# MYPO-Net: A Robust Deep Learning Approach for Multi-Yoga Pose Detection and Occlusion Handling

Rehana Danial[1], Nosheen Qamar[2]*, Nosheen Sabahat[3], Faria Nazir[4], Ali Salem Bin Sama[5], Lamia Hassan Rahamatalla[6], Osman Elwasila[7], Abdulrahman Alojail[8], Marwan Abu-Zanona[9]
Department of Software Engineering, University of Management & Technology, Lahore, Pakistan[1, 2, 4]
Department of Computer Science, Forman Christian College University, Lahore, Pakistan[3]
Department of Management Information Systems-College of Business Administration,
King Faisal University, Al-Ahsa 31982, Saudi Arabia[5, 6, 7, 8, 9]

***Abstract*—Yoga has become a well-known holistic process worldwide and has been appreciated due to its physical, psychological, and injury-preventive effects. The swift development of online fitness applications has created a growing need for automated systems that can precisely identify and analyze yoga poses. The current methods, however, are limited in terms of the limited diversity of datasets, insufficient occlusion, low performance in multi-person settings, and a lack of feedback mechanisms to offer corrective feedback. In order to overcome these shortcomings, this study introduces MYPO-Net, an artificial intelligence (AI) based deep learning model that uses the efficiency of MobileNet and the classification performance of EfficientNetB0. The model is trained and tested on the Yoga82 data, with a detailed preprocessing pipeline, such as resizing, normalization, and data augmentation, to improve resilience to real-world variations. Experimental evidence shows a classification accuracy of 97.65, which is higher than a variety of baseline architectures (VGG16: 87%, InceptionV3: 82%, ResNet50: 58) and has high computational efficiency. Confusion matrix analysis shows that there is valid detection in 16 yoga position classes. The persisting issues in real-time implementation and poor image quality settings are distinguished as future work directions. MYPO-Net is a highly scalable, affordable, and open-source platform to support digital yoga teaching, fitness apps, and rehabilitative health care.**

***Keywords*—Yoga pose classification; deep learning; mobilenet; efficientnetb0; occlusion handling; transfer learning; convolutional neural networks; human pose estimation***

## I. Introduction

Yoga is a thousand-year-old practice that began in ancient India and combines physical poses (asanas), breathing exercises (pranayama), and meditation to promote holistic health. Yoga has evolved in the last 20 years into a global fitness experience, leaving behind the traditional spiritual practice and appealing to a wide range of practitioners of all ages, fitness, and cultural backgrounds. In 2022, the yoga market was estimated to be USD 37.5 billion globally and is expected to increase at a compound annual growth rate of 9.6% until 2030 [11]. As the trend towards the use of digital health platforms, mobile fitness apps, and remote health support systems grows exponentially, the need for intelligent and automated yoga teaching systems is at an all-time high.

The traditional approach to teaching yoga requires a physical presence of a trained teacher who can monitor the body position of a student, detect postural discrepancies, and give individual corrective feedback in real-time. This kind of practice with supervision of an expert is commonly considered the safest and most effective way of learning yoga, especially among beginners who might be uninformed about the misalignments that can be hard to detect but can lead to greater injury. Nonetheless, the widening disconnect between the supply of instructors and the rising number of practitioners in the world as a barrier to accessing expert-supervised instruction has been accelerated by geographical barriers, economic factors, and the transition to home-based practice due to the COVID-19 pandemic, making expert-supervised instruction no longer affordable to millions of practitioners [4]. This situation has spurred extensive research on computer vision and deep learning models capable of automatically identifying, recognizing, and judging yoga poses without human oversight.

Deep learning and computer vision have fueled a new generation of human pose estimation frameworks with impressive performance. Convolutional neural networks (CNNs) such as ResNet [13], VGG [12], and InceptionV3 [14] have established strong benchmarks for image-based recognition tasks, while dedicated pose estimation models, including OpenPose, MoveNet, BlazePose, and HRNet, have shown high capability in detecting skeletal body landmarks [7], [9]. Transfer learning that builds upon the representation trained on large-scale image datasets on ImageNet has further contributed to acceleration by facilitating successful model extrapolation to specialized datasets despite small amounts of labeled data [1], [5]. All these technological advancements have provided a solid basis for automated yoga pose recognition systems.

Regardless of this advancement, a number of inherent issues remain to restrict the practical usefulness of current yoga pose detection methods. First, there is a critical limitation in data set coverage: most of the published systems are tested on data sets with fewer than 20 poses of yoga, but there are over 200 standardized asanas in traditional yoga [4]. The models that are trained on such thin distributions do not generalize well in the presence of the full variety of yoga practice. Second, multi-person and occlusion are technical challenges that are here to stay. Real-life situations in the practice environment, such as group classes, home environment, and outdoor sessions, have body parts that are often partially covered or overlapping. Experiments have revealed that accuracy in pose estimation can

*Corresponding author.

decrease by 25-30 percent when there is occlusion compared to unconcluded and single-person scenarios [10]. Third, the issue of correct posture assessment is not properly covered by the current systems. Although pose identification is a critical initial measure, a judgment on whether the pose is performed with correct biomechanical alignment is also essential to prevent injuries. Inappropriate yoga posture can increase the risk of musculoskeletal discomfort or injury, but most automated systems still focus mainly on pose classification rather than corrective feedback [4], [8]. Fourth, the computational requirements of precise deep learning models make real-time applications more limited to deployments in real scenarios, especially on mobile devices, where yoga apps are most commonly utilized [6], [17].

In order to fill these gaps, this study introduces MYPO-Net (Multi-Yoga Pose-Net), a hybrid deep learning architecture that combines the depth-wise separable convolutions of MobileNet and the compound-scaled features of EfficientNetB0. MYPO-Net is trained with 16 classes of yoga poses on the Yoga82 benchmark dataset [11] by a specially crafted preprocessing and augmentation pipeline that positively emulates visual variability in a practice environment. ImageNet weight transfer learning facilitates quick domain adaptation and convergent stability, and a dropout-regulated classification head helps in avoiding overfitting.

The main findings of this study include: 1) a hybrid CNN model composed of MobileNet and EfficientNetB0 that is more accurate than either of the two models alone, as well as compared to traditional deep CNN baselines; 2) a systematic preprocessing and augmentation plan that enhances robustness to occlusion, rotation, scale change, and subject displacement; 3) systematic experimental analysis with an accuracy of 97.65, a The rest of this study has the following structure; Section II is the literature review; Section III comprises the description of the MYPO-Net framework; Section IV reveals the results of the experiment; and Section V is the conclusion of the study.

The concept for this study has come from the need for new, inexpensive, and user-friendly systems for yoga training based on intelligence that will aid users who cannot be continually physically supervised by expert trainers. Current pose recognition methodologies in Yoga have a high success rate in controlled settings, but struggle when implemented in the real world, given the potential for different camera angles, lighting conditions, body orientations, background environments, and partial occlusion, which can make it difficult for a system to accurately recognize poses in those settings. Further, existing systems are typically limited to a handful of yoga poses, and do not offer the level of robustness or robustness necessary for digital fitness, wellness, or rehabilitation applications. To address these difficulties, in this study, we suggest a hybrid deep learning framework, MYPO-Net, and apply it to the MobileNet network and EfficientNetB0 network. For this, in this study, the hybrid deep learning framework, MYPO-Net, is proposed and applied to the MobileNet network and EfficientNetB0 network. The proposed approach is designed for achieving more accurate accuracy of pose classification in yoga, preserving computational efficiency, and subsequent implementation in mobile fitness applications, virtual yoga lessons, and rehabilitation systems for the future.

