2024 Second International Conference on Network, Multimedia and Information Technology (NMITCON)

Leveraging Pre-trained ResNet-18 with Transfer Learning for Yoga Posture Classification

1st Aruna M*
Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai, India.
arunam@srmist.edu.in

2nd Gruhit Kaneriya
Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai, India.
gk9009@srmist.edu.in

3rd Prashuk Jain
Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai, India.
pj2566@srmist.edu.in

2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON) | 979-8-3503-7289-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/NMITCON62075.2024.10699235

Abstract— **This research paper explores the application of pre-trained ResNet-18 with transfer learning for the classification of yoga postures. The study utilizes a dataset comprising images of various yoga poses taken from Kaggle. Through fine-tuning, the pre-trained model achieved impressive training accuracy of 98%. Furthermore, on unseen data, the model maintained accuracy of 94.93%. Advantages and disadvantages of the methodology are discussed, along with experimental setup details and metrics evaluation. The findings underscore the efficacy of leveraging pre-trained models and transfer learning techniques for complex image classification tasks in specialized domains like yoga postures. However, a notable limitation arises in the difficulty of removing noise during model implementation, leading to a decrease in confidence scores. For future work, the project aims to address this limitation by focusing on noise removal techniques while maintaining accuracy and confidence scores, thus enhancing the robustness of the classification model.**

Keywords— Pre-trained ResNet-18, Transfer learning, Yoga postures classification, Fine-tuning, Experimental setup, Image classification

I. Introduction

Given the diversity of yoga positions and their variations among various traditions, it is a difficult task to recognize and classify them from photos. Conventional machine learning techniques frequently necessitate extensive human feature engineering, which can be labour-intensive and may not be successful in accurately capturing complex pose features. Convolutional neural networks (CNNs), in particular, are deep learning approaches that have shown impressive performance in image classification tasks in recent years.

The methodology adopted in this study involves utilizing a pre-trained ResNet-18 model, a variant of the popular ResNet architecture, which has demonstrated strong performance in various image classification tasks. Transfer learning is employed to fine-tune the pre-trained model on a specialized dataset of yoga postures. With comparatively fewer annotated instances, this method enables the model to apply knowledge from a large-scale dataset (such as ImageNet) to the target domain.

As a member of the ResNet family, ResNet-18 is a convolutional neural network architecture that was first presented by Kaiming He et al. Its eighteen layers include residual blocks with skip connections, which facilitates the more efficient propagation of gradients during training. The architecture comprises initial convolutional and pooling layers followed by four sets of residual blocks. Each residual block consists of two convolutional layers with ReLU activation functions and a skip connection, facilitating the learning of residual functions. Max-pooling layers reduce spatial dimensions, and the network concludes with average pooling and fully connected layers for output predictions. In PyTorch, ResNet-18 is readily accessible through torchvision.models, offering pre-trained weights on the ImageNet dataset, making it applicable for image classification, feature extraction, and transfer learning tasks.

Transfer learning repurposes pre-trained models from one task to another, leveraging knowledge from a source domain to adapt to a related target domain with limited data. Usually used in deep learning, it optimizes the model parameters on the fresh dataset, resulting in faster convergence and better performance, especially in situations where data is scarce. Pre-trained models are widely used in speech recognition, natural language processing, and computer vision as powerful feature extractors or initializers for task-specific training. Moreover, transfer learning helps mitigate challenges like overfitting and improves model generalization, rendering it a popular strategy in modern machine learning workflows.

Paper is structured into seven sections: 1. Introduction, providing an overview and rationale; 2. Related work, reviewing prior studies; 3. Materials and methods, detailing data preprocessing, analysis, and experimental setup; 4. Results and discussion, presenting outcomes and further analysis; 5. Conclusions, summarizing key findings; and 6. References, listing cited sources.

