Received: 31 August 2025 | Revised: 8 January 2026 | Accepted: 22 April 2026 | Published online: 8 May 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–17
DOI: 10.47852/bonviewAIA62027514

BON VIEW PUBLISHING

ResProtoNet: A Skeleton-Aware Few-Shot Framework for Yoga Pose Classification

Chean Khim Toa[1,*], Kai Liang Lew[2] and Sin Pei Ton[1]

[1] *School of Computing and Data Science, Xiamen University Malaysia, Malaysia*

[2] *Faculty of Engineering & Technology, Multimedia University, Malaysia*

Abstract: Deep learning models for yoga pose classification traditionally require large, diverse, carefully annotated datasets, which are costly and time-consuming. While open-source yoga datasets are available, collecting and annotating new ones, especially for complex poses, remains a significant challenge. This limitation motivated the use of few-shot learning (FSL) to enable efficient pose classification under limited data conditions. This study proposes ResProtoNet, combining a ResNet-18 feature extractor with a Prototypical Network classifier, evaluated against a supervised baseline using RGB and skeleton-based images. The latter contains joint-based pose encodings extracted via the MediaPipe model. Experiments conducted on five fundamental yoga poses under 1-shot, 3-shot, and 5-shot configurations demonstrated several findings. First, ResProtoNet consistently outperformed the ResNet-18 baseline, achieving 98.2% accuracy in the 3-shot setting, compared to 97.6%. Second, ResNet-18 consistently delivered the strongest baseline performance across both modalities, underscoring its robustness and justifying its role as backbone. A key contribution is a comprehensive stress test comparing RGB and skeleton modalities within an FSL context. Robustness analysis highlighted that skeleton-based input preserved up to 20% higher accuracy under heavy occlusion and reduced accuracy loss by 5.7% under resolution degradation, confirming resilience when visual information degrades. In contrast, RGB retained advantages under light occlusion and higher-resolution inputs, where texture and background remained informative. Overall, applying FSL to a supervised baseline enables reliable pose classification under limited supervision, substantially reducing dependence on large-scale annotated datasets. ResProtoNet demonstrated strong potential for real-world healthcare and exercise monitoring systems, where annotated resources and data quality are constrained.

Keywords: few-shot learning, human pose estimation, RGB input, skeleton-based input, yoga pose classification

1. Introduction

Human pose estimation (HPE) has emerged as a crucial area in computer vision, with broad applications in healthcare, sports analytics, rehabilitation, and human–computer interaction. The ability to automatically detect and classify human body postures enables systems to provide real-time feedback and guidance, which is particularly valuable in physical activity monitoring, healthcare settings, and training environments [1, 2]. With the growing trend of home-based fitness, there is an increasing demand for systems that can provide posture analysis and corrective feedback without requiring physical supervision. Among various physical activities, yoga has gained popularity across age groups due to its health benefits [3]. However, the correct execution of posture is essential to avoid injuries and to maximize effectiveness [4]. Professional guidance is not always readily available, and online classes often lack real-time, personalized feedback. This creates an opportunity for automated yoga pose classification systems to provide accessible, flexible, and accurate guidance for practitioners.

Recent advances in deep learning and computer vision have enabled the development of automated models for yoga pose classification [5]. These systems rely on convolutional neural networks (CNNs) or related architectures to classify human postures from image or video data. However, existing approaches that rely on raw RGB images face significant challenges due to high variability in body structures, clothing, lighting, and backgrounds [6]. Such factors introduce noise that distracts models from learning structural posture features, resulting in reduced classification accuracy. To mitigate these issues, skeleton-based pose representations have been proposed as a promising alternative [7, 8]. By extracting keypoint landmarks and encoding them as skeletal structures, these representations emphasize spatial and structural relationships while minimizing irrelevant appearance features. MediaPipe, for instance, has demonstrated reliability in generating skeletonized input by detecting 33 body landmarks of the human body [9].

Furthermore, collecting and labeling yoga pose data is time-consuming and resource-intensive and requires expert annotation to capture subtle differences between poses. Training robust deep learning models typically requires thousands of labeled samples that capture variations in poses. Although open-source datasets exist, they are often inconsistent and require extensive filtering, leaving only small amounts of usable data. This becomes more

***Corresponding author:** Chean Khim Toa, School of Computing and Data Science, Xiamen University Malaysia, Malaysia. Email: cheankhim.toa@xmu.edu.my

problematic in small variation pose classification tasks, where small differences between poses demand a large amount of labeled training data. To address the problem, few-shot learning (FSL) has emerged as a promising approach, enabling models to generalize effectively from only a few labeled samples per class [10, 11]. By framing classification as a metric-learning problem, FSL techniques can learn robust representations that transfer knowledge from seen to unseen classes [12].

Nevertheless, FSL studies remain underexplored in the domain-specific problems such as yoga pose classification, where both subtle inter-class differences and intra-class variability pose unique challenges. Another important consideration is the input modality. While RGB images provide rich visual details, they are prone to background noise and appearance variations. In contrast, skeleton-based representations abstract human body keypoints into structural information, potentially offering greater robustness and interpretability.

This work addresses three main objectives. First, to propose ResProtoNet, a skeleton-aware few-shot framework specifically designed to address the dual challenges of limited data and environmental constraints. Unlike generic FSL applications, this framework integrates a geometric pose abstraction pipeline with metric-based learning to ensure reliable classification even when visual information is degraded. Second, to systematically compare RGB and skeleton-based input modalities across several baselines, specifically quantifying their robustness to occlusion and resolution degradation. Third, to assess the input modality trade-offs, identifying the specific conditions under which each modality offers greater advantages for real-world deployment.

These objectives motivate three research questions:

1) Can the integration of FSL enhance yoga pose classification compared to supervised baselines?
2) How do RGB and skeleton-based input modalities differ in feature representation quality, robustness to occlusion and resolution degradation, and classification performance?
3) Under what conditions does each modality (RGB vs skeleton) offer greater advantages?

The paper is organized as follows. The Literature Review section presents related works, including an introduction to HPE, trends in yoga pose classification, and the introduction of FSL. The Methodology section shows the dataset preparation, model architecture, and experimental setup. The result and discussion section reports the performance of classification experiments under a few-shot configuration and baseline, followed by robustness and efficiency analyses of RGB and skeleton-based input modalities. The conclusion section concludes the study and provides recommendations for future work.

2. Literature Review

HPE utilizes computer vision and machine learning to localize keypoints in images and videos, thereby forming a skeletal representation of the body. Modern HPE systems typically output 2D coordinates (x, y) for each joint; some methods also estimate 3D positions with depth (x, y, z) [13]. Recently, HPE has garnered increased attention from various industries due to its wide range of applications across different domains, including video surveillance, medical monitoring, and security systems [14].

A list of HPE applications is presented in Table 1, indicating the growing demand and usage of posture classification in real-life applications [15]. To significantly advance HPE research and deployments, several open-source frameworks were publicly available [16]. OpenPose was the first real-time multi-person 2D pose estimator, providing 18–25 keypoints and achieving robust results on challenging in-the-wild datasets [17]. PoseNet simplified pose estimation by enabling lightweight inference on mobile devices [18]. MediaPipe, developed by Google, introduced a highly efficient deep learning–based model optimized for edge devices, achieving state-of-the-art (SOTA) speed–accuracy trade-offs. It utilizes a BlazePose 33-landmark topology, covering not only major skeletal joints but also finer regions, such as hands, feet, and facial features [9, 19].

2.1. Yoga pose classification

Yoga is a well-known exercise for its numerous health benefits, including stress management, improved muscle strength, and enhanced body coordination. Despite all the attractive benefits of yoga, improper posture can still lead to severe injuries and accidents. Therefore, an increasing amount of research has been conducted in recent years, aiming to utilize the advancements in computer vision and deep learning techniques to improve yoga pose classification and identification. While early systems often relied on cumbersome hardware like depth sensors or wearable Inertial Measurement Units (IMUs) to recognize postures, vision-based deep learning approaches have become increasingly dominant [20]. Typically leveraging CNNs or transfer learning on pretrained models, these modern systems require only standard camera inputs; for example, Gadhvi et al. [5] recently proposed a

Table 1
Applications of HPE in various fields

Field	HPE applications
Sports analytics	Monitoring athlete performance, improving techniques, preventing injuries, and yoga correction
Security systems	Suspicious behavior detection, abnormal posture recognition in surveillance for threat identification
Medical surveillance and rehabilitation	Tracking patient posture during rehabilitation exercises, fall detection, and physiotherapy monitoring
Elderly care	Monitoring posture for fall prevention, activity tracking for health assessment
Human–computer interaction	Controlling devices through body posture and gesture recognition
Fitness and personal training	Providing real-time feedback on exercise posture for home workouts and digital fitness platforms

highly accurate edge-AI solution for posture correction, entirely eliminating the need for specialized external sensors

More recently, skeleton-based approaches have been explored to provide a compact and interpretable representation of human posture. Garg et al. [21] combined MediaPipe keypoint extraction with a CNN to achieve real-time pose recognition, illustrating how modern pipelines that fuse skeleton keypoints with deep learning architectures can yield efficient and robust results. Such methods reduce reliance on raw image data and focus on structural features of the human body, which often improves generalizability across lighting conditions and backgrounds.

Furthermore, several studies have sought to balance recognition accuracy with computational efficiency. For instance, Shourie et al. [22] proposed YogaPoseVision, a lightweight CNN based on MobileNetV3, fine-tuned with transfer learning on a curated dataset of yoga poses. Their results demonstrated strong classification accuracy while retaining efficiency suitable for real-time applications, highlighting the potential of lightweight models for mobile and resource-constrained environments. Similarly, Aruna et al. [23] leveraged a pretrained ResNet-18 architecture with transfer learning, achieving 94.93% test accuracy on a Kaggle dataset. Other studies had focused on maximizing accuracy. Kashyap et al. [24] explored the family of EfficientNet models (B0–B7), demonstrating that EfficientNet-B7 yielded the highest performance with 97.61% accuracy for yoga pose detection. Tayal et al. [25, 26] compared MobileNet, VGG19, and EfficientNet under six different optimizers, reporting that EfficientNet with AdaDelta achieved the strongest results (97.45% accuracy, F1-score 0.98), thereby emphasizing the critical influence of both architecture selection and optimizer choice. Similarly, Rajendran and Sethuraman [27] proposed DensePoseCompare, a comparative framework assessing DenseNet121, DenseNet169, and DenseNet201 across a five-class yoga dataset. Their results revealed that DenseNet201 attained an outstanding 99.07% accuracy, outperforming its counterparts.