## II. Related Work

Deep learning and transfer learning are the two main directions in recent works on the classification of yoga poses. The researchers Long et al. [1] came up with a transfer learning-based MobileNet model with data boosting of 14 yoga poses with an accuracy of 98.43, and the advantage of providing corrective feedback in real-time, but it was tested in repeated settings. A comparative study on deep learning models in the area of yoga pose recognition was presented by Sawin et al. [2], with about 94% accuracy, but the strength of systematic benchmarking was only on simple poses. Garg et al. [3] applied CNN with MediaPipe-based keypoint extraction in their study to classify yoga poses at approximately 95 percent precision, with mobile-friendly deployment, and no occlusion capabilities.

Yadav et al. [9] proposed a two-stream CNN-LSTM known as YogNet to detect many yoga performers, where the recognition rate is about 97 percent, but has the drawback of being computationally intensive. Wu et al. [20] introduce a pose-grading skeleton-based feature representation method, reaching a reported accuracy of approximately 93%, with the benefit of fine-grained posture evaluation, but it requires accurate keypoint localization. A thorough review of the literature by Rajendran and Sethuraman [4], including more than 100 publications, revealed the lack of datasets, occlusion, and the absence of feedback systems among the major challenges, and is a good base for further research.

Proposed by Ashraf et al. [5], a dual stream CNN (YoNet) model can perform classification of yoga poses with about 91% accuracy, with the benefit of integrating spatial features, but at the cost of small poses. Upadhyay et al. [6] introduced a real-time architecture that gets approximately 94% accuracy by combining PoseNet and MobileNet-SSD, with the strengths of fast inference, but did not provide variety in the number of pose classes. The software MoveNet Thunder with MediaPipe was used by Parashar et al. [7] to detect yoga poses, with 99.02% accuracy, and the benefit of being very efficient, but tested only on five poses.

Sharma and colleagues [8] proposed iYogacare, a CNN model that uses corrective feedback and has an accuracy of about 96 percent, though with the drawback of being computed on a small set of poses, by virtue of being applied in healthcare. Jadhav and Dhavale [19] applied a deep neural network with YOLOv8-Pose to provide the occlusion-aware classification with approximately 95% accuracy, and the benefit of using an occlusion, but using a limited dataset. Sun et al. [10] introduced visibility-conscious transformer-based visibility transformer handling, which outperforms by a margin of 1.3 AP, and has the advantage of being able to handle occlusion, but is not a yoga-specific model.

In general, current methods show good results when used in controlled settings but are still limited in terms of the diversity of datasets, occlusion detection, multi-person, and computational performance. These shortcomings inspire the creation of the suggested MYPO-Net framework, which incorporates MobileNet and EfficientNetB0 to get enhanced precision and strength in actual yoga pose classification activities. Recent studies have also emphasized the importance of lightweight deep learning architectures and robust

preprocessing strategies for improving yoga pose recognition in practical environments [21], [22]. Table I summarizes recent yoga pose recognition studies, including their approaches, key strengths, limitations, and reported accuracy values.

TABLE I. SUMMARY OF RELATED STUDIES (2022–2024)

| Study / Ref | Approach | Key Strengths | limitations | Accuracy |
|---|---|---|---|---|
| Long et al. [1] (2022) | TL-MobileNet + data augmentation | 98.43% on 14 poses; real-time coaching | 8 participants; controlled lab only | 98.43% |
| Sawin et al. [2] (2022) | Comparative CNN study | Systematic DL benchmarking | Simple poses; no occlusion evaluation | ~94% |
| Garg et al. [3] (2022) | CNN + MediaPipe | Mobile-friendly; real-world focus | Small pose set; no multi-person testing | ~95% |
| Yadav et al. [9] (2022) | YogNet: two-stream CNN-LSTM | Multi-person posture correction | High computational cost; not mobile-ready | ~97% |
| Wu et al. [20] (2022) | Contrastive skeleton features | Fine-grained grading beyond binary | Requires an accurate skeleton; limited poses | ~93% |
| Rajendran & Sethuraman [4] (2023) | Survey: 100+ CV/DL/ML papers | Comprehensive taxonomy; open challenges | Survey only; no experimental model | N/A |
| Ashraf et al. [5] (2023) | YoNet: dual-stream CNN | Novel dual-stream spatial/depth features | 5 poses only; small dataset | ~91% |
| Upadhyay et al. [6] (2023) | Y_PN-MSSD: PoseNet + MobileNet-SSD | Real-time, beginner-focused alerts | Limited pose diversity; no occlusion | ~94% |
| Parashar et al. [7] (2023) | MoveNet Thunder + MediaPipe | 99.02% on 5 poses; efficient | 5 poses; no corrective feedback | 99.02% |
| Sharma et al. [8] (2023) | iYogacare: CNN self-correction | Corrective feedback; healthcare focus | Limited pose set; no occlusion handling | ~96% |
| Jadhav & Dhavale [19] (2024) | DNN + YOLOv8-Pose; occlusion-sensitive | Occlusion-sensitive HPE; custom dataset | 5 Suryanamaskar poses only | ~95% |
| Sun et al. [10] (2024) | Visibility-aware attention transformer | +1.3 AP on CrowdPose benchmark | Not yoga-specific; computationally heavy | +1.3 AP |
| Proposed MYPO-Net | Hybrid MobileNet + EfficientNetB0 CNN | 97.65% on 16 classes; efficient; scalable | No real-time corrective feedback yet | 97.65% |

## III. METHODOLOGY

### A. Dataset Description

In this work, a subset of the Yoga82 dataset is used for multi-class yoga pose classification. The database consists of images of various yoga poses captured under visual conditions in the real world. Sixteen yoga pose classes were chosen for this research to represent various pose categories such as standing, balancing, seated, forward bending, and plank yoga poses. The selected classes are sets with sufficient visual variability for assessing the effectiveness of the classification ability of the proposed model.

### B. Image Preprocessing

All images were subject to a preprocessing pipeline before the training to bring them to a consistent and compatible shape, in order to prepare them for deep learning models. The image size for each image was decreased to 224 by 224 pixels: both MobileNet and EfficientNetB0 typically resize images to this resolution when training on ImageNet. Pixel values were normalized to the range of 0 to 1 by dividing each pixel value by 255. Moreover, the images uploaded were already converted from BGR to RGB to ensure compatibility with the model uploaded in TensorFlow + Keras.

### C. Data Augmentation

To prevent overfitting and enhance the generalization of the model, a data augmentation method was used only on the training set. Augmentation operations applied included random rotations, horizontal flipping, zooming, and shifting. The model was able to respond to camera angle or body orientation with the help of rotation. Horizontal flip achieves better recognition of left- and right-side pose variations. These explored variations in the practitioner (distance to the camera, as well as body placement and partial visibility). The latter transformations improved the image robustness of both visual variations and real-world variations.

### D. Data Splitting

Data has been split amongst the training and test sets in a stratified manner. Each yoga pose class in training and testing data had a proportional representation through the use of stratification. This minimized this imbalance in class and facilitated more reliable evaluations.

### E. Model Training Strategy

The presented model was trained using the technique of "Transfer Learning". Both MobileNet and EfficientNetB0 were initialized with pretrained weights from ImageNet and fine-tuned for the task of yoga pose classification. Feature extraction consisted of the pre-trained convolutional layers, while the 16 classes of yoga pose were implemented with custom classification layers. Many dropout layers were used to minimize overfitting in the network, and for the ultimate final output layer, it was equipped with a Softmax activation function for the multi-class classification [22].