II. Literature Review

In recent research on yoga pose analysis and classification using deep learning, two noteworthy studies are highlighted. The first study [[1]] focuses on pose estimation and feedback generation, achieving notable accuracies with SVM (93.19%), CNN (98.58%), and CNN combined with LSTM (99.38%). Despite benefits like flexibility, challenges such as overfitting and computational complexity were encountered. Conversely, the second study [2] explores pose classification using CNN and MediaPipe-inspired methods, reporting promising results with models like VGG16 and InceptionV3. The YogaConvo2d model achieved a validation accuracy of 99.62%. Challenges related to pose complexity and external factors were noted, despite advantages like improved accuracy and reduced latency through skeletonization.

Two recent studies in yoga pose detection and classification employing deep learning methodologies is summarized. In the first study [3] various methods including SVM, CNN, and CNN + LSTM were employed, achieving a validation precision of 99.87% and a test accuracy of 99.38% using the CNN + LSTM method. The dataset used was

979-8-3503-7289-2/24/$31.00 ©2024 IEEE

created by Yadav et al., titled "Real-time yoga recognition using deep learning". While efficient in extracting yoga poses, reliance on the accuracy of Open Pose library for joint detection was noted as a limitation. In the second study [4] titled "YoNet, A Neural Network for Yoga Pose Classification", a CNN-based approach was employed. Using the Yoga-82 dataset, the methodology achieved efficient feature extraction with depthwise separable convolution, yet its performance was contingent upon OpenPose, marking a similar limitation.

In recent research on yoga pose analysis and classification using deep learning, two noteworthy studies are highlighted. Both studies propose deep learning models integrating CNN for yoga pose identification alongside human joints localization and error identification processes. The first study [5] presents a system achieving a classification accuracy of 95%, offering feedback for posture improvement or correction based on user pose data. This approach benefits from its holistic view of pose identification and feedback provision. However, challenges may arise in accurately localizing human joints, potentially leading to errors in feedback generation. Similarly, the second study [6] also achieves a notable classification accuracy of 95%, emphasizing the importance of feedback for posture refinement. This approach offers holistic pose analysis, challenges related to accurate joint localization persists, which could impact feedback accuracy.

In recent advancements in machine learning applications, two notable approaches stand out. Firstly [7], a method introduces an attention model with LSTM for action recognition, demonstrating improved performance over fully connected LSTM models. Achieving an accuracy of 90.05%, this approach competes well with existing methods like Spatio-temporal features and Binary CNN-Flow. While advantageous for enhancing recognition accuracy, complexities in training and interpreting LSTM architectures may pose challenges. Secondly [8], leveraging Google's MediaPipe open-source technology allows for the development of versatile and cross-platform Machine Learning pipelines, enabling multimodal applications. Integrating BlazePose within MediaPipe Pose facilitates high-fidelity body pose tracking, yielding detailed 3D landmark inference and background segmentation. With an impressive accuracy of 98%, this approach ensures robust performance. However, integrating and fine-tuning complex models like BlazePose may demand significant computational resources and expertise, presenting obstacles for developers with limited resources.

III. Materials And Methods

A. Abbreviations and Acronyms

First, the images were resized to a fixed size of 224x224 pixels. Then, Converted the images into PyTorch tensors using transforms. ToTensor(), enabling compatibility with PyTorch models. Following this, it normalized the pixel values of the images using mean and standard deviation values calculated from the ImageNet [9] dataset. Specifically, it normalized with mean values [0.485, 0.456, 0.406] and standard deviation values [0.229, 0.224, 0.225]. Before the input data is supplied into the neural network model for training, these preparation processes make sure it is properly prepared and standardized. This promotes more efficient learning and convergence during the training process.

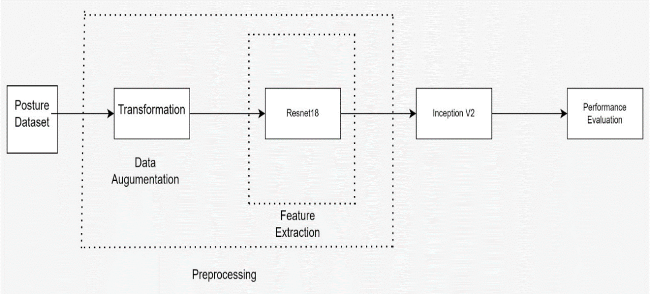


Fig. 1. Model Architecture of Yoga Posture Evaluation.