Research on yoga poses has emerged as a growing domain, with numerous researchers dedicating considerable effort to this field. Figure 1 shows the number of papers published from 2016 to 2024 through Google Scholar. It clearly indicates a significant increase in interest among the scientific community. Although several more keywords can result in relevant publications, such as "yoga pose grading" and "yoga classification," those are not being included in this research count.

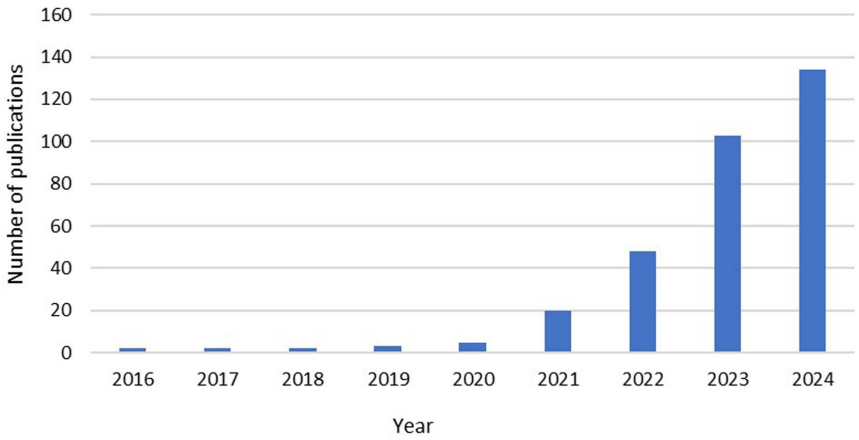
2.2. Data challenges and few-shot learning

A central challenge in yoga pose classification is accurately annotating data. Verma et al. emphasize that prior datasets for pose classification mostly cover basic actions and lack coverage of complex yoga postures, resulting in poor generalization to real-world practice [26]. Collecting large, diverse, labeled datasets for all postures is impractical, as many poses have few examples, and labeling fine-grained body configurations is both labor-intensive and time-consuming [28]. Besides, deep learning models typically require thousands of examples per class to achieve robust accuracy. In domains with limited samples, these models tend to overfit and fail to generalize to new poses or variations. This motivates the use of FSL techniques.

FSL is a technique designed to accurately recognize new classes given only a limited number of labeled samples [29]. Unlike supervised learning, which learns from a large dataset, FSL enables image classification even when fewer than 10 labeled examples are available for each class [30]. This approach focuses on learning the similarity between different classes. In FSL, there is a standard protocol written as N-way K-shot, where K-shot refers to the number of labeled examples in the support set per class, while N-way refers to the number of classes to identify [31]. In the broader domain of human analysis, FSL has been extensively applied to action and gesture recognition. For instance, skeleton-based FSL frameworks often utilize Graph Convolutional Networks or temporal alignment techniques (e.g., Dynamic Time Warping) to match dynamic motion sequences across support and query sets [32]. Similarly, meta-learning approaches such as Relation Networks have been adapted for fine-grained gesture recognition, successfully classifying hand movements from limited video samples by exploiting temporal dependencies [33].

However, these existing approaches are insufficient for yoga pose classification for two key reasons. First, most action recognition models rely heavily on spatiotemporal features extracted from video sequences, whereas yoga pose assessment often requires the analysis of static, peak-pose images where temporal cues are absent. Second, generic skeleton-based FSL methods typically

Figure 1
Estimated number of papers published from 2016 to 2024 with the keyword "yoga pose Classification"



focus on coarse movement patterns (e.g., "walking" vs. "running"). In contrast, yoga classification is a fine-grained structural problem, where subtle differences in joint angles (e.g., the alignment of the knee in Warrior I vs. Warrior II) determine the class validity. Standard FSL baselines frequently struggle to capture these minute structural nuances without domain-specific adaptation, highlighting a significant gap in the current literature.

Furthermore, FSL offers several specific frameworks that could be applied to pose classification. In metric-based FSL, matching networks perform classification by a learned weighted nearest neighbor over the support set [34]. Prototypical Networks compute class centroids in an embedding space and classify query points based on the nearest-prototype distance [35]. Both networks had been extended with episodic training to improve generalization. Pachetti and Colantonio showed that Prototypical Networks serve as strong baselines for medical image tasks [36]. In practice, these networks typically use Euclidean distance as the similarity metric and require a feature extractor trained on a related class. While most FSL research targets action and gesture recognition, the principled application of prototypical or matching networks to skeleton-based pose features remains an open avenue. Incorporating such networks with skeleton-derived embeddings can facilitate the easy clustering of structured coordinate data in the embedding space, thereby offering a promising direction for yoga pose classification. This study focuses strictly on static image classification to evaluate structural integrity without reliance on temporal transition data.

3. Research Methodology

This section outlines the methodology employed to design, implement, and evaluate the proposed ResProtoNet framework for yoga pose classification. The framework leverages a ResNet-18 backbone within a Prototypical Network architecture. While the combination of ResNet-18 and Prototypical Networks is established in general image classification, its application to fine-grained yoga pose assessment remains underexplored. The novelty of the ResProtoNet framework lies not in the backbone architecture itself but in the systematic integration of skeleton-based pose encoding as a prior for metric learning. By discarding irrelevant textural variation (e.g., clothing, background) via the skeletonization module, the framework forces the Prototypical Network to cluster samples based solely on geometric structural similarity. This creates a domain-adapted FSL pipeline that is robust to the high intra-class variability typical of yoga datasets.

To ensure reproducibility and consistency with the reported findings, the methodology follows a structured pipeline comprising seven key components: data preparation, data preprocessing, skeletonization, data merging and selection, data splitting, ResProtoNet model architecture, and experimental setup.

3.1. Data preparation

The dataset was obtained from the Kaggle platform and comprises 1,551 images (stored in .jpg, .png, and .jpeg formats) covering five common poses, including Downward Dog, Goddess, Tree, Plank, and Warrior II, as illustrated in Figure 2. This dataset consists of distinct, independent images collected from diverse sources (web-scraped), rather than correlated video frames extracted from continuous recording sessions. This ensures that the subjects, environments, and lighting conditions are highly variable between samples.

Since the images represent distinct instances rather than temporal sequences, a random train-test split effectively mitigates the risk of data leakage (e.g., the same subject appearing in identical conditions in both sets). Furthermore, the inclusion of the skeleton-based modality serves as a validation check (by removing background and appearance cues); thus, the model relies solely on geometric pose features for classification.

Figure 2
Kaggle open-source dataset of five yoga poses

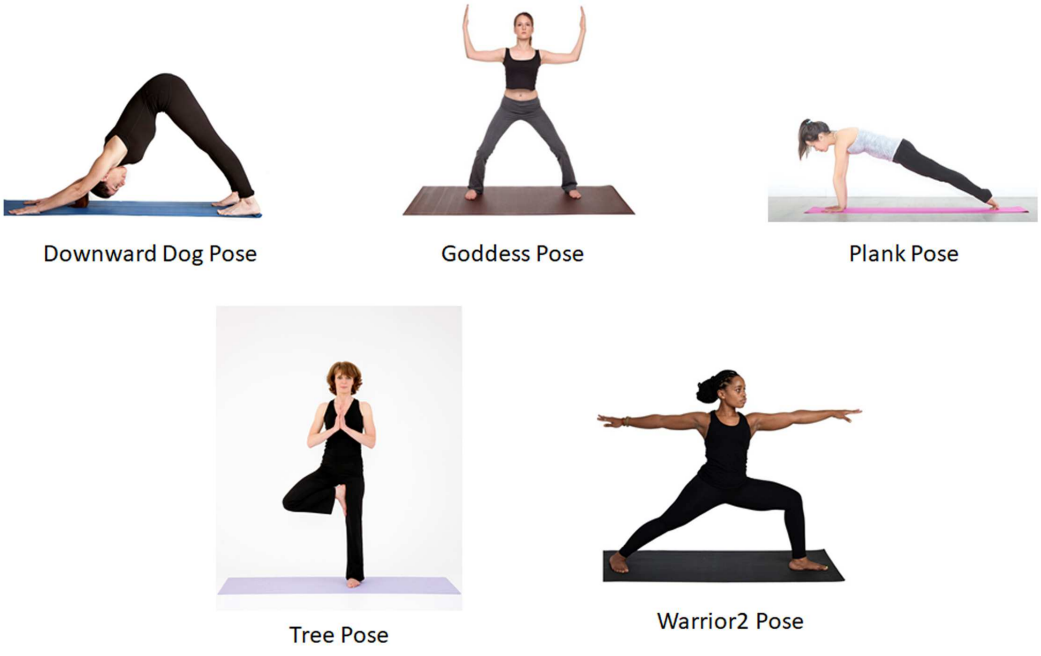


Table 2
Distribution of yoga poses in the dataset

Pose	Training samples	Testing samples	Total
Downward dog	223	97	320
Goddess	180	80	260
Tree	160	69	229
Plank	266	115	381
Warrior II	252	109	361
Total	1081	470	1551

The initial dataset consists of training data and testing data, with a 70:30 ratio, as shown in Table 2.

3.2. Data preprocessing

To ensure data quality and consistency, the preprocessing pipeline was divided into four sequential steps, including skeletonization, merging and selection, and dataset splitting, as shown in Figure 3.

3.3. Skeletonization

In the first stage, skeletonization was performed to generate a dataset that captured only human pose information. All raw images were processed using MediaPipe Pose, a lightweight HPE framework that implements the BlazePose model. BlazePose is capable of detecting 33 anatomical keypoints from a single RGB image, including major joints such as the shoulders, elbows, hips, knees, and ankles.

Figure 4 shows the raw and skeletonized images. For each image in the dataset, the detected landmarks were connected using predefined skeletal edges, and the resulting skeleton was rendered on a black background to form a new image. This process ensured that only body pose information was retained, while eliminating background, lighting, and clothing variations. Each skeletonized image inherited the class label of its corresponding raw image, thereby preserving dataset consistency. In subsequent experiments, the two datasets—raw images and skeletonized images—were used independently to train and evaluate the classification models.