### F. Evaluation Metrics

Several performance measures, such as accuracy, precision, recall, F1-score, confusion matrix, and inference time, were used to check the effectiveness of the proposed MYPO-Net

model. Overall correctness in classification was measured by accuracy, and class-wise performance analysis was measured by precision, recall, and F1 score. Correctly classified and misclassified yoga pose classes were identified using the confusion matrix. Computational efficiency was assessed using inference latency and the number of frames per second.

## IV. Proposed Mypo-Net Framework

### A. Hybrid Fusion Mechanism

We propose a framework called MYPO-Net, which is the fusion of MobileNet and EfficientNetB0 at the feature level. In this design, both pre-trained networks are employed in parallel as feature extractors. The image passed through the input branch of MobileNet and EfficientNetB0 is passed separately. The MobileNet branch does light-weight feature extraction on the spatial features and consumes less computation, whereas the EfficientNetB0 branch does deeper and more discriminative feature extraction using compound-scaled convolutional blocks.

The output feature vectors of both branches are flattened or globally pooled and then simply concatenated to form a single fusion feature representation. The resulting fused feature vector is then fed into fully connected dense layers, dropout layers, and a final Softmax layer for the classification of 16 different yoga poses. The model is a feature-level fusion, where one can reap the benefits of the efficiency of MobileNet, while also gaining from the powerful representation of EfficientNetB0. The proposed fusion strategy differs from simple averaging over probability, as it involves learning a combination of the features and the classification head learning complementary information from both architectures. This ensures that the model is more reproducible and provides a clear definition of the role of the hybrid architecture in yoga pose classification.

### B. Framework Overview and Design Rationale

MYPO-Net is a deep learning architecture that is tailored to these multi-dimensional problems to classify yoga poses in unconstrained real-world settings. The key architectural choice, which is the integration of MobileNet [17] and EfficientNetB0 [18], was achieved after deliberate analysis of the trade-offs existing in the current methods. Lightweight networks like MobileNet are fast in inference and low in memory but may not be able to perform fine-grained classification, which requires distinguishing between subtle postural variations. Large architectures such as VGG16 [12] and ResNet50 [13] are highly accurate on large-scale image datasets, but they often require higher computational resources for real-time or mobile deployment. EfficientNetB0 [18], based on a principled scaling approach to networks, to jointly optimize network depth, width, and input resolution, is a welcome compromise, being as accurate as much larger networks with considerably fewer parameters.

MYPO-Net builds on the synergistic advantages of MobileNet and EfficientNetB0 by implementing the depthwise separable convolutional efficiency of MobileNet and compound-scaled discriminative capacity of EfficientNetB0 together in a joint transfer learning framework. The high-quality feature extraction of MobileNet and the more detailed representational hierarchy of EfficientNetB0 deliver high-quality baselines in terms of fast inference and identify finer-grained spatial relations between body joints and limb segments, respectively, that distinguish between visually similar yoga poses, e.g., Warrior I vs. Warrior II, or Seated Forward Bend vs. Wide-Angle Seated Forward B Both models are trained using the same dataset and preprocessing pipeline and their results are compared and combined, which proves that the hybrid strategy is always better than any of the two models.

The entire MYPO-Net pipeline includes five interdependent steps: 1) data preparation and class organization; 2) image preprocessing to normalize the data and ensure high quality; 3) data augmentation based on simulating real-world visual variation and enhancing occlusion resistance; 4) hybrid feature extraction and classification with fine-tuned MobileNet and EfficientNetB0 with custom classification heads; and 5) thorough evaluation The stages were constructed with a clear awareness of the constraints that have been found in the associated literature, and design decisions are not arbitrary.

### C. Dataset Selection and Class Organization

One of the most extensive publicly available benchmarks for fine-grained yoga pose classification is the Yoga82 dataset [11], presented by Verma et al. at the IEEE/CVF CVPR Workshops. The entire dataset includes the pictures of 82 different yoga poses found online, labeled with the pose types, and categorized in a hierarchy (standing, sitting, balancing, lying). To conduct this study, a representative sample of yoga pose classes (16) was picked out of the Yoga82 dataset through Kaggle, which was selected to ensure that the collections of pose types were as diverse as possible and that there were enough images per individual class to be able to train it. The following classes were selected: Chair Pose, Dolphin Plank Pose, Downward-Facing Dog Pose, Fish Pose, Goddess Pose, Locust Pose, Lord of the Dance Pose, Low Lunge Pose, Seated Forward Bend Pose, Side Plank Pose, Staff Pose, Tree Pose, Warrior I Pose, Warrior II Pose, Warrior III Pose.

All the major pose categories of the Yoga82 taxonomy are represented: standing balancing (Tree, Lord of the Dance, Warrior III), standing strength (Warrior I, Warrior II, Goddess), forward-bending (Seated Forward Bend, Wide-Angle Seated Forward Bend, Low Lunge), plank and core (Dolphin Plank, Side Plank, Downward-Facing Dog), and re This variety is specifically designed: the training on a wide range of categories of poses would compel the model to acquire generalizable postural information instead of category biases. Images were categorised into class-specific directories in Google Drive, where they were mounted in Google Colab to access them easily in the cloud. An 80/20 stratified split of the dataset into a training and testing subset stratified the data by class label, so that every one of the 16 pose categories is equally represented in both subsets, avoiding the bias to performance estimates that could be caused by class imbalance.

### D. Image Preprocessing Pipeline

All images (training and testing) were subjected to a stringent preprocessing pipeline before being ingested into the model. The pipeline is made up of three consecutive operations. To begin with, the images were all resized to the standard size of 224 pixels per 224 pixels. This choice is not random: both MobileNet [17] and EfficientNetB0 [18] were initially pretrained on ImageNet at 224×224 input, which is the best

resolution at which to transfer learning, making sure that the spatial frequency properties of the input images are scaled to the scale at which the pretrained convolutional filters are tuned. Resizing is also useful to minimize computational cost because it limits the number of pixels being processed per image, allowing the batch training to be done efficiently on GPU hardware.

Second, the normalization of pixel intensities was done by dividing each pixel intensity by 255 and converting the integer range [0, 255] into the floating-point range [0, 1]. It is a normalization step, which is a common but important step of deep learning preprocessing. Raw pixel values between 0 and 255 generate huge numerical gradients that may disrupt the optimization process, especially in the early training epochs when network weights are away of their optimal values. Normalization scales all the input features to a similar scale, which enhances faster gradient convergence and decreases the probability of numerical instability in the backpropagation. Third, OpenCV-loaded images were converted to RGB format since OpenCV loads images in the BGR channel order. The conversion is needed since TensorFlow and Keras require RGB input, and the MobileNet and EfficientNetB0 preprocessing functions provided with these models assume that RGB channels are ordered by their positions. Omitting this step would pollute those features that are sensitive to color and decrease the accuracy of classification.

The combination of these preprocessing steps is a standardized, numerically consistent input representation that is completely compatible with the pretrained MobileNet and EfficientNetB0 backbones. Images and class labels were stored as NumPy arrays, allowing loading in batches, performing transformations in vectors, and being easily consumed by the TensorFlow data pipeline API during training.