B. Model Used

In this paper, Transfer learning was employed atop a pre-trained ResNet-18 model for the classification of yoga postures. Initially, the parameters of the pre-trained layers were frozen to preserve their learned features, and the output layer was replaced with a new fully connected layer appropriate for the task with 47 [10] classes. The model was optimised on a specific dataset of photographs of people in yoga poses by only changing the parameters of the new output layer while retaining the features collected from the original ImageNet dataset. Transfer learning was used to adjust a cutting-edge convolutional neural network to a specific domain, resulting in an effective and exact classification of yoga poses with little processing resources. Fig. 1. shows the Model Architecture of the proposed model.

C. Experiment and Discussion

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

1. Experimental Setup

The experimental setup includes details of hardware, software, dataset, and metrics evaluation. The study was conducted on a system equipped with an 11th Gen Intel Core i7 processor, 16GB RAM, and Nvidia GeForce MX450 GPU. The dataset was sourced from Kaggle, consisting of 47 yoga posture classes with 80% training and 20% testing split. Metrics [11] evaluation involved accuracy, precision, recall, and F1 score calculations.

2. Dataset Description

The dataset used as shown in Fig. 2 includes 47 classes of yoga postures, each with photos and metadata that lists the names of the positions in both Sanskrit and English. The dataset provides a diverse collection of approximately 3,000 images, offering comprehensive coverage [12] of both traditional and modern yoga practices.

3. Metrics Evaluation

Cross-Entropy Loss Function: This quantifies the difference between the actual class labels and the expected class probabilities. The cross-entropy loss formula (1) is:

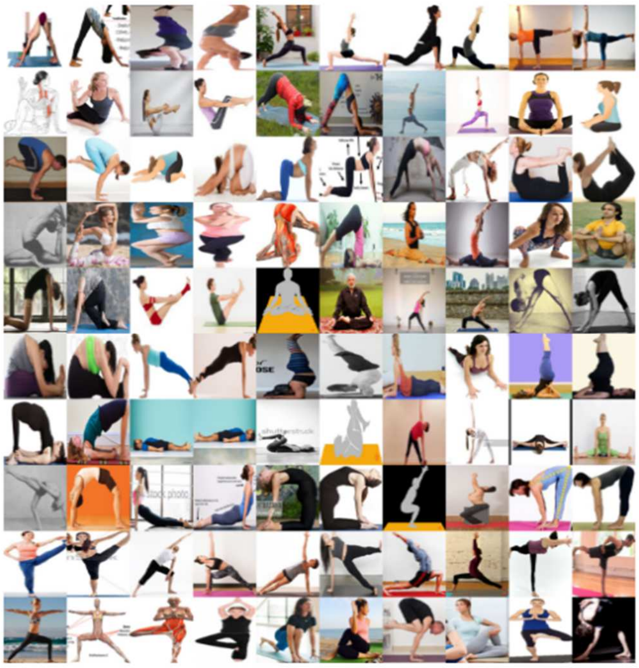


Fig. 2. Yoga Pose 47 classes Dataset.

$$H\left(p,q\right) = -\sum_{i=1}^{N} p_i \log\left(q_i\right) \qquad (1)$$

where, p_i is the true probability for class i (usually 1 for the correct class and 0 for all others in a one-hot vector), q_i is the predicted probability for class i and N is the total number of classes [13].

Accuracy: The percentage of correctly identified samples relative to the total number of samples is the measure of accuracy. The accuracy formula (2) is:

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \text{ x } 100\% \qquad (2)$$

Equation (2) shows the accuracy formulae used for the evaluation of the model.