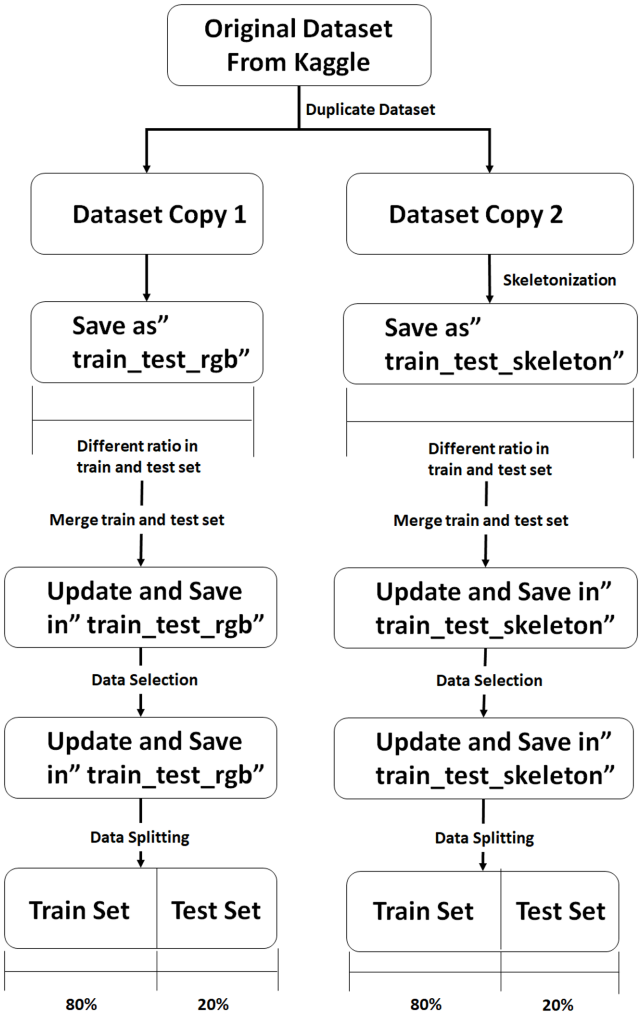
3.4. Data merging and selection

The original dataset obtained from Kaggle contained images distributed across separate training and testing folders. However, the division was inconsistent across classes; some poses had proportionally more training images, while others had more testing samples. To eliminate this imbalance, all images were first merged into a single dataset for each representation type (raw and skeletonized). This ensured uniformity and allowed the application of a consistent splitting strategy in later stages.

After the merge, a manual cleaning process was conducted to enhance data quality. Images containing multiple individuals were removed to avoid ambiguity in class labeling, as the system is designed to classify single-person poses. In addition, duplicate images were identified and discarded to prevent model overfitting caused by repeated samples. Another critical cleaning step involved discarding incomplete poses, where subjects were partially cropped, misaligned, or captured mid-transition between asanas, as these could introduce noise into the training process.

For the skeleton dataset, additional quality checks were necessary. Since skeletonization relies on accurate keypoint detection,

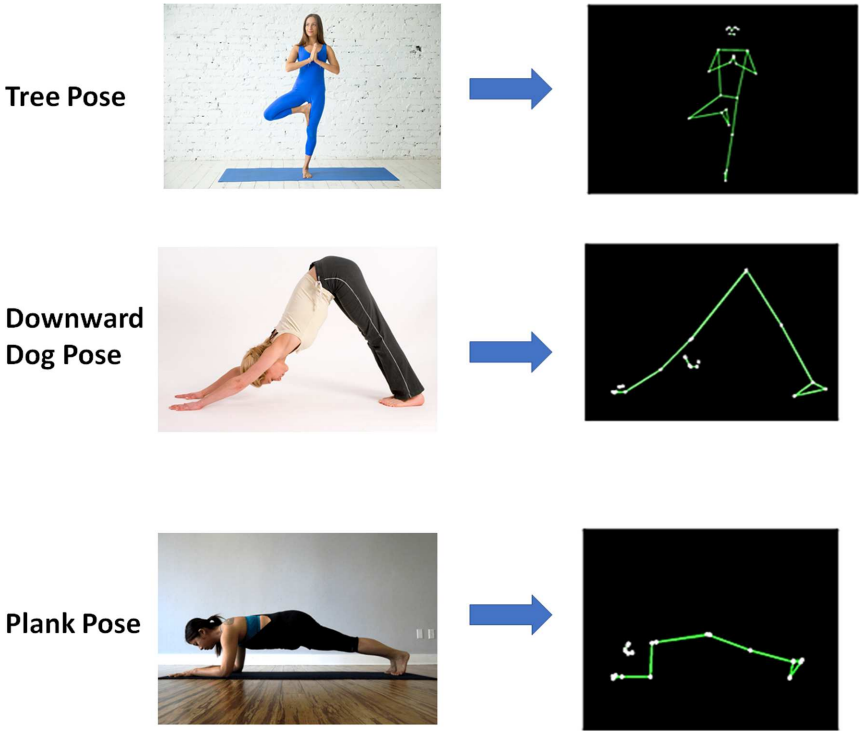
Figure 3
Overview of the data preprocessing pipeline



some images produced incomplete or distorted skeletons due to occlusion, unusual camera angles, or poor lighting. To ensure a fair comparison between input modalities, a synchronized cleaning strategy was employed; if a sample was identified as invalid in either domain (e.g., containing multiple individuals in RGB or failing keypoint extraction in skeleton), it was removed from both datasets.

This intersectional approach ensured that the final RGB and skeleton datasets were perfectly aligned, containing the exact same set of valid yoga instances. Through this merging and selection process, both the raw and skeletonized datasets were refined into

Figure 4
Examples of skeletonized yoga poses generated using MediaPipe Pose



high-quality collections of single-person, correctly labeled, and fully visible yoga poses, ensuring reliable inputs for subsequent training and evaluation phases. Figure 5 shows an example of valid and invalid sample data.

3.5. Data splitting

After merging and cleaning, the refined datasets were divided into training and testing subsets to facilitate model development and performance evaluation. An 80:20 split ratio was adopted, following common practice in machine learning research, where 80% of the data is allocated for training and the remaining 20% for testing. This ratio provides a sufficient number of samples to train the model effectively while preserving a representative portion of the data for unbiased evaluation.

The split was applied consistently across both the raw image dataset and the skeletonized dataset to ensure a fair comparison between the two input modalities. Within each class, images were randomly partitioned to preserve class balance across training and testing subsets. After splitting, the training set consisted of a total of 519 images, for the following classes, including 100 images for Downdog and Goddess each, 99 for Plank, 80 for Tree, and 140 for Warrior2. The test set, used for model evaluation, comprised 130 images with 25 images for each of the following classes, including Downdog, Goddess, and Plank each, 20 for Tree, and 35 for Warrior2.

3.6. ResProtoNet model architecture

The skeleton-ResProtoNet framework integrates a ResNet-18 backbone network with a Prototypical Network under FSL configuration. The overall architecture is illustrated in Figure 6. This design utilizes the feature extraction capability of ResNet-18 while leveraging the distance-based classification mechanism of Prototypical Networks to generalize effectively from limited training samples.

Figure 5
Examples of invalid data removed during cleaning. If a sample failed in either (a) RGB visual check or (b) skeleton extraction, excluded from both experimental datasets

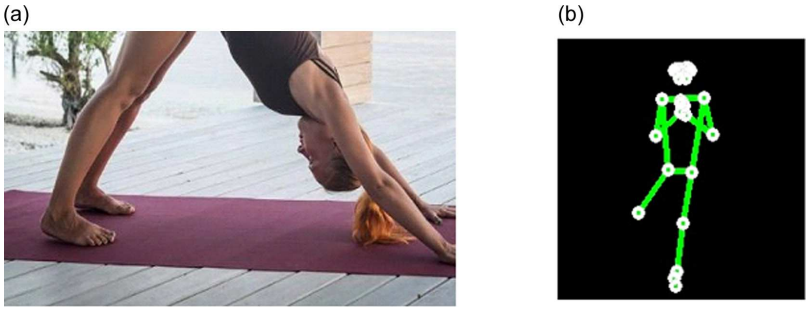
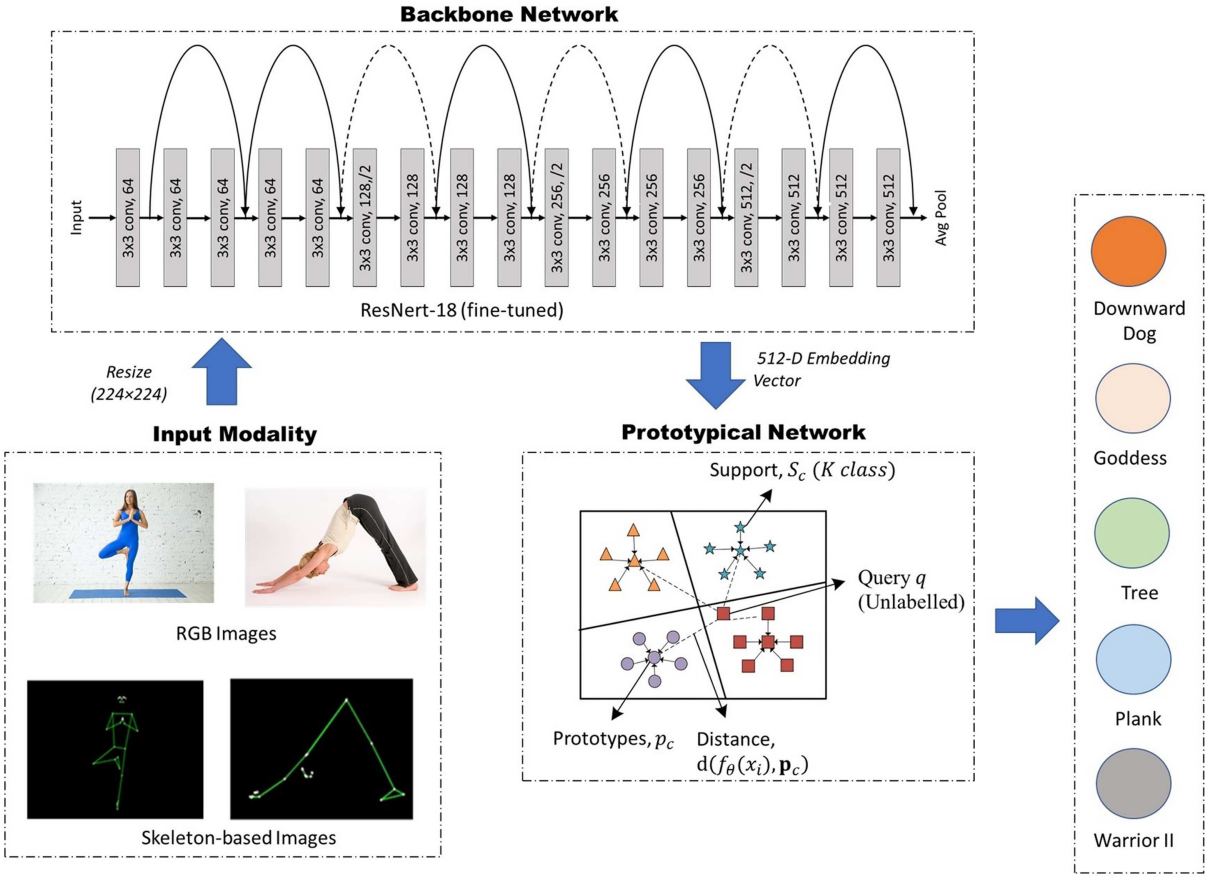


Figure 6
ResProtoNet model architecture



3.6.1. *Input modality*

Two input modalities are considered in the architecture. First is the raw RGB image, which contains full visual information, including background, texture, and color. Second is the skeleton-based images, which are generated using MediaPipe BlazePose. This tool detects 33 landmarks that connect into a skeletal structure and are drawn on a plain background to highlight body geometry. Both modalities are resized to 224 × 224 pixels before being fed into the backbone network.