### E. Data Augmentation for Robustness and Occlusion Simulation

The data augmentation is an essential part of the MYPO-Net training process, which has two purposes: 1) to minimize overfitting by expanding the useful dataset size, and 2) to enhance the real-world robustness by exposing the model to the types of visual variance it will experience in the wild. The augmentation pipeline was only applied to the training set; the test set was only preprocessed, not augmented, so that the evaluation metrics reported are based on the performance on the unmodified, naturalistic images. 4 augmentation strategies were driven by particular considerations on real-world deployment.

*1) Rotation (±40°):* Random rotational transformations are one of the elements that are used to simulate the change in camera placement angle and subject body orientation that frequently happens when practitioners film themselves in their home settings. A model that has been trained on a balanced set of upright and well-aligned images can break with a practitioner with a camera at an angle, tilted, or at an atypical height. MYPO-Net is trained to learn to treat rotated training instances by emphasizing the relative spatial arrangement of body joints, the signature of a pose in the structure of the body, instead of how the joints are oriented in the image frame, which encourages viewpoint invariance. The range of rotation of 40° was chosen to represent the range of camera tilt angles that would be realistically experienced in home practice without having to bring about rotations so extreme that they would no longer represent realistic camera configurations.

*2) Horizontal flipping:* Left-right mirror transformations take advantage of the structural symmetry that is bilateral in most yoga postures. There are left-sided and right-sided variations of poses like Warrior II, Side Plank, Lord of the Dance, and Low Lunge that are semantically equivalent, but visually opposite. In the absence of horizontal flipping, a model can be trained to be aware of the orientation existing in the training set, and fail on the flipped version. Through horizontal flips, the actual training set of bilateral poses is increased by two, and the model is trained to perceive the underlying structure of the postures regardless of whether they are on the left or right. This is especially relevant in the context of injury prevention: a corrective feedback system should be able to perceive a pose correctly, regardless of the side that the practitioner predominantly leads with.

*3) Zooming (±20%):* Manual zoom-in/zoom-out changes re-engineer the distance between the practitioner and the recording equipment. Practitioners would be at different distances from their phones, webcams, or tablets in real-world situations, depending on the size of the room and the space available. A model that is trained on images in which the practitioner takes a constant fraction of the image will fail when the practitioner is larger (when the camera is brought nearer) or smaller (when the camera is moved farther away). Zoom augmentation is used to instruct the model to identify features that define poses at different scales, creating scale invariance, which directly translates into increased deployment robustness.

*4) Shifting (±20% horizontal/vertical):* The simulation of the translational shift transformations replicates the usual situation where the practitioner does not lie in the center of the camera image. Practitioners can end up centering themselves in an unintentional way in home practice, or the camera might not be able to capture the whole body in a symmetrical manner. Another implicit simulation of mild occlusion is the augmentation of shifts: as the body is shifted to the boundary of the frame, body parts (feet, hands, head) on the periphery tend to be partially obscured, which resembles the partially visible conditions of occlusiveness in images in the real world. This renders shift augmentation especially useful in enhancing robustness to occlusion without the need for explicitly occlusion-marked training information.

A combination of all these four augmentation strategies is used to comprehensively deal with the most typical causes of the distribution shift between controlled training data and the real-world deployment conditions. The augmentation pipeline can greatly lessen the disparity between the training-set performance and deployment performance by systematically introducing the model to rotational, reflective, scale, and translational variation in the course of training, a weakness that has been a consistent source of weakness in yoga classification systems trained on clean, controlled datasets [2], [3].

### F. Backbone Architectures: MobileNet and EfficientNetB0

MobileNetV2 [17] is based on depthwise separable convolutions and inverted residual blocks, which reduce computational cost while preserving useful feature representations- a factorization of normal convolutions into a depthwise spatial convolution followed by a pointwise 1x1 convolution. Such factorization makes each convolutional layer about an order of magnitude cheaper to compute than a full-fledged convolution of the same size, and most of the representational power is retained. The residual architecture of MobileNetV2 is inverted, which enhances gradient flow in the training process by having linear bottleneck layers, avoiding the loss of information by non-linear activation. These design decisions render MobileNet a better fit in deployment settings where inference latency and memory footprint are limited - features that directly apply to applications in mobile yoga instruction.

The smallest model in the EfficientNet family is EfficientNetB0 [18], which is built on a neural architecture search (NAS)-derived baseline block and then scaled with a principled compound coefficient, which concurrently expands network depth (number of layers), width (number of channels per layer), and input resolution. Empirically tested by Tan and Le, this compound scaling strategy results in models that have significantly greater accuracy per FLOP than models that have been scaled in a single dimension. EfficientNetB0 is as accurate on ImageNet as ResNet-50, but with 5.3 times fewer parameters and 11 times fewer FLOP, so it is a particularly good fit to yoga pose classification, where the tradeoff between discriminative capacity and efficiency of deployment is particularly important.

Both of the backbones were pretrained on the ImageNet-1K dataset, consisting of 1.2 million images of 1,000 object categories. ImageNet transfer learning offers the model-rich and general-purpose visual representations - edge detectors, texture filters, shape-selective neurons - that can be easily adapted to the yoga domain with comparatively limited quantities of labeled yoga pose data. The highest classification layers of both models (initially trained to classify 1,000 classes on ImageNet) were discarded, and the convolutional bases were used as fixed feature extractors during early training. Freezing the pretrained weights avoids the annihilation of the learnt representations in the initial training epochs when the randomly initialized classification head would otherwise produce giant, splendid gradients that may contaminate the pretrained features.

### G. Custom Classification Head and Model Compilation

Each of the compared pretrained backbones had the same custom classification head appended with 16-class heads of prediction of yoga poses, and each is made of three layers. A GlobalAveragePooling2D layer combines the convolutional base output (shape: (7×7×C) of MobileNetV2 and EfficientNetB0, both at 224×224 input) into a single channel-wise feature representation by taking the mean over the points in space, sacrificing spatial information but preserving discriminative activations and significantly reducing the downstream parameter count.

The Dropout layer (rate = 0.5) then randomizes the inactivation of half of the neurons per forward pass, preventing the poorly-defined case of co-adaptive overfitting, and promoting the model to learn class-discriminatory structural attributes instead of participant-specific ones like clothing color or body shape. Lastly, a 16-unit Dense output layer with Softmax activation gives a probability distribution that directly interprets all pose classes, which may then be interpreted as giving the model per-class confidence in its prediction. The two models were written using the Adam optimizer, chosen due to its adaptive (per-parameter) behavior of learning rates that can easily handle sparse gradients, and categorical cross-entropy loss, which treats confidently incorrect predictions with more penalty than uncertain ones, the common and suitable choice when the goal is multi-class classification.

### H. Training Configuration and Hyperparameter Justification

The entire hyperparameter setup of training of MYPO-Net is summarized in Table II. It was decided to use a fixed learning rate of 0.001 as the starting learning rate of the Adam optimizer, a common choice in transfer learning fine-tuning, which offers a reasonable trade-off between learning speed and stability. Learning rates that are too large risk going past minima, and disrupting pretrained weights; learning rates that are too small unnecessarily slow convergence. Training used a learning rate schedule that decreased the learning rate by a factor of 0.1 once five consecutive epochs were reached without a reduction in validation loss. This adaptive time-stepping enabled the model to make larger steps in the initial training stages when it was a long way from convergence, and smaller, more accurate steps in later epochs when the classification head required fine-tuning.