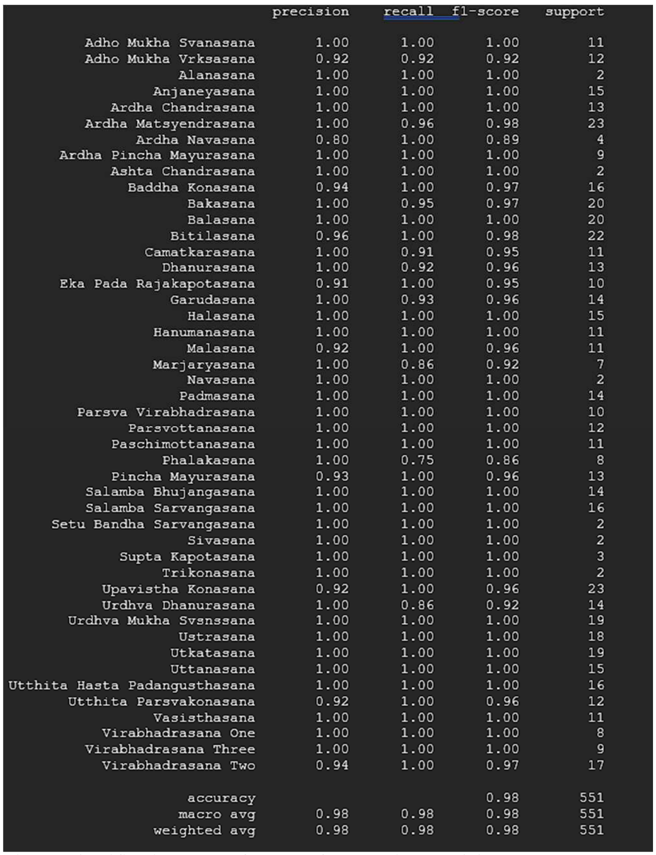
Validation Loss: This metric evaluates the model's performance on unseen data during validation. It provides insights into the generalization ability of the model and helps prevent overfitting [14]. Validation loss is calculated using the same formula as cross-entropy loss.

The classification model trained using the pre-trained ResNet-18 with transfer learning achieved remarkable performance. During training, the model attained an accuracy of 98%, demonstrating its ability to capture intricate features within the yoga posture dataset. Subsequently, on unseen data, the model maintained a high accuracy of 94.93%, indicating robustness and generalization capabilities [15].

IV. Results And Discussion

A. Classification report for Transfer Learning Model

Applying transfer learning to the ResNet model as shown in Fig. 3 effectively addressed the challenges encountered in accurately detecting Ashta Chandrasana, Marjayasana, and Bitilasana yoga poses. Leveraging knowledge from a pre-trained model significantly improved classification accuracy for these poses, mitigating previous errors. The findings highlight the effectiveness of transfer learning in enhancing model [16] performance and underscore its potential to overcome classification challenges in yoga pose recognition tasks. This demonstrates the value of leveraging pre-trained models to refine classification algorithms and optimize performance in specialized domains. As Shown in Fig. 4. the Confusion Matrix provides a detailed analysis of proposed model for different classes and can be used to revalidate the accuracy of the model.



```
                             precision    recall  f1-score   support

       Adho Mukha Svanasana       1.00      1.00      1.00        11
       Adho Mukha Vrksasana       0.92      0.92      0.92        12
                  Alanasana       1.00      1.00      1.00         2
               Anjaneyasana       1.00      1.00      1.00        15
          Ardha Chandrasana       1.00      1.00      1.00        13
      Ardha Matsyendrasana       1.00      0.96      0.98        23
             Ardha Navasana       0.80      1.00      0.89         4
   Ardha Pincha Mayurasana       1.00      1.00      1.00         9
          Ashta Chandrasana       1.00      1.00      1.00         2
            Baddha Konasana       0.94      1.00      0.97        16
                   Bakasana       1.00      0.95      0.97        20
                   Balasana       1.00      1.00      1.00        20
                 Bitilasana       0.96      1.00      0.98        22
              Camatkarasana       1.00      0.91      0.95        11
                Dhanurasana       1.00      0.92      0.96        13
    Eka Pada Rajakapotasana       0.91      1.00      0.95        10
                 Garudasana       1.00      0.93      0.96        14
                   Halasana       1.00      1.00      1.00        15
               Hanumanasana       1.00      1.00      1.00        11
                   Malasana       0.92      1.00      0.96        11
               Marjaryasana       1.00      0.86      0.92         7
                   Navasana       1.00      1.00      1.00         2
                  Padmasana       1.00      1.00      1.00        14
      Parsva Virabhadrasana       1.00      1.00      1.00        10
           Parsvottanasana       1.00      1.00      1.00        12
          Paschimottanasana       1.00      1.00      1.00        11
                Phalakasana       1.00      0.75      0.86         8
           Pincha Mayurasana       0.93      1.00      0.96        13
        Salamba Bhujangasana       1.00      1.00      1.00        14
        Salamba Sarvangasana       1.00      1.00      1.00        16
   Setu Bandha Sarvangasana       1.00      1.00      1.00         2
                   Sivasana       1.00      1.00      1.00         2
             Supta Kapotasana       1.00      1.00      1.00         3
                Trikonasana       1.00      1.00      1.00         2
           Upavistha Konasana       0.92      1.00      0.96        23
          Urdhva Dhanurasana       1.00      0.86      0.92        14
       Urdhva Mukha Svsnssana       1.00      1.00      1.00        19
                  Ustrasana       1.00      1.00      1.00        18
                  Utkatasana       1.00      1.00      1.00        19
                 Uttanasana       1.00      1.00      1.00        15
Utthita Hasta Padangusthasana       1.00      1.00      1.00        16
       Utthita Parsvakonasana       0.92      1.00      0.96        12
                Vasisthasana       1.00      1.00      1.00        11
           Virabhadrasana One       1.00      1.00      1.00         8
         Virabhadrasana Three       1.00      1.00      1.00         9
           Virabhadrasana Two       0.94      1.00      0.97        17

                   accuracy                           0.98       551
                  macro avg       0.98      0.98      0.98       551
               weighted avg       0.98      0.98      0.98       551
```