3.6.2. *ResNet-18 backbone network*

ResNet-18, as shown in Table 3, was selected as the feature extractor due to its residual learning framework, which enables stable training of deeper networks by reducing the vanishing gradient problem [37, 38]. The model consists of 18 convolutional layers grouped into four residual blocks, with feature maps of increasing depth (64, 128, 256, and 512 filters). For this study, the pretrained ImageNet version of ResNet-18 was adopted, and the final fully connected classification layer was removed. Instead, a flattening layer was added to generate a 512-dimensional embedding vector for each input image, representing its high-level visual features in a compact mathematical space. This embedding serves as the input to the Prototypical Network.

3.6.3. *Prototypical Network*

The core of the FSL implemented in this study is the Prototypical Network. It operates on the principle that each class in an episode can be represented by a prototype p_c, defined as the mean centroid of the support set embeddings for that class. First, the ResNet19 acts as the feature extractor, f_θ, parameterized by θ. It maps each input image, x_i to a 512-dimensional feature vector $f_\theta(x_i) \in R^{512-D}$.

For a given class c, let the support set be denoted as $S_c = \{(x_i, y_i)\}$, where x_i is the support image and $y_i = c$ is the corresponding label. If the support set contains K labeled examples (K-shot), the p_c is computed as the mean vector of the embedded support points:

$$p_c = \frac{1}{|S_c|} \sum_{(x_i,\, y_i) \in S_c} f_\theta(x_i) \tag{1}$$

where S_c denotes the subset of support samples belonging to class c, S_c is the number of samples in the support set (equivalent to K in K-shot learning), and $f_\theta(x_i)$ is the embedding of the support image x_i.

3.6.4. *Classification*

Classification is performed using a nearest-neighbor approach in the embedding space. For a given unlabeled query sample, q, the distance to each class prototype is calculated using the squared Euclidean distance:

$$d\left(f_\theta(q), p_c\right) = \|f_\theta(q) - p_c\|_2^2 \tag{2}$$

Table 3
Architecture of ResNet-18

Stage	Layers	Output size	Notes
Input	RGB/skeleton-based image	224 × 224	After preprocessing
Conv1	7 × 7 Conv + MaxPool	112 × 112	Extracts low-level features
Residual block 1	2 basic blocks	56 × 56	Feature refinement
Residual block 2	2 basic blocks	28 × 28	Deeper feature maps
Residual block 3	2 basic blocks	14 × 14	Higher-level semantics
Residual block 4	2 basic blocks	7 × 7	Global structure
Average pooling	–	1 × 1	Embedding aggregation
Flatten	–	512-D	Final embedding vector

The model then assigns a probability distribution over classes based on a softmax over distances or simply assigns the discrete label, $\hat{y}$, corresponding to the nearest prototype:

$$\hat{y} = \arg \min_{c \in \{1,\ldots,N\}} \mathrm{d}\left(f_\theta(q), \mathrm{p_c}\right) \tag{3}$$

where q is the query image to be classified, $\mathrm{d}(\cdot)$ denotes the Euclidean distance function, N is the total number of classes in the episode (N-way), and $\hat{y}$ is the predicted class label. The q is assigned to the label of the nearest p_c in the embedding space, and the distance is computed using the squared Euclidean metric. This distance-based classification avoids the need for a fully connected layer and enables generalization to unseen classes with minimal labeled data.

3.6.5. *Parameter setup*

The implementation parameters for training and evaluation were defined through a structured configuration. The experiments followed a 5-way classification configuration, where episodes were sampled from either the original dataset or the skeleton-based dataset. The key configuration details are summarized in Table 4.

Table 4
Coding setup for few-shot learning experiments

Parameter	Value
Number of classes (N-way)	5
Support samples (N-shot)	1, 3, 5
Query samples (N-query)	5 per class
Number of tasks (N-tasks)	100
Training episodes	200
Validation tasks	600
Image size	224 × 224 pixels
Seeds (number of independent training runs on the dataset)	0, 1, 2

3.6.6. *Episodic training*

The proposed ResProtoNet model was trained using an episodic learning paradigm, which is a standard approach in FSL. In this paradigm, episodic learning simulated a small classification task per iteration with a support set and a query set. The support set contains a small number of labeled examples for each class and is used to compute class prototypes. The query set, on the other hand, includes unseen samples from the same classes and is used to evaluate classification performance within that episode.

All episodes were designed for 5-way classification, corresponding to the five yoga poses considered. Three configurations of support samples were explored, namely, 1-shot, 3-shot, and 5-shot, which provided one, three, or five labeled samples per class in the support set, respectively. For instance, in a 5-way 5-shot classification, each episode consists of five classes, with five support images and five query images per class.

A total of 200 training episodes were used to enhance the model's classification accuracy. This training budget was determined empirically, as preliminary experiments indicated that training loss stabilized and converged within this range. To mitigate overfitting and ensure generalization, a "model checkpointing" strategy was employed using the 600 validation tasks. Specifically, the model was evaluated on the validation set at regular intervals, and the model state yielding the highest validation accuracy was saved for final testing. This ensures that the selected model is not overfitted to the training support sets.

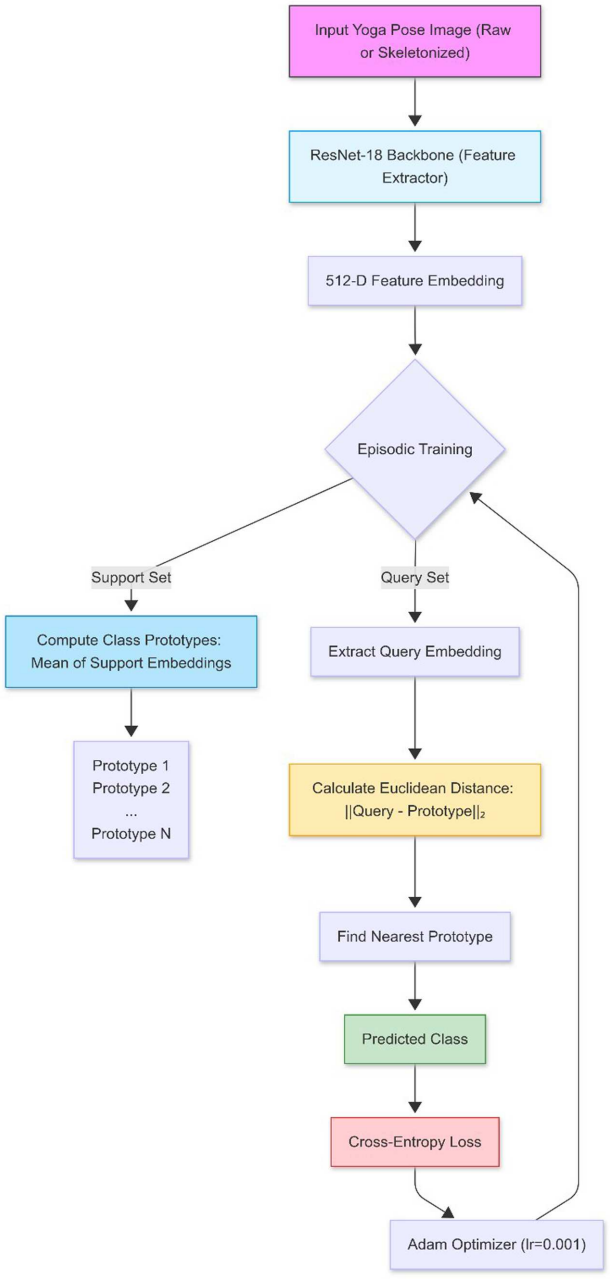
Regarding optimization, a constant learning rate of 0.001 was utilized with the Adam optimizer. No dynamic learning rate scheduler was required, as the ResNet-18 backbone was initialized with pretrained ImageNet weights, requiring only fine-tuning of the feature space rather than training from scratch. The TaskSampler library from EasyFSL was used to generate episodes by sampling classes and examples according to these configurations, ensuring consistency and reproducibility across experiments. Through repeated exposure to such episodes, the model progressively learned to construct robust prototypes and classify unseen queries, thereby strengthening its generalization capability in real-world few-shot scenarios.

3.6.7. *End-to-end process workflow*

The overall workflow of the proposed framework is illustrated in Figure 7. The process begins with the input yoga pose image, which may be either a raw image or its skeletonized representation. Subsequently, features are extracted using the ResNet-18 backbone, producing a 512-dimensional embedding vector for each sample. During episodic training, the dataset is divided into a support set and a query set.

For the support set, embeddings are averaged to form class prototypes, which serve as representative vectors for each yoga pose. For the query set, embeddings are computed and compared against the prototypes using the Euclidean distance metric. Each query is then assigned the label of the nearest prototype, yielding the predicted class. The model is optimized by minimizing the cross-entropy loss, with parameters updated using the

Figure 7
End-to-end workflow of the proposed framework



Adam optimizer (learning rate = 0.001). This workflow effectively integrates feature extraction, prototype computation, and distance-based classification within the FSL framework, enabling accurate recognition of yoga poses even under limited data settings.

3.7. Experimental setup

The experiments were conducted on a local workstation with the specifications summarized in Table 5. All experiments were carried out on a workstation equipped with a 5090 GPU (4 GB) and running Windows 11. The implementation was developed using Python, with deep learning models built in PyTorch, and the EasyFSL library was employed to facilitate FSL experiments. Each model configuration was trained over 200 episodes, ensuring consistency across the different setups.