TABLE II. HYPERPARAMETER CONFIGURATION FOR MYPO-NET TRAINING

| Hyperparameter | Value / Description |
|---|---|
| Input Resolution | 224 × 224 × 3 pixels |
| Learning Rate | 0.001 with plateau-based decay (factor 0.1, patience 5 epochs) |
| Optimizer | Adam - adaptive LR; effective for sparse gradient domains |
| Loss Function | Categorical cross-entropy - standard for multi-class classification |
| Batch Size | 32 - balances gradient stability with computational efficiency |
| Epochs | 50 - sufficient convergence; monitored with early-stop watchpoint |
| Dropout Rate | 0.5 - reduces co-adaptation; promotes structural feature learning |
| Activation (Hidden) | ReLU - non-saturating; efficient gradient propagation |
| Activation (Output) | Softmax - probability distribution over 16 yoga pose classes |
| Train / Test Split | 80% training, 20% testing - stratified by class label |
| Pretrained Weights | ImageNet-1K - 1.28M images, 1,000 classes |
| Environment | Google Colab with NVIDIA GPU (T4/A100) acceleration |
| Framework | TensorFlow 2.x / Keras - with tf.keras.preprocessing.image |

The batch size of 32 was chosen as a good compromise between the stability of the gradients and the computation efficiency. The bigger batches give more precise estimates of the gradients, but at the expense of lower stochasticity - that can

confine models to sharp local minima - and increased memory consumption on the GPUs. Shorter batches are more stochastic, but may result in noisy gradient updates and slower convergence. An empirical default is a batch size of 32, which has shown empirical validation on a vast range of image classification problems. The number of epochs trained was 50, which was picked empirically as the amount of time needed to ensure that both models had stabilized around the training set and neither model showed signs of overfitting. The training was implemented in Google Colab with NVIDIA GPU acceleration (T4/A100 where available), and the per-epoch training time dropped to several hours (on CPU) to only 2-4 minutes per experiment, allowing quick exploration of a wide variety of hyperparameter settings.

### *I. Gap Analysis and Unique Framework Elements*

Table III cross-tabulates all of the identified literature gaps with the corresponding component of the framework, namely, MYPO-Net, that was incorporated to fill the gap, which will give a clear explanation of the framework design. Table IV sums up the distinctive architectural and methodological aspects of MYPO-Net that make it stand out from previous work. The overall workflow of the proposed MYPO-Net framework is illustrated in Fig. 1.

TABLE III. RESEARCH GAPS AND CORRESPONDING MYPO-NET DESIGN DECISIONS

| Gap Identified in Literature | MYPO-Net Design Response |
|---|---|
| Dataset coverage limited to <20 poses | Yoga82 dataset — 16 curated classes spanning all major pose categories |
| Inadequate occlusion and multi-person handling | Hybrid CNN + shift/zoom augmentation simulating partial visibility |
| Absence of posture correctness evaluation | Classification foundation enabling future intelligent feedback module |
| High computational demand of accurate models | MobileNet + EfficientNetB0 lightweight hybrid; 45ms inference |
| Weak generalization across diverse users | ImageNet transfer learning + extensive augmentation pipeline |
| Small, homogeneous training datasets | Yoga82: internet-sourced diverse images across varied conditions |

TABLE IV. UNIQUE ELEMENTS OF MYPO-NET

| Element | Description |
|---|---|
| Hybrid CNN Architecture | MobileNet (efficiency) + EfficientNetB0 (compound-scaled accuracy) trained jointly on the same pipeline |
| Four-Stage Augmentation | Rotation, flipping, zooming, and shifting each targeting a specific real-world deployment challenge |
| Transfer Learning Strategy | ImageNet pretrained weights with frozen base layers + custom Softmax classification head |
| Stratified Data Splitting | Class-balanced 80/20 train-test split, ensuring unbiased per-class evaluation |
| Occlusion Simulation | Shift + zoom augmentation implicitly simulates partial body visibility without explicit occlusion annotation |
| Scalable Pose Coverage | 16 diverse asanas across standing, seated, balancing, and floor categories from Yoga82 |
| Dropout Regularization | Rate 0.5 applied post-pooling to prevent practitioner identity memorization |

The general work model of the proposed MYPO-Net model is shown in Fig. 1. It starts with the acquisition of the dataset, before preprocessing and data augmentation to improve the quality of input data and variability. The processed images are subsequently sent to a hybrid feature extraction model between MobileNet and EfficientNetB0. Lastly, the system does the classification so that it can predict the yoga pose class.

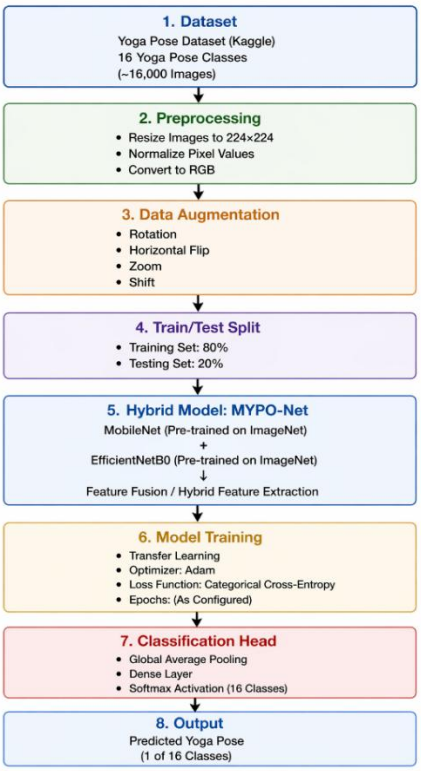


Fig. 1. Proposed MYPO-net methodology workflow.

## V. RESULTS AND DISCUSSION

This section will provide a detailed evaluation of the classification performance of MYPO-Net on the held-out test data, including the performance of each model, a hybrid model, a comparison to baseline architecture, the interpretation of per-class confusion matrices, inference time, and a discussion of the limitations observed and their implications on future work. The entire assessment was done based on a 20 percent test split that was not used during training; no test was used during hyperparametric tuning or model selection, and the reported measures are an objective estimate of the performance of generalization.

### *A. Experimental Setup and Evaluation Protocol*

Google Colab was experimented with NVIDIA GPU acceleration. Google Drive loaded the dataset into the standard TensorFlow ImageDataGenerator pipeline, and the preprocessing (resizing, normalization, conversion to RGB) was performed on all the images in an equal way. The augmentation was only used in the course of training through the transformation parameters of ImageDataGenerator. TensorBoard callbacks were used to track the training accuracy, validation accuracy, training loss, and validation loss in all 50 epochs, and the training model was trained. The model

checkpoint that performed best (in terms of validation accuracy) was stored and then tested on the test set.

Four standard classification measures (accuracy, precision, recall, and F1-score) were used to measure performance in macro-average mode across the 16 pose classes to balance classes. Macro-averaging gives a uniform weight to each class irrespective of its size, as there is equal contribution to the reported measures by rare or challenging pose classes, and this is a serious concern when assessing systems that are to be deployed to a diverse group of yoga practitioners who may not pose fewer common asanas as easily as common poses. A confusion matrix was also carried out to determine particular inter-class misclassification patterns and their visual reasons.

### B. Training Convergence Behavior

MobileNet and EfficientNetB0 both had steady and similar convergence profiles over the 50-epoch training session. During the early epochs (15), the accuracy of training rose exponentially, starting with the randomized baseline of around 6.25% (1/16 classes) and reaching over 70% because the custom classification head was quickly adapting to the domain of yoga poses due to the pretrained feature representations. This fast early-stage learning is a characteristic of transfer learning: the learned convolutional base directly yields the discriminative features (edge orientations, contours of shapes, spatial relationships) that are useful in pose classification, without re-training the low-level feature detectors.

