenPose introduced *Part Affinity Fields (PAFs)*, a bottom-up representation that detects parts and learns association fields to assemble full multi-person skeletons with strong accuracy and real-time performance on public benchmarks. [11], [12] For single-person, on-device tracking, *BlazePose* regresses **33 body landmarks** (x,y,z) and runs at ∼30 FPS on a Pixel 2, enabling real-time fitness applications.

Classifier backbones.: We compare standard backbones that span the accuracy–efficiency spectrum. *VGG16* showed that pushing depth to 16–19 layers with 3×3 filters yields strong transferable features on ImageNet. [6] *Inception-ResNet-v2* combines Inception modules with residual connections, accelerating optimization and improving accuracy at relatively modest compute. [13] *MobileNetV3* was designed via hardware-aware neural architecture search and architectural tweaks (e.g., SE, h-swish), achieving superior latency–accuracy trade-offs for *mobile CPUs*. [4]

Rule-based rep counting.: For repetition counting in weight exercises (bicep curls, shoulder press, push-ups) we adopt a classical pipeline: OpenPose keypoints → joint angles → a two-state finite-state machine (DOWN↔UP) with hysteresis for stability. This leverages well-established pose estimation while avoiding supervised training when quality, labeled datasets are unavailable. [11], [12]

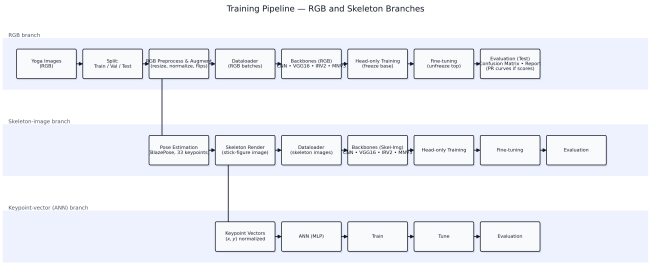


Fig. 4: Training Pipeline

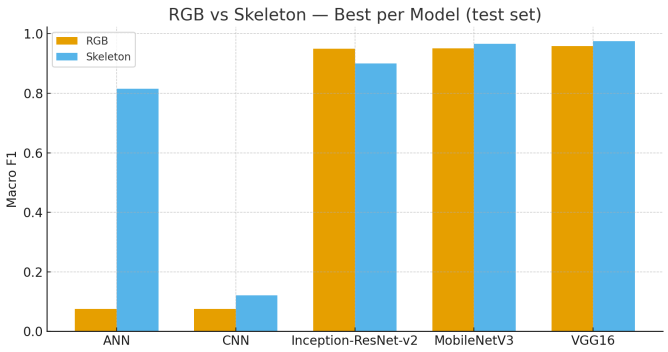


Fig. 5: RGB vs Skeletal Training differences

A. Overview

We implement two complementary tracks: (i) *pose classification* for five yoga asanas using deep models trained on RGB and on *skeletonized* inputs; and (ii) *real-time rep counting* for bicep curls, shoulder press, and push-ups using a finite-state machine (FSM) driven by 2D pose keypoints.

B. Data and Splits

For yoga, we use the public Kaggle *Yoga Poses* dataset (five classes) and create train/validation/test folders. Results are reported on the held-out test set only. [1]

C. Skeletonization

We extract 33 landmark keypoints per frame with MediaPipe BlazePose and build two inputs: (1) a 66-D normalized keypoint vector for MLPs; (2) a clean skeleton rendering (lines/joints on a blank canvas) for CNN/transfer backbones. [**?**]

D. Models and Training

We compare: (a) a compact ANN on keypoint vectors and a small CNN on images; (b) transfer learning with VGG16, Inception–ResNet–v2, and MobileNetV3 (ImageNet initialization). We train a new classification head, then fine-tune upper blocks with a lower learning rate. Training uses Adam in TensorFlow/Keras with early stopping and standard, light augmentations (flip/rotate/translate). [3], [4], [6], [14]–[16]

