Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 260 (2025) 1000–1008

Procedia
Computer Science

www.elsevier.com/locate/procedia

Seventh International Conference on Recent Trends in Image Processing and Pattern Recognition (RTIP2R-2024)

Quantization Techniques for Optimizing MobileNetV3Large in Yoga Pose Recognition on Edge Devices

Komal S Totad[a], Abhishek R Hanchinal[a], Neha R Shanbhog[a], Trupti V Patgar[a], Praveen M Dhulavvagol[a,*]

[a] *School of Computer Science and Engineering, KLE Technological University, Hubballi, Karnataka, 580031*

Abstract

In recent years, advances in deep learning have made a major impact in fields such as computer vision and image classification, and convolutional neural networks (CNN) have played an important role in visual recognition such as yoga pose classification. However, this model is difficult to implement on edge devices with limited resources due to the large number of calculations and operations. Memory intensive needs. Quantitative methods offer solutions by reducing sample size and increasing inference speed while maintaining accuracy. In this work, we investigate how to apply quantization techniques to the MobileNetV3Large model for yoga pose classification. The main goals include using post-training quantization (PTQ) and quantization-aware training (QAT) to improve the model's performance, compare the performance of PTQ and QAT models, and send the best models to the edge. Kaggle's yoga poses dataset contains 3700 images of 43 poses, which were pre-processed and used to fine-tune the MobileNetV3Large model. Using the PTQ, the sample size increased slightly from 12.5 MB to 3.43 MB. clear. In comparison, the QAT model has a higher accuracy of 84.71% and a model size of 11.2 MB. These results demonstrate the effectiveness of the quantization method in optimizing Yoga's MobileNetV3Large model. Simplify deployment of edge devices by sharing beacons. Future research should focus on further refinement and experimental validation of these models to improve their use and support their development. Interactive and easy-to-use yoga practice tools for mobile and embedded platforms.

© 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the Seventh International Conference on Recent Trends in Image Processing and Pattern Recognition.

Keywords: Quantization Techniques; Post-Training Quantization (PTQ); Quantization-Aware Training (QAT); MobileNetV3Large; Deep Learning

1. Introduction

In recent years, profound advancements in deep learning have reshaped numerous domains, particularly in computer vision and image classification. Convolutional Neural Networks (CNNs) have emerged as formidable assets for

[*] Corresponding author.
E-mail address: praveen.md@kletech.ac.in (Praveen M Dhulavvagol).

1877-0509 © 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the Seventh International Conference on Recent Trends in Image Processing and Pattern Recognition.
10.1016/j.procs.2025.03.284

addressing complex visual recognition tasks, like categorizing yoga poses[1]. Nonetheless, deploying such sophisticated models on edge devices, constrained by limited resources, presents formidable challenges due to their substantial computational and memory requirements[2]. Quantization techniques have surfaced as pragmatic solutions, offering avenues to shrink model sizes and expedite inference speeds while upholding accuracy[3].

Among the plethora of CNN architectures, MobileNetV3Large [4] has captured attention for its commendable performance and efficiency, rendering it apt for mobile and embedded applications. This study delves into applying quantization techniques to MobileNetV3Large within the context of yoga pose classification. Its aims encompass implementing post-training quantization (PTQ) and quantization-aware training (QAT) to augment model efficiency, juxtaposing the performance of PTQ and QAT quantized models, and deploying the most suitable quantized model on edge devices for practical deployment.

Yoga, deeply rooted in ancient Indian traditions, has garnered global acclaim for its multifaceted benefits encompassing physical, mental, and spiritual well-being[5]. Precise classification of yoga poses holds promise for furnishing personalized feedback and guidance to practitioners, enriching their learning journey, and averting potential injuries[6]. However, deploying yoga pose classification models on edge devices, such as smartphones or smart mirrors, poses intricacies concerning resource utilization and power efficiency. By leveraging quantization techniques on MobileNetV3Large, this research aspires to craft an efficient and precise yoga pose classification system, fostering heightened accessibility and interactivity in yoga practice worldwide.