Table 5
Experimental setup details

Component	Specification
Graphical Processing Unit	NVIDIA RTX 5090
Operating System	Windows 11
Python Version	3.9
Libraries	OpenCV, MediaPipe, PyTorch, EasyFSL, Matplotlib
FSL Libraries	EasyFSL (PyTorch-based)

The experimental design involved testing six distinct configurations, combining both raw image inputs and skeleton-based representations under varying shot conditions. Specifically, the evaluation was conducted using a 5-way classification protocol across three distinct shot settings: 1-shot, 3-shot, and 5-shot. This configuration was applied consistently to both the raw RGB dataset and the skeleton-based dataset to facilitate a direct modality comparison. To account for variability introduced by episodic sampling in FSL, results were reported as the mean accuracy across multiple runs. Evaluation metrics, such as precision, recall, and F1-score, are commonly used to evaluate model performance in this context.

4. Results and Discussion

The performance evaluation of yoga pose classification is presented in this section, focusing on two primary input modalities (raw RGB images and skeleton-based images). The experiment was conducted under the proposed ResProtoNet model under 1-shot, 3-shot, and 5-shot configurations. For comparison, several supervised baselines, including ResNet-18, DenseNet, MobileNetV3, and EfficientNet, were included. These have been widely applied in yoga pose classification studies, as mentioned in the related literature (Section 2.1). Table 6 summarizes the number of training and testing samples used in each configuration. This provides context for the model comparisons. To ensure the reliability of the outcomes, each experiment was conducted with three different random seeds, and the performance is reported in terms of classification performance, generalization, clustering indices, and robustness.

4.1. Input modality diagnostic analysis

Before evaluating classification performance, this subsection assesses the effectiveness of RGB and skeleton-based inputs across baseline backbones. The analysis focuses on three aspects: feature clustering quality, robustness under occlusion, and robustness under resolution degradation. To ensure statistical reliability, all robustness metrics are reported as the mean across three independent experimental runs (Seeds 0, 1, and 2), with variance indicated by standard deviations or confidence intervals.

4.1.1. Clustering indices

Tables 7 and 8 present the clustering indices for RGB and skeleton-based input baselines, measured using the Silhouette

Table 6
Dataset size per model

Model/input	Shot configuration	Training samples per class	Total training samples	Test samples per class	Total test samples
Skeleton few-shot	1-shot	1	5	20–35	130
Skeleton few-shot	3-shot	3	15	20–35	130
Skeleton few-shot	5-shot	5	25	20–35	130
Skeleton baseline	Full dataset	80–140	519	20–35	130
Raw image baseline	Full dataset	80–140	519	20–35	130

Table 7
Clustering indices of RGB input models

Baseline	Silhouette Score	Davies–Bouldin Index	Calinski–Harabasz Index
DensenNet	0.4981	0.9092	10138.4206
EfficientNet	0.6332	0.5965	19984.2637
MobileNetV3	0.6274	0.5749	22760.1986
ResNet18	0.6913	0.4413	36138.8489

Table 8
Clustering indices of skeleton-based input models

Baseline	Silhouette Score	Davies–Bouldin Index	Calinski–Harabasz Index
DensenNet	0.535	0.7411	14484.362
EfficientNet	0.6434	0.5389	25574.7354
MobileNetV3	0.6642	0.517	29616.9186
ResNet18	0.6836	0.4647	35513.5677

Score, Davies–Bouldin Index (DBI), and Calinski–Harabasz Index (CHI). These indices evaluate the quality of latent feature representation, where the Silhouette Score measures intra-class cohesion and inter-class separation, the DBI reflects cluster compactness relative to separation, and the CHI quantifies the ratio of between-class to within-class variance.

Based on Table 7, ResNet-18 produced the most structured feature space, achieving the highest Silhouette Score (0.6913), the lowest DBI (0.4413), and the highest CHI (36,138.85). EfficientNet and MobileNetV3 demonstrated reasonably well-separated clusters, although their scores were clearly lower than those of ResNet-18. DenseNet showed the weakest representation quality, with a low Silhouette Score of 0.4981 and a high DBI of 0.9092, indicating overlapping and less compact feature clusters.

Table 8 demonstrated that skeleton-based inputs generally improved clustering quality compared with RGB. Across most baselines, Silhouette Scores and CHI values increased, while DBI values decreased, confirming that skeleton representations yield more compact and separable clusters. ResNet-18 again achieved the strongest feature space (Silhouette Score: 0.6836, DBI: 0.4647, CHI: 35,513.57), maintaining its dominance across modalities.

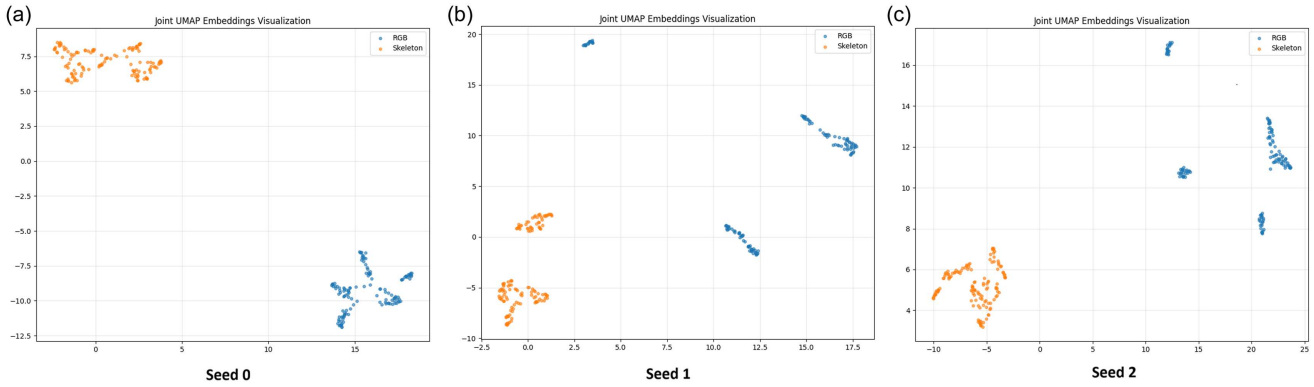
The cross-modality analysis in Table 9 reveals that skeleton-based input consistently improved clustering for DenseNet, EfficientNet, and MobileNetV3, with clear gains in Silhouette, CHI values, and reductions in DB. In contrast, ResNet-18 showed only marginal changes, with a slight reduction in Silhouette (−0.8%), an increase in DBI (+2.3%), and a decrease in CHI (−625). These marginal variations fall within expected experimental fluctuation, reflecting the strong clustering capability of ResNet-18 across both input modalities. To provide a qualitative perspective on the clustering metrics of the ResNet18 as reported in Tables 7 and 8, we visualized the feature embeddings of both modalities. Presents the Uniform Manifold Approximation and Projection (UMAP) embeddings for RGB and skeleton data across three random seeds (0, 1, and 2) to demonstrate the robustness of the feature distributions.

Table 9
Cross-modality differences in clustering indices (skeleton - RGB deltas, Δ)

Model	Δ Silhouette	Δ DBI	Δ CHI
DenseNet	+3.7%	–16.8%	+4346
EfficientNet	+1.0%	–5.8%	+5590
MobileNetV3	+3.7%	–5.8%	+6857
ResNet-18	–0.8%	+2.3%	–625

Note: Positive (+): improvement with skeleton-based input; negative (–): decrease.

Figure 8
Visual analysis of few-shot feature embeddings across multiple random initializations using UMAP. (a), (b), and (c) show joint embeddings for RGB (blue) and skeleton (orange) modalities under seeds 0, 1, and 2, respectively



Consistent with the quantitative metrics, the visualizations shown in Figure 8 reveal a clear structural distinction between the two modalities. The RGB embeddings (blue) tend to form tight, dense clusters, which correlates with the high Silhouette Scores observed in Table 7. In contrast, the skeleton embeddings (orange) exhibit elongated, manifold-like structures, reflecting continuous geometric variations in pose. Furthermore, the disjoint separation between the two color groups across all seeds confirms that the modalities occupy distinct regions of the feature space.

4.1.2. Robustness under occlusion

Table 10 presents the classification performance of RGB and skeleton inputs under varying occlusion levels, along with skeleton–RGB deltas. Occlusion robustness measures how well the model performs when portions of the input are masked. As illustrated in Figure 9, synthetic occlusion was generated by applying random rectangular masks covering 10%, 20%, 40%, and 60% of the image area. This simulates real-world scenarios where the subject may be partially hidden by furniture, equipment, or other practitioners.

At 0% occlusion, skeleton-based inputs provided modest performance gains for DenseNet (+0.74%) and MobileNetV3 (+0.66%), whereas ResNet-18 slightly favored RGB (−0.28%) and EfficientNet showed negligible change (0%). Under light occlusion (10–20%), RGB generally maintained superior performance across baselines. EfficientNet exhibited consistent drops when using skeleton input (−0.71%, −2.02%), while MobileNetV3 also declined at 20% occlusion (−2.56%). ResNet-18 demonstrated notable changes to skeleton input, showing a consistent drop of −1.85% and −2.82%. DenseNet exhibited a slight decrease at 10% occlusion (−0.56%) but recovered with a modest gain at 20% occlusion (+0.93%).

An interesting pattern emerged under heavy occlusion conditions (40–60%), where skeleton-based representations demonstrated superior robustness across most baselines. DenseNet achieved the largest gains (+20.04%, +11.70%), followed by EfficientNet (+7.57% and +2.03%). MobileNetV3 initially showed a drop at 40% occlusion (−3.34%) but reversed to demonstrate a strong skeleton advantage at 60% (+7.13%). ResNet-18, which remained nearly balanced at 40% (−0.42%), similarly shifted to favor skeleton at 60% occlusion (+3.69%).

These findings reveal a clear modality trade-off phenomenon. RGB input maintains an advantageous under light occlusion conditions, where texture and contextual background information remain accessible and informative. However, skeleton input demonstrates superior resilience under severe occlusion, particularly for DenseNet and EfficientNet, which achieved substantial relative performance improvements. Even MobileNetV3 and ResNet-18, which initially favored RGB, benefited from skeleton abstraction as occlusion levels increased. This suggested that pose-encoding representations from skeleton-based input provided improved reliability when visual information is severely compromised.

4.1.3. Robustness under resolution

Figure 10 [22, 23, 37, 39, 40] illustrates the classification accuracy of RGB and skeleton-based baselines under varying input resolutions. Resolution robustness evaluates model stability when image detail is reduced, simulating practical deployment scenarios on resource-constrained devices or with low-quality video streams.