Fig. 3. Classification report for Transfer Learning Model.

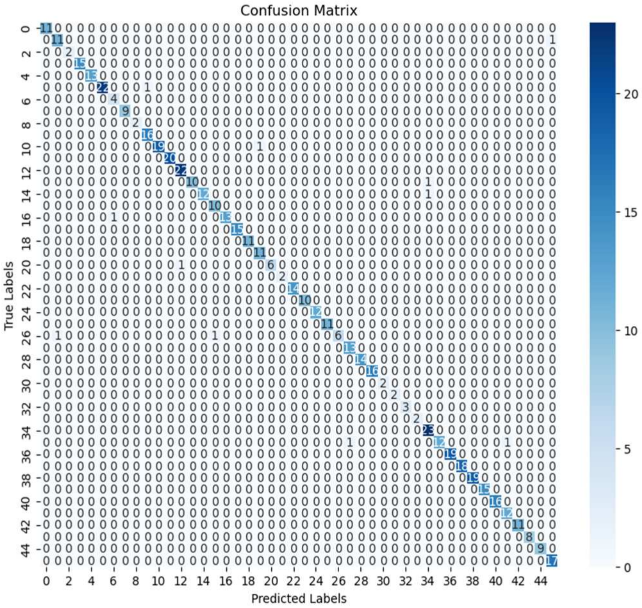


Fig. 4. Confusion matrix for transfer learning model.

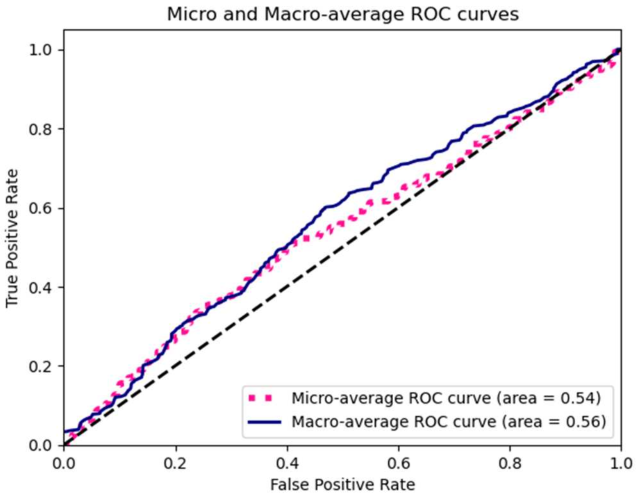


Fig. 5. Micro and Macro ROC curves.