In this work, we focus on enhancing the efficiency of the MobileNetV3Large model through the implementation of post-training quantization (PTQ) and quantization-aware training (QAT) techniques.

The main contributions of this work are as follows:

1. We implemented both post-training quantization (PTQ) and quantization-aware training (QAT) techniques to enhance the efficiency of the MobileNetV3 Large model for edge deployment.
2. A comprehensive comparison between the PTQ quantized model and the QAT quantized model was conducted, analyzing their performance.
3. The suitable quantized model was successfully deployed on an edge device, demonstrating its practical viability and efficiency in real-world applications.

The rest of this paper is organized as follows. In Section 2, we discuss the related work and background information relevant to our study. Section 3 presents our methodology and the proposed approach in detail. The results and discussion are covered in Section 4. Finally, we conclude the paper and discuss potential future work in Section 5.

2. Related Work

In their study, Rokh et al. [7] provide a comprehensive overview of model quantization techniques tailored for deep neural networks (DNNs) in image classification tasks. They delve into various aspects, including fundamental principles, advanced methodologies, and cutting-edge approaches, addressing the challenges of deploying DNNs on resource-constrained devices and proposing quantization as a promising strategy for compression and acceleration. The authors elucidate fundamental concepts such as quantization types, applicable network components, and timing considerations, followed by an exploration of training techniques for quantized neural networks, including the use of the straight-through estimator (STE) and the impact of parameters like learning rate, network architecture, and regularization techniques. Additionally, the study scrutinizes operations within quantized DNNs, focusing on multiply-accumulate (MAC) operations and layer sensitivity to quantization, while introducing mixed-precision quantization. Comprehensive evaluation metrics and comparisons of state-of-the-art methods across popular benchmark datasets are provided, leading to conclusions that summarize key findings and highlight challenges and future research directions, emphasizing the necessity for further investigation into quantizing weights and activations within deeper networks while maintaining high accuracy standards.

In their paper, Krishnamoorth [8] delivers an extensive examination of techniques aimed at quantizing deep convolutional neural networks to enable efficient inference with integer weights and activations. The authors meticulously outline various quantizer designs, encompassing uniform affine, uniform symmetric, and stochastic quantizers, while elucidating their implications on inference performance and accuracy. Through rigorous evaluation, they investigate the efficacy of post-training quantization and quantization-aware training approaches across diverse CNN architectures, spanning MobileNet, NASNet, Inception, and ResNet, revealing that 8-bit quantization can yield accuracies within a mere 2% deviation from floating-point networks. Additionally, the study delves into the ramifications of different quantization granularities, whether per-layer or per-channel, and addresses the treatment of batch normalization during quantization. Moreover, the authors dispense valuable insights into optimal training methodologies, recommend model architectures conducive to quantization, and furnish runtime measurements on CPUs and DSPs, showcasing notable speed enhancements and reductions in model size. Finally, they proffer recommendations concerning neural network accelerators to fully harness the advantages conferred by quantized networks.

In their research documented [9], Patel et al. delve into the quantization of MobileNet models designed for image classification tasks, particularly emphasizing the classification of retinal fundus images based on the severity of diabetic retinopathy. They emphasize the critical necessity for streamlined and efficient architectures capable of delivering robust performance on resource-constrained devices like mobile phones. Leveraging quantization-aware training (QAT) methodologies, they facilitate the quantization of pre-trained MobileNetv1 and MobileNetv2 models, thereby achieving reductions in both model size and computational overhead by transitioning to integer computations from floating-point operations. Additionally, the authors present an exhaustive overview of the evolutionary progression of deep learning architectures, tracing the development from early shallow models such as LeNet to more complex counterparts like AlexNet, VGG16, GoogLeNet, and ResNet152. They also delve into the emergence of compact and resource-efficient architectures like SqueezeNet, MobileNetv1, and MobileNetv2, specifically tailored for edge devices characterized by limited computational resources.