At high resolutions (224 × 224), both modalities demonstrated robust performance with accuracies exceeding 90%. ResNet-18 achieved 92.70% with RGB inputs compared to 97.33% with skeleton-based inputs, while MobileNetV3 reached 96.73% and 97.41%, respectively. Similarly, DenseNet and EfficientNet performed comparably across both modalities, with skeleton-based inputs slightly outperforming RGB in both cases.

When resolution was reduced to 128 × 128, performance differences between modalities became notable. The result reveals distinct architectural preferences under these conditions. DenseNet and ResNet-18 maintained relatively high accuracy with RGB inputs (88.73% and 93.85%), demonstrating their capacity to extract meaningful representations from degraded visual information through deep hierarchical feature processing. In contrast, EfficientNet and MobileNetV3 achieved superior performance with skeleton inputs (91.66% and 90.53%), where structural pose abstraction effectively compensated for the loss of fine visual detail. Notably, ResNet-18 exhibited minimal performance differences between modalities (93.85% RGB vs. 92.70% skeleton), with a 1.15% margin difference suggesting that both input modalities remain feasible for this baseline.

4.1.4. Summary of input modality diagnostics

The diagnostic analyses confirmed that while different backbones exhibit varying sensitivities to modality and degradation,

Table 10
Classification performance under occlusion

Occlusion	DenseNet			EfficientNet			MobileNetV3			ResNet-18		
	RGB/ Skeleton	ci95	Delta	RGB/ Skeleton	ci95	Delta	RGB/ Skeleton	ci95	Delta	RGB/ Skeleton	ci95	Delta
0	0.9561/ 0.9635	±0.0248/ ±0.0315	0.0074	0.9653/ 0.9653	±0.0141/ ±0.0364	0	0.9658/ 0.9724	±0.0017/ ±0.0062	0.0066	0.9762/ 0.9734	±0.0036/ ±0.0092	–0.0028
10%	0.9534/ 0.9478	±0.0355/ ±0.0284	–0.0056	0.9656/ 0.9585	±0.0173/ ±0.0247	–0.0071	0.9665/ 0.9675	±0.0045/ ±0.0086	0.001	0.9837/ 0.9652	±0.0019/ ±0.0260	–0.0185
20%	0.9126/ 0.9219	±0.0170/ ±0.0204	0.0093	0.9618/ 0.9416	±0.0366/ ±0.0228	–0.0202	0.9295/ 0.9039	±0.0089/ ±0.0371	–0.0256	0.9701/ 0.9419	±0.0132/ ±0.0493	–0.0282
40%	0.5799/ 0.7803	±0.0553/ ±0.1062	0.2004	0.7775/ 0.8532	±0.1238/ ±0.0135	0.0757	0.7043/ 0.6709	±0.0838/ ±0.0597	–0.0334	0.7201/ 0.7159	±0.1272/ ±0.0945	–0.0042
60%	0.3239/ 0.4409	±0.0179/ ±0.0392	0.117	0.4517/ 0.4720	±0.0792/ ±0.0614	0.0203	0.3840/ 0.4553	±0.0631/ ±00.0201	0.0713	0.4102/ 0.4471	0.0058/ ±0.0503	0.0369

ResNet-18 consistently provided well-structured embeddings, maintained robustness under occlusion, and preserved competitive accuracy across resolutions. These characteristics highlight its suitability as a reliable feature extractor, supporting its use as the backbone in the proposed ResProtoNet framework.

4.2. Classification performance metrics

Following the diagnostic analyses, this section evaluates the classification performance of the proposed ResProtoNet against baselines using both RGB and skeleton-based input modalities. The proposed model was evaluated under 1-shot, 3-shot, and 5-shot learning configurations to assess its FSL capabilities. To ensure experimental consistency, all models were trained and evaluated using identical protocols with three independent random seeds (0, 1, and 2). Performance metrics were measured in terms of mean accuracy (±95% CI), balanced accuracy, precision, recall, and F1-score, with the results summarized in Table 11 for RGB inputs and Table 12 for skeleton-based inputs, respectively.

Based on the result in Table 11, ResNet-18 achieved the strongest supervised baseline accuracy of 97.62% (±0.36%), outperforming DenseNet (95.61%) and slightly surpassing MobileNetV3 (96.58%) and EfficientNet (96.53%). Despite the strong baseline, the introduction of FSL further improved classification accuracy. The proposed ResProtoNet surpassed the ResNet-18 baseline, achieving 97.69% (±0.42%) in the 1-shot configuration, which highlights its ability to adapt to novel tasks with minimal support examples. Accuracy increased further with additional support samples, reaching 98.22% in the 3-shot and 98.17% in the 5-shot configurations. Interestingly, the 3-shot configuration produced slightly higher mean accuracy than the 5-shot configuration for both RGB and skeleton inputs. This marginal difference (0.05%) lies within the confidence interval and is therefore not statistically significant. Such variations are common in FSL, where performance depends on support set quality rather than size alone.

For Table 12 results, overall accuracy was comparable to or higher than RGB across models, particularly for backbones such as DenseNet, EfficientNet, and MobileNetV3, which is consistent with the clustering analysis and supports the effectiveness of pose abstraction. Among supervised baselines, ResNet-18 achieved the highest accuracy (97.34% ±0.92%), followed closely by MobileNetV3 (97.24%). Nevertheless, ResProtoNet delivered the best results, with accuracy reaching 98.11% in the 3-shot and 98.13% in the 5-shot. Notably, in the 1-shot configuration, ResProtoNet achieved performance comparable to the ResNet-18 baseline, differing by a small margin of 0.01%. While the improvements are marginal and within the confidence intervals, they demonstrate that the few-shot framework can match or exceed ResNet-18 even when provided with only a single labeled sample per class.

Across both modalities, precision, recall, and F1-score closely mirrored mean accuracy, indicating balanced classification without systematic bias toward specific yoga classes. The findings reveal that ResProtoNet not only matches but also consistently exceeds all supervised baseline performance, particularly in limited data, where FSL approaches demonstrate their practical advantages. Moreover, the accuracy differences between the 3- and 5-shot configurations were minimal. For RGB inputs, performance peaked at 98.22% in the 3-shot configuration and declined slightly to 98.17% at 5-shot, with a gap of 0.05%. For skeleton-based input, accuracy values were nearly identical at

Figure 9
Visualization of synthetic occlusion levels applied during robustness testing. (a) Original image (0%), (b) 10% occlusion, (c) 40% occlusion, and (d) 60% occlusion. Random rectangular masks were applied to simulate obstruction

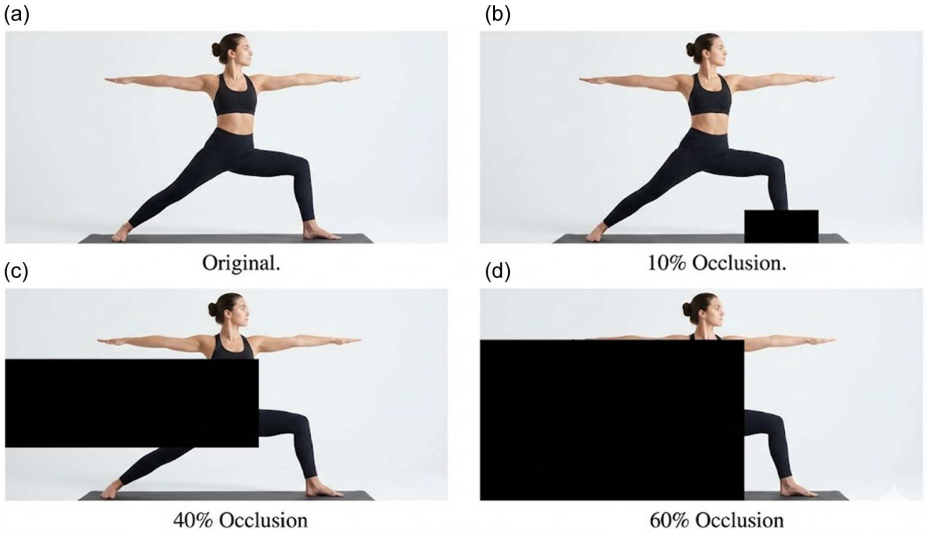


Figure 10
Classification accuracy of RGB and skeleton-based input baselines under varying input resolution

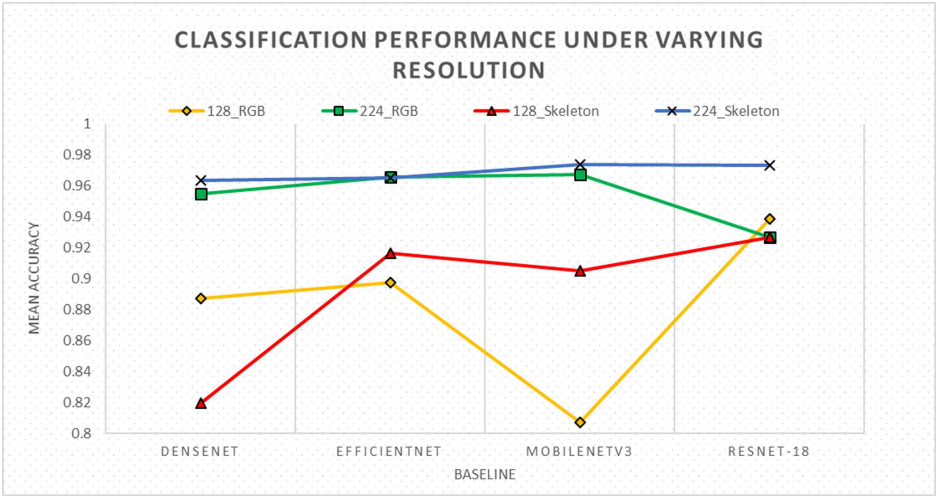


Table 11
Classification performance of RGB input models

Models	Mean accuracy ± 95% CI	Balanced accuracy	Precision (macro)	Recall (macro)	F1-score (macro)
DensenNet	0.9561 ± 0.0248	0.9561	0.9563	0.9561	0.9556
EfficientNet	0.9653 ± 0.0141	0.9653	0.9660	0.9653	0.9649
MobileNetV3	0.9658 ± 0.0017	0.9658	0.9660	0.9658	0.9657
ResNet18	0.9762 ± 0.0036	0.9762	0.9766	0.9762	0.9762
ResProtoNet 1-shot	0.9769 ± 0.0042	0.9769	0.9777	0.9769	0.9770
ResProtoNet 3-shot	0.9822 ± 0.0083	0.9822	0.9830	0.9822	0.9823
ResProtoNet 5-shot	0.9817 ± 0.011	0.9817	0.9825	0.9817	0.9818

Note: CI = Confidence Interval.