The precision, recall, and F1-score evaluation metrics for various models in a classification job are shown in the Table 1. Lower performance is shown by the simple CNN model across various metrics, which may be related to overfitting or model complexity. The CNN Mobilev2 model shows improvement over the basic CNN, likely benefiting from its optimized architecture for mobile applications. ResNet [17], with its deeper architecture and skip connections, outperforms the CNN models, indicating its suitability for the task. But the ResNet + Transfer Learning model gets the best precision, recall, and F1-score, demonstrating how transfer learning works by using pre-trained models on large datasets to fine-tune and generalize to the particular classification task in an efficient manner, leading to better performance.

The ROC curves generated for the model as shown in Fig. 5 indicate that its performance in distinguishing between true positives and true negatives is marginally above random chance. Specifically, the Micro-average ROC curve achieves an area under the curve (AUC) of 0.54, while the Macro-average ROC curve shows an AUC of 0.56, suggesting a modest discriminatory capability. For optimal performance, an AUC closer to 1 is desired, indicating stronger predictive ability in class differentiation. These results imply potential room for improvement through further refinement of model parameters or incorporation of more discriminative features to enhance predictive accuracy and efficacy in real-world applications.

B. Limitations

When using the initial yoga pose detection model without segmentation, the prediction accuracy was high at 99.99%, showcasing the model's robust performance in recognizing yoga poses directly from images. However, integrating segmentation into the workflow resulted in a notable decrease in prediction accuracy to 88.03%. This decline suggests that while segmentation can aid in focusing the model's attention on relevant image regions, it may alter the input in a way that hampers the pose detection model's ability to accurately classify poses.

TABLE I. REPRESENTS DIFFERENT METRICS BASED ON WEIGHTED AVERAGE OF 4 DIFFERENT MODELS.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
CNN	0.54	0.57	0.54	0.54
CNN Mobilev2	0.68	0.67	0.67	0.66
ResNet	0.86	0.87	0.86	0.86
ResNet + TL	**0.98**	**0.98**	**0.98**	**0.98**

There is a decrease in accuracy when using segmentation likely occurred because the segmented image passed to the pose detection model may have lacked some critical visual context or details required for precise classification. In the second scenario, the DeepLabv3 model was used for segmentation, which effectively isolated regions of interest such as the person performing the yoga pose. However, this alteration of the input image, although intended to highlight relevant features, potentially removed or obscured subtle details necessary for the pose detection model to make an accurate prediction.

V. CONCLUSION

The study showcases the exceptional performance of the classification model trained with pre-trained ResNet-18 using transfer learning techniques. With an impressive training accuracy of 98%, the model effectively captured nuanced features inherent in the yoga posture dataset, underscoring its proficiency in learning complex patterns. Furthermore, on unseen data, the model exhibited remarkable robustness, maintaining a high accuracy of 94.93%. These results affirm the effectiveness of transfer learning in leveraging pre-trained models for domain-specific tasks, highlighting its potential in achieving superior classification performance with minimal computational resources. The findings not only validate the efficacy of the methodology but also underscore its applicability in real-world scenarios, particularly in healthcare analytics and computer vision applications.

VI. FUTURE DIRECTION

To address the reduction in accuracy caused by segmentation, future research should look into adaptive segmentation approaches that preserve crucial visual elements required for pose detection. Furthermore, fine-tuning the integration process between segmentation and posture detection models may improve the model's capacity to retain high accuracy while taking advantage of concentrated attention on key image regions. Using more complex pose estimation techniques or multimodal approaches may also help to reduce the impact of segmentation on classification accuracy. Furthermore, stronger noise reduction algorithms can be utilised, which will give a better adaptability to function in low illumination and lousy cameras while keeping or even enhancing accuracy, and then can be further turned into live posture detection so that it can be used in yoga apps.

REFERENCES

[1] Vivek Anand Thoutam,1Anugrah Srivastava,1Tapas Badal, Vipul Kumar Mishra, G. R. Sinha, Aditi Sakalle, Harshit Bhardwaj, and Manish Raj, “Lightweight Deep Learning Models for Resource Constrained Devices”, Volume 2022 | Article ID 4311350 | https://doi.org/10.1155/2022/4311350.