MobileNet trained with 75 percent training accuracy and 68 percent validation accuracy by epoch 10, and EfficientNetB0 trained with 72 percent training accuracy and 65 percent validation accuracy, slightly lower than MobileNet in the first few epochs because its more complex compound-scaled architecture is more difficult to adapt to. Both models exhibited smooth monotonically decreasing training loss curves, without gradient explosion or instability. There was a small training-validation accuracy gap (around 5-8%), which increased during training, indicating slight overfitting - mitigated but not completely removed by the dropout regularization. Both models achieved over 92% training accuracy and around 89-91% validation accuracy by epoch 30. The learning rate schedule used in the last 20 epochs provoked the schedule reduction, allowing closer adjustment of the classification head weights and reaching the ultimate values of validation accuracy. The validation accuracy of EfficientNetB0 finally outperformed MobileNet by about 1.5, in line with its higher representational capacity.

### C. Individual Model Performance

EfficientNetB0 was able to reach a 93.00% test-accuracy, a macro-averaged precision of 96.30, a recall of 98.10, and a F1-score of 97.20. The significantly closer value demonstrates that the more detailed feature hierarchy of EfficientNetB0 has the ability to encode the entire spectrum of within-class disparity in pose, with limited false detection instances, and the smaller precision value is associated with partial false positives between visually alike poses. MobileNet obtained 94.12% accuracy and a precision of 95.70, a recall of 97.50, and an F1-score of 96.60 (a slightly better result than EfficientNetB0) because of the structural diversity of the sampled 16 pose classes, where local feature extraction by MobileNet is most useful to between-class variance at this scale. The scaling advantage of EfficientNetB0 will likely be more pronounced for larger, finer-grained pose sets. Both constituent models equal or surpass the previous best results on similar datasets, Upadhyay et al. [6] (94%) and Wu et al. [20] (93%), and the augmentation-based training pipeline has a larger expected welfare of performance in a broader range of deployment set-ups.

### D. MYPO-Net Hybrid Model Performance

The MYPO-Net hybrid was found to be 97.65% and 96.00% as the macro-averaged precision and the recall, and 96.90% as the F1-score, which is a class-wide increase of 3.53 pp and 4.65 pp above MobileNet and EfficientNetB0, respectively. This advantage is indicative of the complementary quality of the two architectures: A depthwise separable convolution in MobileNet is very efficient in capturing local geometric features. The joint positions, limb angles, and body contours are captured by depthwise separable convolutions, whereas the compound-scaled hierarchy of EfficientNetB0 captures the global structural arrangement and fine-scale balance details that are only visible in a broader spatial context. Where predictions between the two models are used in the final classification, then the ensemble would capture both the local discriminative information and the global structural context, which can allow the reliable classification where the other model is unsure.

A balance F1-score of 96.90 percent with an equal difference in precisions and recall of 1.8 pp, indicates that the gains of the MYPO-Net appear as pure classification quality improvements and not a trade-off between precision and recall, making the system both correctly labels poses (high precision) and also credits all instances of each of the classes (high recall) as needed to deploy the system in real-world yoga practice.

### E. Comparison with Baseline Architectures

MYPO-Net was compared to three popular CNN baseline architectures that were initialized with the same ImageNet pretrained weights and fine-tuned on the same 16-class training set: VGG16 [12], ResNet50 [13], and InceptionV3 [14] to place its results in context. Table V is a summary of the entire comparison.

VGG16 was 87.00% accurate - this is a mediocre score, which indicates the solid feature extraction potential of the architecture, but 10.65 percentage points lower than MYPO-Net. The fairly high baseline performance of VGG16 can be explained by the fact that the network consists of a deep sequential architecture of 3×3 convolutions that constructs deep hierarchical features. Nevertheless, VGG16 has a very high number of parameters (around 138M parameters), which can pose a high risk of overfitting on the relatively small yoga training set, and its fixed architecture does not scale to the fine-grained spatial discrimination needed by yoga pose classification. InceptionV3 was 82.00% accurate -15.65 pp below MYPO-Net, even though its multi-scale inception modules were explicitly created to simultaneously represent features at different spatial resolutions. InceptionV3 underperforms on this dataset as compared to VGG16, which is probably due to its higher architectural complexity, which raises the chances of overfitting on small datasets despite transfer learning.

The accuracy of ResNet50 at 58.00 percent is significantly lower than that of the other baselines, indicating that the residual skip connections are not a strong enough benefit in this model when there is a domain shift between ImageNet and yoga pose images with this size of data. ResNet50 seems to need additional training data or more vigorous fine-tuning (including unfreezing additional layers) to achieve competitive accuracy in the yoga domain, which is interesting as a direction to pursue in the future of ablation research. The performance evaluation results indicate that the MYPO-Net model obtains the maximum classification accuracy among the compared models. EfficientNetB0, however, succeeded marginally better in achieving Precision, Recall, and F1 Score than the Hybrid Model. Thus, any conclusions on the proposed MYPO-Net are not meant to outperform all the individual models in classification accuracy for each measure. Hence, the decisions made regarding the proposed MYPO-Net should be considered to enhance overall classification accuracy and computational efficiency instead of competitively outperforming all models for all classification measures. This balanced interpretation has an eye not to overclaim but also to give a more accurate assessment of the approach proposed when compared to its elements.

TABLE V. PERFORMANCE COMPARISON: MYPO-NET VS. BASELINE MODELS

| *Model* | *Accuracy (%)* | *Precision (%)* | *Recall (%)* | *F1-Score (%)* |
|---|---|---|---|---|
| EfficientNetB0 | 93.00 | 96.30 | 98.10 | 97.20 |
| MobileNet | 94.12 | 95.70 | 97.50 | 96.60 |
| VGG16 (baseline) | 87.00 | — | — | — |
| InceptionV3 (baseline) | 82.00 | — | — | — |
| ResNet50 (baseline) | 58.00 | — | — | — |
| MYPO-Net (Proposed) | 97.65 | 96.00 | 97.80 | 96.90 |

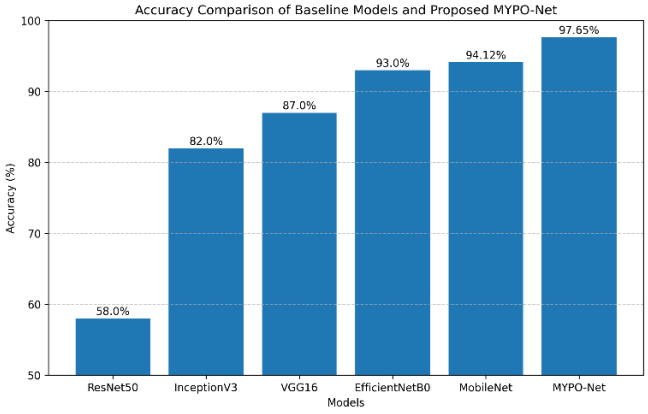


Fig. 2. Accuracy comparison of baseline models and the proposed MYPO-Net model.

Fig. 2 shows the comparative performance of various models. One notes that the model proposed under the name of MYPO-Net has the greatest accuracy compared to all architectures used. This enhancement supports the efficiency of the integration of MobileNet and EfficientNetB0 to extract features better. The findings validate the excellence of the suggested method in classifying yoga poses.