In their work detailed in [10], Jacob et al. present a novel quantization strategy tailored for efficient integer-arithmetic-only inference within deep convolutional neural networks (CNNs). Their proposed scheme involves quantizing both weights and activations as 8-bit integers, with only a limited number of parameters, specifically bias vectors, represented as 32-bit integers. The authors offer a comprehensive framework for quantized inference, designed for seamless integration with integer-arithmetic-only hardware, exemplified by the Qualcomm Hexagon architecture. Furthermore, they outline an optimized implementation on ARM NEON, ensuring both efficiency and accuracy. Additionally, they introduce a co-designed quantized training framework aimed at mitigating accuracy loss attributed to quantization. Demonstrating the versatility of their approach, the authors apply it to efficient classification and detection systems built upon MobileNet architectures. Benchmark evaluations conducted on prominent ARM CPUs reveal substantial enhancements in the latency-accuracy tradeoff across various tasks, including ImageNet classification, COCO object detection, and face detection.

3. Proposed Work

The methodology adopted in this study follows a structured workflow illustrated in Figure 1.It illustrates the process of training, quantizing, and deploying machine learning models on edge devices. Initially, raw input data undergoes preprocessing to produce a lightweight trained model. This model is then subjected to two parallel quantization techniques: post-training quantization (PTQ) and quantization-aware training (QAT). In PTQ, the trained model is directly quantized to produce the PTQ quantized model (QM), which is then converted into a TensorFlow Lite (TF Lite) model. In QAT, the trained model is first quantized and then retrained using additional training data to produce the QAT quantized model (QM), which is similarly converted into a TF Lite model. Both TF Lite models are subsequently evaluated and compared for performance. The best-performing model is then deployed on an edge device, demonstrating the workflow from data preprocessing to model deployment in edge computing scenarios. Detailed explanations of each stage are provided in the subsequent sections.

3.1. Data Preprocessing

The data used in this study comes from Kaggle and contains 3,700 images divided into 43 different yoga poses. To prepare the training data, images were subjected to a preliminary step that included resizing to a uniform 224x224 pixel resolution and normalizing to be compatible with the methods required in MobileNetV3Large models. After this, the dataset is divided into training and reference sets using a hierarchical method to manage parallel classes.

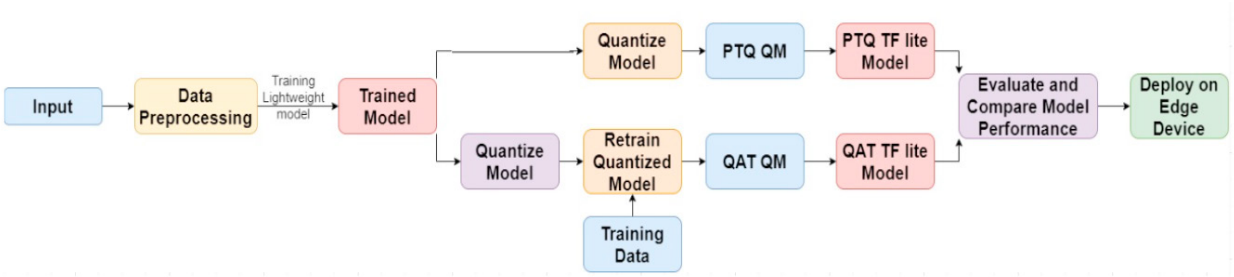

Fig. 1: Proposed Workflow

3.2. Model Training

The MobileNetV3Large model was first trained on the ImageNet dataset and formed the basis for classifying yoga postures. Transfer learning is done by modifying a good data model before doing yoga poses. The model was trained using the Adam optimizer with a learning rate of 0.0001 and a cluster size of 32. Model performance was evaluated using a categorical cross-entropy loss function as the optimization objective and accuracy as the key metric. Early starting and sampling control procedures are used to prevent overfitting and maintain the best possible model.

3.3. Quantization Techniques

We applied two quantization methods, Post-Training Quantization (PTQ) and Quantization Awareness Training (QAT), to the trained MobileNetV3Large model to reduce its size and improve the final data transfer result.