[2] Garg, S., Saxena, A. and Gupta, R., “Yoga pose classification: a CNN and MediaPipe inspired deep learning approach for real-world

application," J Ambient Intell Human Comput 14, 16551–16562 (2023). https://doi.org/10.1007/s12652-022-03910-0.

[3] Deepak Kumar and Anurag Sinha, "Yoga Pose Detection and Classification Using Deep Learning", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), Volume 6, Issue 6, Page Number: 160-184, 2020. doi : https://doi.org/10.32628/CSEIT206623.

[4] Ashraf, F.B., Islam, M.U., Kabir, M.R. et al., "YoNet: A Neural Network for Yoga Pose Classification", SN COMPUT. SCI. 4, 198 (2023). https://doi.org/10.1007/s42979-022-01618-8.

[5] Y. Shavit and R. Ferens, "Introduction to camera pose estimation with deep learning.", 2019, https://arxiv.org/abs/1907.05272. DOI: 10.48550/arXiv.1907.05272.

[6] Chaudhari, A., Dalvi, O., Ramade, O., and Ambawade, D., "Yog-Guru: Real-Time Yoga Pose Correction System Using Deep Learning Methods", In 2021 International Conference on Communication information and Computing Technology (ICCICT), pp. 1-6, 2021. DOI:10.1007/s11042-021-10687-5.

[7] K. Pothanaicker, "Human action recognition using CNN and LSTM-RNN with attention model", Intl Journal of Innovative Tech. and Exploring Eng, vol. 8, no. 8, 2019.

[8] H.-T. Chen, Y.-Z. He, and C.-C. Hsu, "Computer-assisted yoga training system", Multimedia Tools and Applications, vol. 77, no. 18, pp. 23969–23991, 2018.

[9] A. Guler, N. Kardaris, S. Chandra et al., "Human joint angle estimation and gesture recognition for assistive robotic vision", in Proceedings of the European Conference on Computer Vision, pp. 415–431, Springer, Amsterdam, The Netherlands, October 2016. DOI: 10.3390/s21175728.

[10] S. K. Yadav, A. Singh, A. Gupta, and J. L. Raheja, "Real-time Yoga recognition using deep learning", Neural Computing and Applications, vol. 31, no. 12, pp. 9349–9361, 2019. DOI:10.1155/2022/4311350.

[11] C. Hsieh, B. S. Wu, and C. C. Lee, "A distance computer vision assisted yoga learning system", Journal of Computers, vol. 6, no. 11, pp. 2382–2388, 2011.

[12] A. Toshev and C. Szegedy, Deeppose: human pose estimation via deep neural networks, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1653–1660, Columbus, OH, USA, August 2014.

[13] Nivas Maddukuri and Srinivasa Rao Ummity, "Yoga Pose prediction using Transfer Learning Based Neural Networks", Research Square, 2023. DOI: https://doi.org/10.21203/rs.3.rs-2807080/v1

[14] Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., and Torr, P. H., "Randomized trees for human pose detection", In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2008.

[15] Nadeem, A., Jalal, A., and Kim, K., "Automatic human posture estimation for sport activity recognition with robust body parts detection and entropy markov model", Multimedia Tools and Applications, 80(14), 21465-21498, 2021.

[16] M. S. Abd Rahim, F. Yakub, M. Omar, R. Abd Ghani, I. Dhamanti and S. Sivakumar, "Prediction of Influenza A Cases in Tropical Climate Country using Deep Learning Model," 2023 IEEE 2nd National Biomedical Engineering Conference (NBEC), Melaka, Malaysia, 2023, pp. 188-193, doi: 10.1109/NBEC58134.2023.10352612.

[17] Soubraylu Sivakumar, S.S. Sridhar, Ratnavel Rajalakshmi, M. Pushpalatha, S. Shanmugan, G. Niranjana, "Intelligent and assisted medicine dispensing machine for elderly visual impaired people with deep neural network fingerprint authentication system". Internet of Things. 23, ISSN 2542-6605, (2023). Doi: 10.1016/j.iot.2023.100821