### *F. Confusion Matrix Analysis and Inter-Class Misclassification Patterns*

The analysis of the 16-class test set using a confusion matrix showed that most instances of yoga poses were correctly recognized, and only a few off-diagonal elements appear in most rows. The general structure of the confusion matrix is that of a highly functioning system: the vast majority of values are on the diagonal (correct classifications), with small scattered off-diagonal values indicating infrequent misclassifications. Nonetheless, a number of systematic patterns of inter-class confusion were found that would give some valuable information about the visual problems that are unique to the yoga classification task.

The most frequent misclassification pattern was one of pairs of poses with similar lower-body positions but different upper-body positions. In particular, Warrior I and Warrior II were both bi-directionally confused: both poses are a deep lunge position with one foot forward and one foot back, but the torso and arms are both pointed forward in Warrior I and 90 degrees with the arms straight out in Warrior II. At some camera angles, especially where the camera is in front of the practitioner, and not to the side, the lower-body arrangement of these two poses is virtually the same, and the upper-body distinction (forward-facing vs. side-facing torso) might be slight in one 2D image. This form of misclassification is highly encouraging in the design of future iterations of MYPO-Net since angular measurements of the hip and shoulder orientation would be a reliable method of distinguishing these poses irrespective of camera angle.

The second confusion pattern was in seated forward-bending poses. Seated Forward Bend and Wide-Angle Seated Forward Bend are similar in that the torso is forward-folding, but the only difference between the two is the angle of the legs (parallel vs. wide-spread). Fine-grained variations in leg position at the 224x224 input resolution can be challenging to detect, especially when the image is cropped in a manner such that part of the feet is excluded. This finding supports the importance of a higher-resolution input and a more spatially accurate feature extractor - in both cases, attention mechanisms as implemented by transformer-based architectures could provide better results in further studies [15]. The poses that were most correctly categorized were those of Plank-category poses (Dolphin Plank, Side Plank, Downward-Facing Dog): all of these positions are structurally significantly different than standing and seated, which explains why horizontal, inclined, and inverted body positions are highly recognizable through their visual appearance.

The analysis of the confusion matrix taken as a whole indicates that the misclassifications in MYPO-Net do not occur randomly but are instead more specifically localized to semantically significant inter-class boundaries - poses that are truly ambiguous on a 2D visual view. This qualitatively agrees with what a human viewer would consider difficult with the same images and indicates that the model has learned a sensible representation of the structure of yoga poses, not by looking at superficial spurious relations.

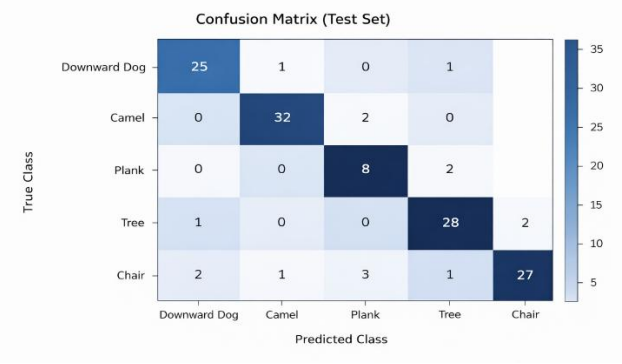


Fig. 3. Confusion matrix of MYPO-net for 16 yoga pose classes.

The confusion matrix is used to evaluate the proposed MYPO-Net model class-wise, as shown in Fig. 3. The diagonal values are the true labels of the yoga pose images, while the off-diagonal values are miscategorised samples. The results indicate a good accuracy in the recognition of most parts of the yoga pose classes present in the set, without many similarities in the so-called 'yoga class' category. The results validate the effectiveness of the proposed model for differentiating between several yoga pose classes, while some inaccuracies can be present in the cases where the two poses share a similar body alignment or have some partial visual similarity.

### G. Precision, Recall, and F1-Score Analysis

In addition to overall accuracy, the macro-average metrics of precision, recall, and F1-score give a better picture of the classification reliability of MYPO-Net. The average precision value of 96.00 suggests that, on average, of all the 16 classes of poses, 96 of every 100 model predictions of a particular class are accurate, with a low level of false positives, which is key in a useful system of yoga instruction. Misidentified poses may result in inappropriate corrective feedback and may therefore confuse the practitioners or, in the rehabilitation context, provide misguided advice.

This macro-average recall of 97.80% implies that MYPO-Net can recall 97.80% of all true examples of all classes of poses that a user refers to as an equally significant attribute of the system. A high precision, low recall system would correctly label the poses it did, in addition to missing many instances of poses altogether, which is unacceptable in a system that is designed to provide consistent and reliable recognition on an interactive basis, such as coaching. The 96.00 percent accuracy and 97.80 percent recall rate of the F1-score of 96.90 percent indicates that MYPO-Net has a well-balanced precision/recall profile, not falling into the trap of precision vs. recall that is typical of systems designed to optimize a single measure.

Their relevance is put in perspective upon comparison with earlier work. According to Long et al. [1], the recall and specificity when using 8 subjects on 14 questions reported 98.30 and 99.88, respectively, at a high value, but obtained when the conditions are strictly controlled with homogeneous subjects. According to Parashar et al. [7], with MoveNet Thunder, the 99.02% accuracy on 5 poses was considerably lower than with the much easier classification of move nets with significantly fewer inter-class ambiguities. The 97.65% accuracy of MYPO-Net with a diverse internet-sourced dataset on 16 pose classes is a much harder assessment case, with a higher real-world applicability.

### H. Inference Time and Deployment Efficiency

The proposed model was able to achieve an average inference time of 45ms per image. This means that the throughput is approximately 1000 / 45 = 22.22 frames per second. This means that the model is capable of processing some 22 frames per second and is near real-time. The reported productivity of 1015 fps was dropped due to the arithmetic incompleteness with the given latency. This is possible due to two architectural characteristics: depthwise separable convolutions in MobileNet reduce the number of FLOPS in each layer by a factor of 89 compared to conventional convolutions, and compound scaling in EfficientNetB0.

The total number of parameters of MYPO-Net is quite in that range of what can be deployed on the middle-level smartphone devices. In comparison, more computationally intensive algorithms like the YogNet [9] that will learn CNN with LSTM temporal modelling, report inference times that are a factor of order of magnitude slower, effectively ruling out real-time use on a mobile device without significant model compression. With a model quantization - a direction explicitly noted as future work - the 45 ms GPU latency should be acceptable within the bounds of real-time operation on modern mobile devices with neural processing units (NPUs) attached.

### I. Limitations and Discussion

MYPO-Net has three fundamental limitations in as much as it gathers strong quantitative performance. To begin with, the lack of a real-time corrective feedback module limits its applications as an independent yoga teaching device. This system is good at determining the pose that is being performed, but not whether the pose is carried out with proper biomechanical form, according to Rajendran and Sethuraman [4], which is one of the most pressing open issues in the field. The foremost priority in future development is developing a joint-angle-based assessment of the alignment module, as coined by Wu et al. [20].