E. FSM Rep Counting

For weight exercises, we avoid deep training (dataset constraints) and compute joint angles from an OpenPose style estimator (COCO/MPII keypoints). A two-state FSM with hysteresis (UPDOWN) increments the counter on full cycles and triggers simple form cues. [5], [17]

F. Metrics

We report accuracy, per-class precision/recall/F1, macro/weighted averages, and confusion matrices; we also export per-image predictions and all summaries to CSV for reproducibility. [18], [19]

V. EVALUATION

A. Experimental Setup

Tasks. (i) Five-class yoga pose classification (`downdog`, `goddess`, `plank`, `tree`, `warrior2`) on the public Kaggle dataset [1]; (ii) real-time rep counting for bicep curls, shoulder press, and push-ups using a keypoint-driven FSM.
Data splits. We use a hold-out split into train/validation/test and report all final numbers on the test set.
Representations. RGB images vs. *skeletonized* inputs: 33-landmark 2D keypoints from MediaPipe BlazePose [**?**] either (a) concatenated as a 66-D vector (ANN) or (b) rendered as a clean skeleton image (CNN/transfer). For the FSM, 2D keypoints are obtained via an OpenPose-style Part Affinity Fields (PAF) estimator [5], [17].
Models. ANN (MLP), compact CNN, and transfer models (VGG16, Inception–ResNet–v2, MobileNetV3). We train a new classification head, then fine-tune the top block(s) at a reduced learning rate.
Training. TensorFlow/Keras with Adam, early stopping on validation accuracy, light augmentations (flip/rotate/translate).
Metrics. Accuracy and per-class precision/recall/F1 (macro and weighted averages) via `sklearn.metrics.classification_report`; confusion matrices via `sklearn.metrics.confusion_matrix` [**?**], [19].

B. Main Results

Tab. I summarizes test-set performance across models and input types. Consistent with the literature on skeleton-based recognition, the **skeletonized variants outperform RGB** across backbones, indicating improved robustness to background/appearance variation while retaining pose geometry.

C. Per-Class Behavior

Confusion matrices (one per best model) reveal typical confusions among visually similar poses (e.g., `warrior2` vs. `goddess`); skeleton inputs reduce background-induced errors and sharpen the diagonal. See Appendix for full matrices and per-class reports (CSV exports).

TABLE I: Yoga classification results on the test set. Skeletons from BlazePose; ANN/CNN are baselines.

Model	Accuracy	Macro-F1	Macro-Precision	Macro-Recall
VGG16 + BlazePose	0.9745	0.9750	0.9756	0.9745
MobileNetV3 (skeletons, head)	0.9656	0.9666	0.9673	0.9662
MobileNetV3 (skeletons, finetune)	0.9613	0.9625	0.9634	0.9619
VGG16 (RGB)	0.9596	0.9590	0.9619	0.9566
MobileNetV3 (RGB, finetune)	0.9511	0.9506	0.9523	0.9493
IRv2 (RGB)	0.9489	0.9493	0.9482	0.9516
MobileNetV3 (RGB, head)	0.9404	0.9396	0.9449	0.9360
IRv2 (skeletons)	0.9000	0.9004	0.9035	0.9109
ANN (skeletons)	0.8108	0.8151	0.8278	0.8098
CNN (skeletons)	0.2581	0.1205	0.2828	0.2233
CNN (RGB)	0.2319	0.0753	0.0464	0.2000
ANN (RGB)	0.2298	0.0747	0.0461	0.1982

D. Runtime and Practicality

Skeleton pipelines remain light enough for real-time feedback: BlazePose runs at >30 FPS on mobile devices while producing 33 landmarks [**?**]. The FSM counter is deterministic and low-latency; the classification models with MobileNetV3 backbones are suitable for CPU-only deployment.