Table 12
Classification performance of skeleton-based input models

Models	Mean accuracy ± 95% CI	Balanced accuracy	Precision (macro)	Recall (macro)	F1-score (macro)
DensenNet	0.9635 ± 0.0315	0.9635	0.9637	0.9635	0.9636
EfficientNet	0.9653 ± 0.0364	0.9653	0.9657	0.9653	0.9652
MobileNetV3	0.9724 ± 0.0062	0.9724	0.9724	0.9724	0.9722
ResNet18	0.9734 ± 0.0092	0.9734	0.9734	0.9734	0.9733
ResProtoNet 1-shot	0.9733 ± 0.0094	0.9733	0.9734	0.9733	0.9732
ResProtoNet 3-shot	0.9811 ± 0.0043	0.9811	0.9812	0.9811	0.9810
ResProtoNet 5-shot	0.9813 ± 0.0028	0.9813	0.9816	0.9813	0.9813

Note: CI = Confidence Interval.

98.11% and 98.13%, with a gap of 0.02%. This indicates that ResProtoNet achieves near-optimal performance with the 3-shot configuration, and additional labeled samples beyond this point provide only marginal benefits.

It is worth addressing the consistently high accuracy (>98%) observed across the experiments. This performance can be attributed to the distinct geometric nature of the five selected classes (Downward Dog, Goddess, Tree, Plank, and Warrior II), which exhibit significant inter-class structural differences that are easily separable in the embedding space. Furthermore, the comparable performance between RGB and skeleton-based modalities serves as a critical validation check. Since the skeleton modality removes all background texture and subject appearance information, the high accuracy achieved in this setting confirms that the model is learning genuine pose geometry rather than overfitting to environmental artifacts or subject identity (data leakage).

4.3. Computational efficiency analysis

To assess the feasibility of deploying ResProtoNet in real-world applications, the computational cost of the ResNet-18 backbone was evaluated. Table 13 summarizes the model parameters, memory footprint, and average inference latency measured on an NVIDIA RTX 5090.

Table 13
Computational performance metrics of the proposed framework

Metric	Value
Backbone architecture	ResNet-18 (feature extractor)
Input resolution	224×224 pixels
Total parameters	11,176,512 (approx. 11.2 M)
Model size (weights)	44.71 MB
Computational cost (FLOPs)	1.82 GFLOPs
Avg. inference time (GPU)	<2.00 ms

The ResNet-18 backbone requires approximately 11.2 million parameters and 1.82 GFLOPs of computation, making it a lightweight architecture suitable for deployment on edge devices such as NVIDIA Jetson processors or modern smartphones. However, overall system latency in practical deployment is highly dependent on the chosen input modality. The RGB modality requires only a single forward propagation, yielding the lowest latency. The skeleton modality, on the other hand, introduces preprocessing overhead (approximately 30–50 milliseconds on CPU) due to the need for MediaPipe inference.

4.4. Comparison with state of the art

To validate the effectiveness of the proposed framework, Table 14 compares ResProtoNet with recent SOTA studies focusing on yoga pose classification using yoga pose-related datasets.

As observed, standard supervised approaches require large-scale annotated datasets to achieve high performance. Rajendran et al. [27] achieved the highest accuracy of 99.07% using a deep DenseNet201 architecture trained on the full dataset. However, the proposed ResProtoNet achieves a comparable accuracy of 98.22% (a difference of only 0.85%) using the 3-shot configuration, which utilizes significantly fewer labeled samples. This demonstrates that the proposed few-shot framework offers a superior trade-off between accuracy and data efficiency, making it highly suitable for applications where data annotation is costly.

4.5. Scalability and generalization analysis

To evaluate the generalization capability of the framework beyond the initial five classes, an additional stress test was conducted using the Yoga-82 dataset, a large-scale benchmark containing 82 fine-grained classes, obtained in the Kaggle platform. Unlike the primary dataset, Yoga-82 features high inter-class similarity, presenting a significantly more complex classification challenge.

Table 15 presents the results under a 5-shot configuration. While the EfficientNet backbone achieved the highest empirical accuracy (58.44%), demonstrating the framework's capacity to scale with more complex feature extractors, our proposed ResProtoNet (utilizing ResNet-18) maintained competitive performance (51.04%) while adhering to a standard lightweight architecture. Critically, the experiments reveal the robustness of the skeleton modality regardless of the backbone used. For instance, the skeleton-based models consistently outperformed their RGB counterparts, with lightweight encoders like MobileNetV3 recovering around 15% accuracy when switching to skeleton inputs. While absolute accuracy is lower than the 5-pose task due to the fine-grained nature of Yoga-82, these results confirm that the framework and particularly the skeleton modality scales effectively to handle complex, real-world pose variations.

Table 14
Comparison of ResProtoNet with existing yoga pose classification benchmarks

Refs.	Method/model	Learning paradigm	Accuracy (%)
Aruna et al. [23]	ResNet-18 (transfer learning)	Supervised (full data)	94.93
Kashyap et al. [24]	EfficientNet-B7	Supervised (full data)	97.61
Tayal et al. [25]	EfficientNet + AdaDelta	Supervised (full data)	97.45
Rajendran et al. [27]	DenseNet201	Supervised (full data)	99.07
Proposed ResProtoNet	ResNet-18 + ProtoNet	Few-shot (3-shot)	98.22

Table 15
Classification performance on the Yoga-82 dataset

Backbone model	Input modality	Mean accuracy (%)	Std dev (±)
EfficientNet	RGB	58.44	2.42
	Skeleton	54.3	16.89
DenseNet	RGB	50.96	5.92
	Skeleton	57.7	2.6
MobileNetV3	RGB	39.76	4.03
	Skeleton	54.63	9.76
ResNet-18	RGB	51.04	0.62
	Skeleton	51.01	4.42

5. Conclusion

The work investigates the integration of FSL for yoga pose classification and assesses its effectiveness relative to supervised baselines. Through the development of the proposed ResProtoNet model, which integrates a ResNet-18 backbone with a Prototypical Network, the study successfully addressed the challenge of limited annotated training data, while maintaining strong generalization across yoga postures. The results demonstrate that the proposed model consistently outperforms all baselines across 1-shot, 3-shot, and 5-shot configurations, achieving notable accuracy improvements with only minimal annotated data. These findings validate the applicability of meta-learning strategies in domains where annotated samples are limited, highlighting their capacity to achieve competitive performance without the need for extensive training.

The comprehensive diagnostic analyses provided important insights into the characteristics of input modality and their interaction with various baselines. Clustering analysis revealed that skeleton-based inputs improve feature space separability for lightweight baselines, including DenseNet, EfficientNet, and MobileNetV3, whereas ResNet-18 maintains robust clustering quality across both modalities with only marginal variation. Robustness analysis further highlighted modality-specific advantages, where under light occlusion (10–20%), RGB input generally outperformed skeleton by leveraging residual texture and background cues. In contrast, under severe occlusion (40–60%), skeleton-based inputs proved superior, mitigating the loss of visual information through pose abstraction. Similarly, in the resolution analysis, DenseNet and ResNet-18 retained higher accuracy with RGB at 128 × 128, reflecting their ability to exploit degraded texture, while EfficientNet and MobileNetV3 benefited more from skeleton abstractions. Overall, skeleton inputs exhibited smaller average accuracy drops compared to baselines, confirming their stronger resilience to both occlusion and resolution degradation. ResNet-18 remained competitive across both modalities, justifying its role as the backbone for the proposed ResProtoNet.

Despite these promising results, several limitations must be acknowledged. First, the framework relies on a pretrained ResNet-18 backbone, meaning its performance is contingent on the availability of high-quality, transferable features from large-scale datasets like ImageNet. Second, the study was limited to five fundamental yoga poses from a single open-source dataset; while sufficient for proof-of-concept, this narrow scope does not capture the full diversity of advanced asanas or the variability of real-world clinical environments. Third, the current approach focuses exclusively on static peak-pose classification. It therefore lacks the ability to analyze the temporal dynamics of pose transitions, which are critical for assessing fluidity and preventing injury during movement.

In conclusion, the research objectives were fully achieved. The study confirms the effectiveness of ResProtoNet in limited data yoga pose classification, demonstrates the advantages of FSL over traditional supervised learning, and provides new insights into the modality trade-offs between RGB and skeleton-based inputs. The outcomes highlight FSL as a practical and scalable approach for healthcare and exercise monitoring systems, where robustness to occlusion, resolution changes, and constrained annotation resources is essential for real-world deployment.

6. Future Works

While the ResProtoNet framework demonstrates promising results for few-shot yoga pose classification, several directions remain open for future exploration. First, dataset expansion and validation are essential. The current evaluation was limited to five fundamental yoga poses, which capture only a small portion of yoga poses. Future research should extend the classification task to encompass more challenging poses, including advanced poses that require greater precision in joint positioning and alignment. Moreover, the proposed framework requires validation across multiple datasets to establish generalization capabilities beyond the present study.

Second, hybrid multimodal architectures offer significant opportunities. The comparative analysis of RGB and skeleton

modalities highlights their complementary strengths: RGB provides rich textural context, while skeletons offer structural precision. Future work should investigate dual-stream fusion networks that dynamically weight these inputs using attention mechanisms to maximize robustness under varying environmental conditions.

Third, moving beyond static analysis, future studies should explore Temporal FSL. Since yoga involves dynamic transitions between poses, static analysis cannot capture movement fluidity or breathing patterns. Integrating spatiotemporal modules, such as Long Short-Term Memory networks or Transformers, within the few-shot framework would enable the system to assess the quality of pose transitions and provide comprehensive, real-time feedback to practitioners.

Funding Support

This work is sponsored by Xiamen University Malaysia Research Fund (XMUMRF) Cycle 14/2024 with the grant number XMUMRF/2024-C14/IECE/0052.

Ethical Statement

The authors declare that this study did not require formal ethical approval because Xiamen University Malaysia does not require Institutional Review Board or ethics committee approval for secondary computational research utilizing pre-existing, publicly available datasets. The datasets analyzed in this study, which include facial images, were obtained from open-access repositories. This exemption is based on an Official Document issued by XMUM.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in Yoga Poses Dataset [Kaggle] at https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset, in EasyFSL (PyTorch library for few-shot learning) [GitHub] at https://github.com/sicara/easy-few-shot-learning, in Yoga-82 Dataset [Kaggle] at https://www.kaggle.com/datasets/akashrayhan/yoga-82, and in Blazepose_skeletons_Yoga_82 Dataset [Kaggle] at https://www.kaggle.com/datasets/rashiniyasp/blazepose-skeletons-yoga-82.