3.3.1. Post-Training Quantization (PTQ) - Dynamic Range Quantization

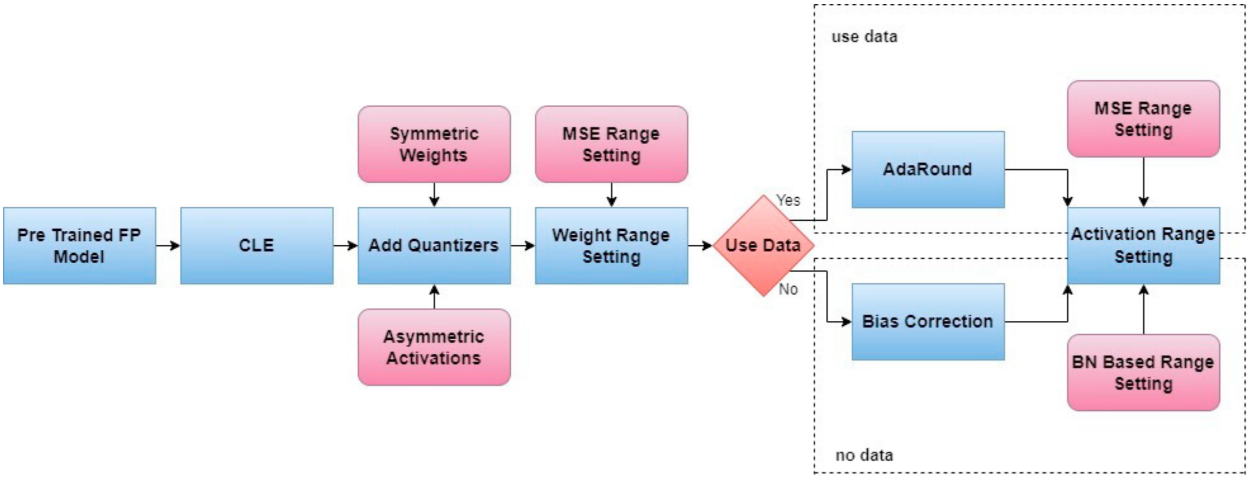

Fig. 2: PTQ Model Flow

Dynamic range quantization is a post-training quantization technique that quantizes the model's weights to 8-bit integers while keeping the activations in floating-point format. The quantization process involves mapping the floating-point values to a discrete set of integer values based on the minimum and maximum values of the weights in each layer. The quantized weight w_q is calculated as:

$$w_q = \text{round}\left(\frac{w - w_{\min}}{s}\right) + z$$

where w is the original floating-point weight, $w_{\min}$ is the minimum value of the weights in the layer, s is the scaling factor, and z is the zero-point. The scaling factor s is determined by:

$$s = \frac{w_{\max} - w_{\min}}{2^b - 1}$$

where $w_{\max}$ is the maximum value of the weights in the layer, and b is the number of bits used for quantization (in this case, 8).

Algorithm 1 Dynamic Range Quantization

```
for each layer in the model do
    $w_{\min}$ = min(layer.weights)
    $w_{\max}$ = max(layer.weights)
    $s = \frac{w_{\max} - w_{\min}}{2^8 - 1}$
    $z = -\text{round}\left(\frac{w_{\min}}{s}\right)$
    layer.weights_quantized = round$\left(\frac{\text{layer.weights} - w_{\min}}{s}\right) + z$
end for
```