VI. Conclusions

A. Summary.

We built a real-time pipeline for exercise form analysis and rep counting, combining classical finite-state machines (FSMs) with 2D pose estimation for weight exercises (biceps curls, shoulder press, push-ups) and supervised deep learning for multi-class yoga pose recognition. Among image backbones, Inception-ResNet-v2 (IRV2) and VGG16 performed strongly on RGB images, while MobileNetV3 offered an attractive accuracy–latency trade-off. Converting images to skeleton renderings (via MediaPipe/BlazePose) substantially improved some simpler models and reduced sensitivity to background, but high-capacity CNNs generally remained better on full RGB.

Why IRV2 is stronger on RGB than on skeletons IRV2 is pretrained on ImageNet natural images and is architecturally tuned for rich appearance cues; thus it leverages textures, shading, and background layout that are absent in stick-figure skeletons [20], [21]. Skeleton inputs are sparse and inherit pose-estimation noise; moreover, state-of-the-art pose pipelines often favor graph-based models over image CNNs when operating on joint coordinates [22]. MediaPipe BlazePose provides dense 33-landmark keypoints that help, but without pose-specific architectures or multimodal fusion, RGB still holds an advantage for IRV2

B. Contributions.

- (i) A practical FSM baseline for rep counting on common weight exercises using commodity 2D pose estimation.
- (ii) a comparative study of RGB vs. skeleton representations across multiple CNN backbones for yoga classification.
- (iii) an engineering recipe (data curation, evaluation scripts, confusion matrices/PR summaries) suitable for reproducible benchmarking.

C. Future Work.

We plan to

- (a) fuse RGB and pose streams to exploit complementary appearance and geometry,
- (b) evaluate pose-native models (e.g., ST-GCN) on joint coordinates instead of rasterized skeletons,
- (c) explore temporal modeling and 3D keypoints for improved robustness.

References

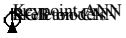


Fig. 6: Training pipeline with RGB and skeleton branches. Matrix layout prevents node overlap; short labels keep boxes compact.

[1] Niharika41298, "Yoga poses dataset," Kaggle, accessed 2025-09-12. [Online]. Available: https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset

[2] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," *arXiv:2006.10204*, 2020.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016. [Online]. Available: https://arxiv.org/abs/1602.07261

[4] A. Howard, M. Sandler, G. Chu, L.-C. Chen, and et al., "Searching for mobilenetv3," *arXiv:1905.02244*, 2019.

[5] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021. [Online]. Available: https://dl.acm.org/doi/10.1109/TPAMI.2019.2929257

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2015.

[7] B. Ren, M. Liu, R. Ding, and H. Liu, "A survey on 3d skeleton-based action recognition using learning method," *arXiv preprint arXiv:2002.05907*, 2020, latest arXiv version accessed 2025-09-12. [Online]. Available: https://arxiv.org/abs/2002.05907

[8] H. Duan *et al.*, "Skeletr: Towards skeleton-based action recognition in the wild," in *ICCV*, 2023.

[9] H. Wang *et al.*, "Understanding the robustness of skeleton-based action recognition under adversarial attack," in *CVPR*, 2021.

[10] B. Ren *et al.*, "A survey on 3d skeleton-based action recognition using deep learning," *Cognitive Computing and Systems*, 2024.

[11] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[12] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE TPAMI*, 2019.

[13] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv:1602.07261*, 2016.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[15] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[16] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 60, 2019. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

[17] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/papers/Cao_Realtime_Multi-Person_2D_CVPR_2017_paper.pdf

[18] scikit-learn developers, "classification_report — scikit-learn 1.7.2 documentation," https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html, accessed: 2025-09-12.

[19] ——, "confusion_matrix — scikit-learn 1.7.2 documentation," https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html, accessed: 2025-09-12.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11231/11090

[21] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *International Conference on Learning Representations (ICLR)*, 2019. [Online]. Available: https://openreview.net/forum?id=Bygh9j09KX

[22] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI*, 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/12328/12187