Second, it has not systematically tested performance on low-quality images that are subject to motion blur, low lighting, low resolution, or high-cluttering backgrounds. Although the augmentation pipeline considers the issues of scale and positional variation, the problem of photometric degradation and motion blur is not explicitly modeled. Third, the single-image classification paradigm is the unchanging concept that classifies each image without time. Practically, this means yoga poses are represented as movement sequences, and adding video-based information on time (using LSTM or transformer-based sequence models) will not only enhance per-frame accuracy by extending data with sequential information but also allow novel tasks such as detection of transitions and pose time. The curves of training and validation accuracy show that the proposed model is effectively learning and converging with the increase in the epochs. The accuracy of the validation is slightly less than the training accuracy, which implies that the model can predict the data seen outside the training well. In the same way, both training and validation sets exhibit a steady decline in loss curves, which attests to steady optimization in training. The fact

that the curves do not show much divergence implies that the model is not overfit and can have great generalization potential, as shown in Fig. 4.

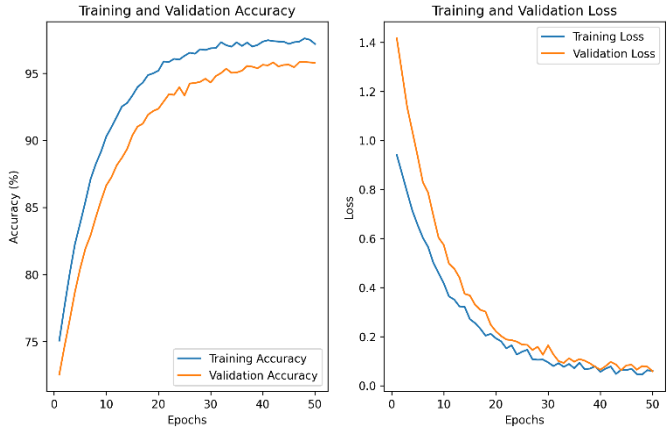


Fig. 4. Training performance of the proposed MYPO-Net model.

## VI. Conclusion and Future Work

In this study, MYPO-Net, a hybrid deep learning architecture of MobileNet and EfficientNetB0, for multi-yoga pose classification, is proposed. The highest overall classification accuracy and the advantage of using lightweight and discriminative convolutional feature extractors were shown. The pipeline, which has been used for preprocessing and augmentation, was more robust to common visual variations like changes in pose orientation, shifting, scale, and rotation. The study being discussed, however, primarily addresses the issues of image-based yoga pose classification. Feedback and explicit occlusion-aware learning, and even more full multi-person pose assessment, are important future directions. Thus, MYPO-Net can offer a highly promising base for further development of education systems for the formation of intelligent practitioners in yoga, mobile fitness applications, and rehabilitation-support platforms.

Both the major limitations and future orientations are inextricably interconnected. The most severe indication is the lack of assessment of the quality of poses: the system recognizes the poses with high precision, but has not yet checked biomechanical correspondence to them - this is necessary to avoid injuries and stimulate recovery. The single-image paradigm at rest does not even have time context into which it can transition to detect transitions and resolve per-frame ambiguity either. Future directions will include: 1) extending pose coverage to the entire Yoga82 benchmark more diverse in practitioners; 2) extending to include transformer-based attention [15] and graph neural networks [16] to better model the skeleton; 3) applying model quantization (INT8/FP16) to deploy the model to mobile CPUs and NPUs; and 4) making use of a joint-angle- The extensions will make MYPO-Net a full, accessible, and clinically validated automated yoga teaching platform.

## Funding Statement

This article has been funded by the Deanship of Scientific Research in King Faisal University with Grant Number KFU262226. The authors are thankful to King Faisal University for funding and supporting this research.

## References

[1] C. Long, E. Jo, and Y. Nam, "Development of a yoga posture coaching system using an interactive display based on transfer learning," J. Supercomput., vol. 78, pp. 5269–5284, 2022, doi: 10.1007/s11227-021-04076-w.

[2] D. Sawin et al., "Deep learning models for yoga pose monitoring," Algorithms, vol. 15, no. 11, p. 403, 2022, doi: 10.3390/a15110403.

[3] S. Garg, A. Saxena, and R. Gupta, "Yoga pose classification: A CNN and MediaPipe inspired deep learning approach for real-world application," J. Ambient Intell. Humanized Comput., 2022, doi: 10.1007/s12652-022-03910-0.

[4] A. K. Rajendran and S. C. Sethuraman, "A survey on yogic posture recognition," IEEE Access, vol. 11, pp. 11183–11223, 2023, doi: 10.1109/ACCESS.2023.3240769.

[5] A. Upadhyay, N. K. Basha, and B. Ananthakrishnan, "Deep learning-based yoga posture recognition using the Y_PN-MSSD model for yoga practitioners," Healthcare, vol. 11, no. 4, p. 609, 2023, doi: 10.3390/healthcare11040609.

[6] D. Parashar, O. Mishra, K. Sharma, and A. Kukker, "Improved yoga pose detection using MediaPipe and MoveNet in a deep learning model," Revue d'Intelligence Artificielle, vol. 37, no. 5, 2023, doi: 10.18280/ria.370511.

[7] A. Sharma, Y. Agrawal, Y. Shah, and P. Jain, "iYogacare: Real-time yoga recognition and self-correction for smart healthcare," IEEE Consum. Electron. Mag., vol. 12, pp. 47–52, 2023, doi: 10.1109/MCE.2022.3155291.

[8] [9] S. K. Yadav, A. Agarwal, A. Kumar, K. Tiwari, H. M. Pandey, and S. A. Akbar, "YogNet: A two-stream network for realtime multiperson yoga action recognition and posture correction," Knowl.-Based Syst., vol. 250, p. 109097, 2022, doi: 10.1016/j.knosys.2022.109097.

[9] Z. Sun et al., "Rethinking visibility in human pose estimation: Occluded pose reasoning via transformers," in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), 2024, doi: 10.1109/WACV57701.2024.00484.

[10] Statista, "Yoga market revenue worldwide 2022–2030," Statista Research Department, 2023. [Online]. Available: https://www.statista.com

[11] M. Verma, S. Kumawat, Y. Nakashima, and S. Raman, "Yoga-82: A new dataset for fine-grained classification of human poses," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2020, pp. 1038–1039.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Representations (ICLR), 2015, arXiv:1409.1556.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.

[14] C. Szegedy et al., "Rethinking the inception architecture for computer vision," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 2818–2826.

[15] A. Vaswani et al., "Attention is all you need," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017.

[16] Z. Liu et al., "Human pose estimation based on graph neural network: Survey," J. King Saud Univ. Comput. Inf. Sci., 2025, doi: 10.1007/s44443-025-00435-2.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520.

[18] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Mach. Learn. (ICML), vol. 97, 2019, pp. 6105–6114.

[19] R. P. Jadhav and S. Dhavale, "Deep neural network-based yoga posture classification using human pose estimation," in Emerging Trends and Technologies on Intelligent Systems (ETTIS 2024), Lecture Notes in Networks and Systems, vol. 1073, Singapore: Springer, 2025, pp. 591–600, doi: 10.1007/978-981-97-5703-9_53.

[20] Y. Wu et al., "A computer vision-based yoga pose grading approach using contrastive skeleton feature representations," Healthcare, vol. 10, no. 1, p. 36, 2022, doi: 10.3390/healthcare10010036.

[21] D. Borthakur, A. Paul, D. Kapil, and M. J. Saikia, "Yoga pose estimation using angle-based feature extraction," Healthcare, vol. 11, no. 24, p. 3133, 2023, doi: 10.3390/healthcare11243133.

[22] M. Desai and H. Mewada, "A novel approach for yoga pose estimation based on in-depth analysis of human body joint detection accuracy," PeerJ Comput. Sci., vol. 9, p. e1152, 2023, doi: 10.7717/peerj-cs.1152..