3.3.2. *Quantization-Aware Training (QAT)*

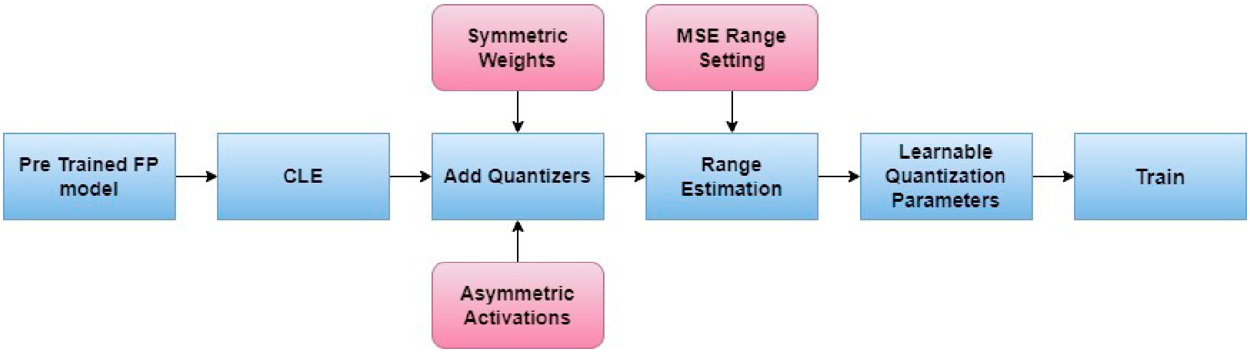

Fig. 3: QAT Model Flow

Quantization-aware learning involves retraining the model using simulated quantization tasks. This model can adapt its conflicts to a variety of representations. Sample weights during QAT activation are quantized to 8-bit integers using the following quantization function:

$$x_q = \text{round}\left(\frac{\text{clamp}(x, x_{\min}, x_{\max}) - x_{\min}}{s}\right)$$

where x is the original floating-point value (weight or activation), $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the tensor, s is the scaling factor, and $\text{clamp}(x, x_{\min}, x_{\max})$ ensures that the values are within the specified range. The scaling factor s is calculated as:

$$s = \frac{x_{\max} - x_{\min}}{2^b - 1}$$

During the forward pass, the quantized values are used for computation, and during the backward pass, the gradients are computed with respect to the original floating-point values using the straight-through estimator (STE).

Algorithm 2 Quantization-Aware Training

```
for each epoch do
    for each batch do
        /* Forward pass */
        for each layer in the model do
            $x_{\min}$ = min(layer.input)
            $x_{\max}$ = max(layer.input)
            $s = \frac{x_{\max}-x_{\min}}{2^8-1}$
            layer.input_quantized = $\text{round}\left(\frac{\text{clamp}(\text{layer.input},x_{\min},x_{\max})-x_{\min}}{s}\right)$
            layer.weights_quantized = $\text{round}\left(\frac{\text{layer.weights}-w_{\min}}{s}\right)+z$
            layer.output = layer.input_quantized × layer.weights_quantized
        end for
        /* Backward pass */
        compute gradients with respect to original floating-point values using STE
        update model parameters
    end for
end for
```

3.4. Model Evaluation and Comparison

Performance of the original training model, the quantitative PTQ model, and the quantitative QAT model. It is evaluated and compared using the validation process. Measures such as precision and sample size are measured. Determine the impact of quantity on quality standards and the quality of work. The results of the analysis determine which quantum model is best for edge deployment.

Table 1: Comparison of Model Variants

Model variants	Accuracy	Model size
Baseline	73.78%	12.5 MB
PTQ (Dynamic)	74.30%	3.43 MB
QAT	84.71%	11.2 MB

The Table 1 presents an evaluation of three different model variants based on their accuracy and model size. The baseline model achieves an accuracy of 73.78% with a model size of 12.5 MB. The post-training quantization (PTQ) with dynamic quantization method shows a slight decrease in accuracy to 74.30%, but significantly reduces the model size to 3.43 MB, demonstrating its efficiency in reducing storage requirements. The quantization-aware training (QAT) model achieves an accuracy of 84.71%, slightly lower than the baseline but with a reduced model size of 11.2 MB. This comparison highlights the trade-offs between model accuracy and size, showing that while PTQ offers the greatest size reduction, QAT provides a balanced approach with minimal accuracy loss and a moderate reduction in model size.

By adhering to this methodology, the study aims to provide a comprehensive understanding of the application of quantization techniques to the MobileNetV3Large model for yoga pose classification and their impact on model performance and efficiency when deployed on edge devices.