Author Contribution Statement

Chean Khim Toa: Conceptualization, Methodology, Validation, Resources, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition. **Kai Liang Lew:** Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – review & editing, Supervision, Project administration. **Sin Pei Ton:** Conceptualization, Formal analysis, Resources.

References

[1] He, M. H., & Toa, C. K. (2024). A review of sensor-based gait recognition. In *2024 International Conference on Computing Innovation, Intelligence, Technologies and Education*, 1–6. https://doi.org/10.1109/CIITE62244.2024.10987605

[2] Ang, Y. H., Chan, C. K., Yap, S. C., Toa, C. K., Tran, P., & Goh, S. K. (2022). Markerless human motion analysis for telerehabilitation: A case study on squat. In *International Conference on Future Smart Cities*, 249–259. https://doi.org/10.1007/978-3-031-52303-8_18

[3] Syed, S. A., Akram, M., Rashid, A., Khalil, M. T., Anwar, H., Laila, U., . . . , & Mohiuddin, G. (2022). A brief review of beneficial effects of yoga on physical and mental health: Yoga on physical & mental health. *Medical and Health Science Journal*, *6*(2), 30–34. https://doi.org/10.33086/mhsj.v6i02.3212

[4] Sinha, P., Shetty, T., Pandey, A., Rahman, S. A., & Sen, A. (2023). Detection and correction of yoga poses. *International Journal of Future Generation Communication and Networking*, *5*(5), 1–7. https://doi.org/10.36948/ijfmr.2023.v05i05.7523

[5] Gadhvi, R., Desai, P., & Siddharth. (2025). Posepilot: An edge-AI solution for posture correction in physical exercises. In N. Gonçalves, H. P. Oliveira, & J. A. Sánchez (Eds.), *Iberian conference on pattern recognition and image analysis*, (pp. 208–219). Springer. https://doi.org/10.1007/978-3-031-99568-2_17

[6] Samkari, E., Arif, M., Alghamdi, M., & Al Ghamdi, M. A. (2023). Human pose estimation using deep learning: A systematic literature review. *Machine Learning and Knowledge Extraction*, *5*(4), 1612–1659. https://doi.org/10.3390/make5040081

[7] Aouaidjia, K., Zhang, C., & Pitas, I. (2025). Spatio-temporal invariant descriptors for skeleton-based human action recognition. *Information Sciences*, *700*, 121832. https://doi.org/10.1016/j.ins.2024.121832

[8] Liu, Y., Zhang, H., Li, Y., He, K., & Xu, D. (2023). Skeleton-based human action recognition via large-kernel attention graph convolutional network. *IEEE Transactions on Visualization and Computer Graphics*, *29*(5), 2575–2585. https://doi.org/10.1109/TVCG.2023.3247075

[9] Ferraris, C., Amprimo, G., Cerfoglio, S., Vismara, L., & Cimolin, V. (2025). A deep dive into MediaPipe pose for postural assessment: A comparative investigation. *IEEE Access*, *13*, 211055–211074. https://doi.org/10.1109/ACCESS.2025.3643126

[10] Vangalapat, T., & Gopireddy, R. R. (2022). Deep learning with limited data: A comprehensive survey of few-shot and zero-shot learning paradigms. *Journal of Scientific and Engineering Research*, *9*(3), 328–351. https://doi.org/10.5281/zenodo.17384469

[11] Zeng, W., & Xiao, Z. Y. (2024). Few-shot learning based on deep learning: A survey. *Mathematical Biosciences and Engineering*, *21*(1), 679–711. https://doi.org/10.3934/mbe.2024029

[12] Song, Y., Wang, T., Cai, P., Mondal, S. K., & Sahoo, J. P. (2023). A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, *55*(13s), 1–40. https://doi.org/10.1145/3582688

[13] Kappan, M. M., Sandoval, E. B., Meijering, E., & Cruz, F. (2026). A survey on deep learning for 2D and 3D human pose estimation. *Artificial Intelligence Review*, *59*(1), 32. https://doi.org/10.1007/s10462-025-11430-4

[14] Lan, G., Wu, Y., Hu, F., & Hao, Q. (2022). Vision-based human pose estimation via deep learning: A survey. *IEEE*

Transactions on Human-Machine Systems, *53*(1), 253–268. https://doi.org/10.1109/THMS.2022.3219242

[15] Jiang, X., Hu, Z., Wang, S., & Zhang, Y. (2023). A survey on artificial intelligence in posture recognition. *Computer Modeling in Engineering & Sciences: CMES*, *137*(1), 35. https://doi.org/10.32604/cmes.2023.027676

[16] Chung, J. L., Ong, L. Y., & Leow, M. C. (2022). Comparative analysis of skeleton-based human pose estimation. *Future Internet*, *14*(12), 380. https://doi.org/10.3390/fi14120380

[17] Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7291–7299. https://doi.org/10.1109/TPAMI.2019.2929257

[18] Jo, B., & Kim, S. (2022). Comparative analysis of OpenPose, PoseNet, and MoveNet models for pose estimation in mobile devices. *Traitement du Signal*, *39*(1), 119. https://doi.org/10.18280/ts.390111

[19] Desai, M., & Mewada, H. (2023). A novel approach for yoga pose estimation based on in-depth analysis of human body joint detection accuracy. *PeerJ Computer Science*, *9*, e1152. https://doi.org/10.7717/peerj-cs.1152

[20] Liu, Y. (2024). Deep learning approaches on computer vision. *Applied and Computational Engineering*, *92*(1), 59–67. https://doi.org/10.54254/2755-2721/92/20241682

[21] Garg, S., Saxena, A., & Gupta, R. (2023). Yoga pose classification: A CNN and MediaPipe inspired deep learning approach for real-world application. *Journal of Ambient Intelligence and Humanized Computing*, *14*(12), 16551–16562. https://doi.org/10.1007/s12652-022-03910-0

[22] Shourie, P., Anand, V., Upadhyay, D., Devliyal, S., & Gupta, S. (2024). Yogaposevision: Mobilenetv3-powered CNN for yoga pose identification. In *2024 7th International Conference on Circuit Power and Computing Technologies*, *1*, 659–663. https://doi.org/10.1109/ICCPCT61902.2024.10673072

[23] Aruna, M., Kaneriya, G., & Jain, P. (2024). Leveraging pretrained ResNet-18 with transfer learning for yoga posture classification. In *2024 Second International Conference on Networks, Multimedia and Information Technology*, 1–5. https://doi.org/10.1109/NMITCON62075.2024.10699235

[24] Kashyap, S., Gupta, A., Ansari, M. A., & Singh, D. K. (2023). Review of an evolved DNN architecture EfficientNet for yoga pose detection problem. In *2023 IEEE 11th Region 10 Humanitarian Technology Conference*, 829–834. https://doi.org/10.1109/R10-HTC57504.2023.10461771

[25] Tayal, D. K., Nagpal, A., Jain, A., Dattatrey, H., Arya, R., & Gaur, H. (2025). Comprehensive analysis of deep learning approaches for yoga pose detection. In *2025 International Conference on Innovation in Computing and Engineering*, 1–6. https://doi.org/10.1109/ICE63309.2025.10984340

[26] Verma, M., Kumawat, S., Nakashima, Y., & Raman, S. (2020). Yoga-82: A new dataset for fine-grained classification of human poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 1038–1039. https://doi.org/10.1109/CVPRW50498.2020.00527

[27] Rajendran, A. K., & Sethuraman, S. C. (2024). Denseposecompare: A comparative study of DenseNet models in yoga pose classification. In *2024 International Conference on Cognitive Robotics and Intelligent Systems*, 471–480. https://doi.org/10.1109/ICC-ROBINS60238.2024.10534022

[28] Wu, Y., Lin, Q., Yang, M., Liu, J., Tian, J., Kapil, D., & Vanderbloemen, L. (2021). A computer vision-based yoga pose grading approach using contrastive skeleton feature representations. *Healthcare*, *10*(1), 36. https://doi.org/10.3390/healthcare10010036

[29] Liu, Y., Zhang, H., Zhang, W., Lu, G., Tian, Q., & Ling, N. (2022). Few-shot image classification: Current status and research trends. *Electronics*, *11*(11), 1752. https://doi.org/10.3390/electronics11111752

[30] Uzhinskiy, A. (2025). Evaluation of different few-shot learning methods in the plant disease classification domain. *Biology*, *14*(1), 99. https://doi.org/10.3390/biology14010099

[31] Thomas, H., Gravier, G., & Sébillot, P. (2024). One-shot relation retrieval in news archives: Adapting N-way K-shot relation classification for efficient knowledge extraction. *Procedia Computer Science*, *246*, 1060–1069. https://doi.org/10.1016/j.procs.2024.09.525

[32] Liu, X., Zhou, S., Wang, L., & Hua, G. (2023). Parallel attention interaction network for few-shot skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1379–1388. https://doi.org/10.1109/ICCV51070.2023.00133

[33] Hong, J., Fisher, M., Gharbi, M., & Fatahalian, K. (2021). Video pose distillation for few-shot, fine-grained sports action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9254–9263. https://doi.org/10.1109/ICCV48922.2021.00912

[34] Doreen, A., Mwangi, W., & Muriithi, P. (2025). Optimization of matching networks with transfer learning in few-shot pneumonia detection. *Discover Applied Sciences*, *7*(9), 1024. https://doi.org/10.1007/s42452-025-07376-5

[35] Chowdhury, R. R., & Bathula, D. R. (2022). Influential prototypical networks for few shot learning: A dermatological case study. In *2022 IEEE 19th International Symposium on Biomedical Imaging*, 1–4. https://doi.org/10.1109/ISBI52829.2022.9761403

[36] Pachetti, E., & Colantonio, S. (2024). A systematic review of few-shot learning in medical imaging. *Artificial Intelligence in Medicine*, *156*, 102949. https://doi.org/10.1016/j.artmed.2024.102949

[37] Liu, H., Brailsford, T., & Bull, L. (2024). Resnet18 performance: Impact of network depth and image resolution on image classification. In *Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence*, 351–356. https://doi.org/10.1145/3704137.3704173

[38] Toa, C. K., Sim, K. S., & Tan, S. C. (2021). Electroencephalogram-based attention level classification using convolution attention memory neural network. *IEEE Access*, *9*, 58870–58881. https://doi.org/10.1109/ACCESS.2021.3072731

[39] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708. https://doi.org/10.1109/CVPR.2017.243

[40] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, *97*, 6105–6114.

How to Cite: Toa, C. K., Lew, K. L., & Ton, S. P. (2026). ResProtoNet: A Skeleton-Aware Few-Shot Framework for Yoga Pose Classification. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62027514