4. Results and Discussion

The performance of the baseline model, post-training quantization (PTQ) model, and quantization-aware training (QAT) model was evaluated and compared in terms of validation accuracy and model size. Figure 4 illustrates the validation accuracy of the baseline model and the quantization-aware model over the course of 100 epochs. The baseline model achieved a validation accuracy of 26.6% at 20 epochs, which steadily increased to 50.5% at 40 epochs, 61% at 60 epochs, 70% at 80 epochs, and finally reached 74.83% at the end of 100 epochs. In contrast, the quantization-aware model demonstrated superior performance, achieving a validation accuracy of 75% at 20 epochs, 80% at 40 epochs, 81% at 60 epochs, 84% at 80 epochs, and ultimately reaching an impressive 85.2% at the end of 100 epochs.

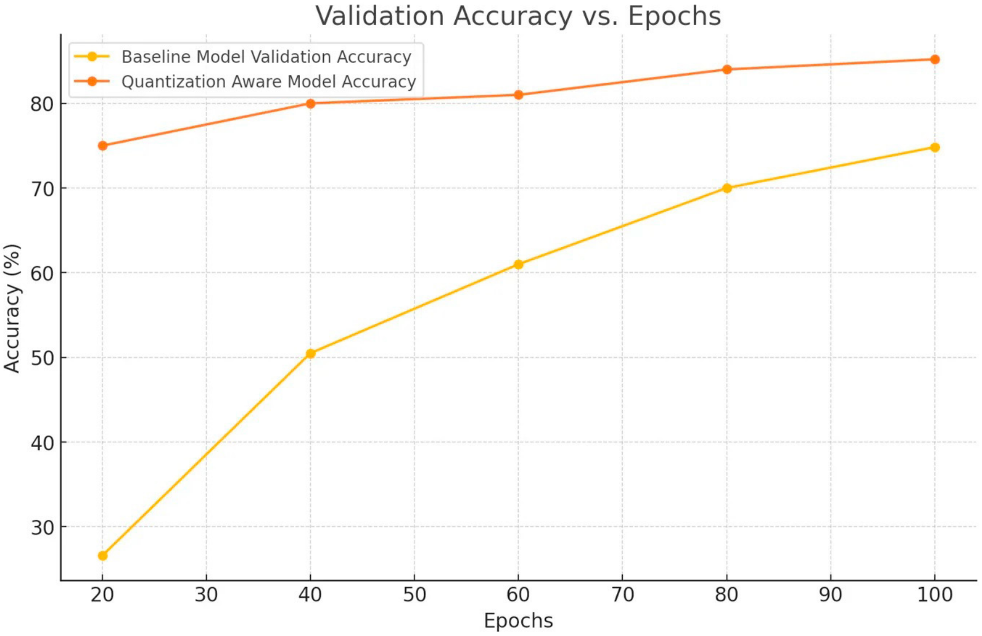

Fig. 4: Validation Accuracy vs Epochs

Figure 5 presents a comprehensive comparison of the three models' performance in terms of accuracy and model size. The baseline model achieved an accuracy of 73.78% with a model size of 12.5 MB. The post-training quantization model, which involved dynamic range quantization, exhibited an improved accuracy of 74.30% while significantly reducing the model size to 3.43 MB. The quantization-aware training model outperformed both the baseline and PTQ models, achieving a remarkable accuracy of 84.71% with a model size of 11.2 MB. These results demonstrate the effectiveness of quantization techniques, particularly quantization-aware training, in enhancing model performance while maintaining a relatively compact model size.

The results of this study demonstrate the effectiveness of applying quantization techniques to the MobileNetV3Large model for yoga pose classification. The quantization-aware training (QAT) model outperformed both the baseline model and the post-training quantization (PTQ) model in terms of validation accuracy, achieving an impressive 84.71% accuracy compared to 73.78% and 74.30%, respectively. This finding aligns with previous research that highlights the benefits of quantization-aware training in preserving model performance while reducing model size [3][8]. Although the PTQ model achieved a slightly higher accuracy than the baseline model (74.30% vs. 73.78%), its main advantage lies in the significant reduction of model size. The PTQ model, which employs dynamic range quantization, reduces the model size from 12.5 MB to just 3.43 MB, a compression of nearly 73%. This substantial reduction in model size is crucial for deploying the model on resource-constrained edge devices, where memory and storage are limited [11]. The PTQ model's ability to maintain comparable accuracy while drastically reducing the model size makes it an attractive option for real-world applications.

It is important to acknowledge the limitations of this study. The quantization techniques applied focused primarily on reducing the model size and improving inference efficiency; however, aspects such as latency and power consumption were not explicitly addressed. Future work could explore additional optimization techniques, such as pruning and model distillation, to further improve the model's performance and efficiency on edge devices [12]. Despite these limitations, the findings have significant implications for deploying yoga pose classification models on edge devices. The quantization techniques, particularly QAT, demonstrate the potential to develop accurate and efficient models that can be integrated into smartphones, smart mirrors, or other resource-constrained devices. This opens up opportunities for

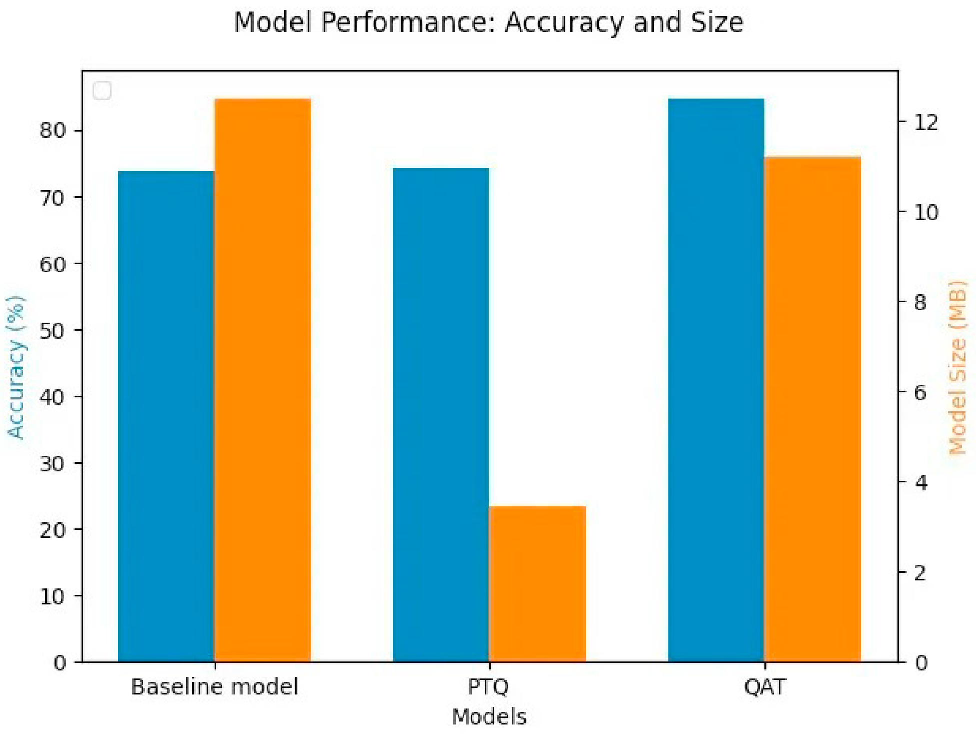

Fig. 5: Model Performance: Accuracy and Size

real-time feedback and guidance systems that can enhance the learning experience of yoga practitioners and promote proper pose execution [6].

5. Conclusions

This study investigated the application of quantization techniques to the MobileNetV3Large model for yoga pose classification, with the aim of improving model efficiency and enabling deployment on edge devices. The quantization-aware training (QAT) model achieved a remarkable accuracy of 84.71%, surpassing both the baseline model (73.78%) and the post-training quantization (PTQ) model (74.30%). The PTQ model, on the other hand, significantly reduced the model size from 12.5 MB to 3.43 MB, facilitating deployment on resource-constrained devices. These findings demonstrate the effectiveness of quantization techniques in optimizing the MobileNetV3Large model for yoga pose classification, paving the way for real-time feedback and guidance systems on edge devices. Future research should focus on exploring additional optimization techniques and validating the models' performance in real-world scenarios. Furthermore, investigating the integration of these optimized models into user-friendly applications and assessing their impact on yoga practitioners' learning experiences could provide valuable insights for the development of effective yoga pose classification systems.

References

[1] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1653-1660, doi: 10.1109/CVPR.2014.214.

[2] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," arXiv preprint arXiv:1510.00149, 2015.

[3] B. Jacob et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 2704-2713, doi: 10.1109/CVPR.2018.00286.

[4] A. Howard et al., ”Searching for MobileNetV3,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 1314-1324, doi: 10.1109/ICCV.2019.00140.

[5] A. Ross and S. Thomas, ”The health benefits of yoga and exercise: a review of comparison studies,” The journal of alternative and complementary medicine, vol. 16, no. 1, pp. 3-12, 2010.

[6] W. Gao, Y. Zhang, Q. Li, and L. Yu, ”Convolutional neural networks for yoga posture recognition and correction,” IEEE Access, vol. 7, pp. 138487-138498, 2019.

[7] abak Rokh, Ali Azarpeyvand, Alireza Khanteymoori, ”A Comprehensive Survey on Model Quantization for Deep Neural Networks in Image Classification”, arXiv preprint arXiv:2205.07877, 2023.

[8] Raghuraman Krishnamoorthi, ”Quantizing deep convolutional networks for efficient inference: A whitepaper”, arXiv preprint arXiv:1806.08342, 2018.

[9] R. Patel and A. Chaware, ”Quantizing MobileNet Models for Classification Problem,” 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2021, pp. 348-351.

[10] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, Dmitry Kalenichenko, ”Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference”, arXiv preprint arXiv:1712.05877,2017

[11] Yu Cheng, Duo Wang, Pan Zhou, Tao Zhang, ”A Survey of Model Compression and Acceleration for Deep Neural Networks”, arXiv preprint arXiv:1710.09282, 2017

[12] Choudhary, T., Mishra, V., Goswami, A. et al. A comprehensive survey on model compression and acceleration. Artif Intell Rev 53, 5113–5155 (2020). https://doi.org/10.1007/s10462-020-09816-7

[13] Praveen M Dhulavvagol, Akhilesh Gadagkar, Ateeth KJ, Gururaj Hegade, Ritik Poonia, S G Totad, Lossless Text Compression Using Recurrent Neural Networks, Procedia Computer Science, Volume 235, 2024, Pages 3340-3349, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2024.04.315.

[14] Praveen M. Dhulavvagol, Pritam S, Vijayalaxmi A, Raghav S, Sneha B, BEE Friendly Flora: Comparative Analysis of Plant Preference among Native and Non-Native Bee Species using EDA and Machine Learning Model, Procedia Computer Science, Volume 233, 2024, Pages 841-850, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2024.03.273.

[15] N. C. Kundur, B. C. Anil, P. M. Dhulavvagol, R. Ganiger, and B. Ramadoss, “Pneumonia Detection in Chest X-Rays using Transfer Learning and TPUs”, Eng. Technol. Appl. Sci. Res., vol. 13, no. 5, pp. 11878–11883, Oct. 2023.

[16] Dhulavvagol, P.M., Totad, S.G., Shirodkar, A., Hiremath, A., Bansode, A., Divya, J. (2022). Performance Analysis of Classification Algorithm Using Stacking and Ensemble Techniques. In: Jacob, I.J., Kolandapalayam Shanmugam, S., Bestak, R. (eds) Expert Clouds and Applications. Lecture Notes in Networks and Systems, vol 444. Springer, Singapore. https://doi.org/10.1007/978-981-19-2500-9-46