Computer Vision-Based Yoga Pose Recognition Using Hybrid Deep Learning Model

Hukam Chand Saini[1] Dr. Renu Bagoria[1] and Dr. Praveen Arora[2]

[1] Jagan Nath University, Jaipur, INDIA
[2] Jagan Institute of Management Studies (JIMS) Delhi-India
hukumchand.saini@jagannathuniversity.org

Abstract. Human action recognition is a critical aspect of computer vision research and has various practical applications. In this paper, we focus on a specific type of action recognition, yoga pose recognition, and propose a computer vision-based model using deep learning. Our proposed model is a hybrid of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) and is designed to aid individuals in their self-practice of yoga. Mediapipe pose estimation is used to extract body keypoint as a feature of yoga poses. The Convolutional Neural Network (CNN) layer is utilized for extracting features from the keypoints, and the Gated Recurrent Unit (GRU) layer follows it to understand the sequence of frames for making predictions. The model is trained on video dataset of yoga poses carried out by various individuals. Model performance is evaluated based on its ability to accurately recognize the poses. The integration of Mediapipe and the combination of CNN and GRU offers a unique approach to yoga pose recognition and provides new insights into the field of human action recognition.

Keywords: Yoga asana, Human action recognition, Computer vision, Human pose estmation, Deep learning

1 Introduction

The recognition of human action through computer vision has been a topic of great interest among researchers for several years due to its vast potential for applications [1]. The realm of recognition systems encompasses a diverse array of fields, ranging from robotics and human-computer interaction to gaming, video surveillance, biometric verification, sports and health monitoring [2]. Despite numerous advancements in recognition technology over the years, the use of such systems to detect yoga postures remains a relatively fresh and untapped area. In today's fast-paced and stressful world, taking care of one's mental and physical health has become increasingly important. Yoga, an ancient Indian practice, has been proven to play a crucial role in maintaining overall well-being [3]. However, it holds the potential to significantly impact public health and wellness through promoting this ancient Indian practice, and serves as an important step forward for researchers in the advancement of human action recognition [4]. Yoga Asana is also an action as involves a sequence of intricate movements starting from a neutral position, proceeding through intermediate steps, and

R. Murugan et al. (eds.), *Proceedings of the International Conference on Signal Processing and Computer Vision (SIPCOV 2023)*, Advances in Engineering Research 239,
https://doi.org/10.2991/978-94-6463-529-4_17

finally reaching the final yoga pose. The practitioner then holds the pose for a few moments and return to the starting position. Earlier, most of work for yoga pose recognition worked with image datasets as considered only finished pose only [5], few works are done under human pose estimation. Human pose estimation provides a static representation of the human body, while human action recognition provides a dynamic representation of human behavior over time. While the two fields are closely related, human action recognition is a more complex problem that builds upon the foundation of human pose estimation.

Advancements in deep learning have led to increased research on yoga pose estimation in the field of human action recognition. The integration of deep learning has improved accuracy and efficiency, opening up new opportunities for growth and practical applications in health, wellness and athletic performance analysis.

The suggested approach involves the deployment of Mediapipe to pinpoint specific body landmarks. These 33 landmarks encompass crucial features such as the wrists, elbows, shoulders, knees, ankles, feet, hips, ears, nose, eyes, are identified and linked to form a skeletal representation. Designed hybrid model of Convolutional Neural Network (CNN) [6] and Gated Recurrent Unit (GRU) [7] uses the extracted keypoints to predict yoga poses, with CNN extracting spatial features and GRU processing temporal information.

The paper is structured into five sections. Section II reviews existing literature on yoga pose recognition. In Section III, the proposed methodology is explained. Section IV analyzes and discusses the results. Finally, Section V concludes the paper and outlines future work.

2 Literature review

In this section, the significant literature review of prior research will be explored and discussed for field of yoga pose analysis. Multiple solutions based on artificial intelligence have been presented in the literature. These solutions can be broadly classified into three categories: wearable device-based [8], Kinect-based and computer vision-based [9].Wearable device-based solutions use wearable devices such as smart watches or fitness trackers to track and analyze various parameters such as joint angles and limb movements.

The Kinect-based approach for yoga pose analysis has also been extensively researched. Yoga posture identification was introduced by Chen et al. [10] through a feature-based approach. They captured the user's body shape and created a body map using a Kinect depth sensor camera. The star skeleton technique was applied for fast skeletonization to describe the human posture. Their dataset contains images of twelve different yoga postures and obtained an accuracy of 99.33%. Another approach to yoga detection was put forward by Trejo et al. [11] using the Kinect camera and Adaboost classification for recognizing six asanas (accuracy >90%). However, using a depth-sensor camera is not as widespread and may not be easily available in households. In contrast, Gochoo et al. [12] developed a yoga recognition system that

leverages the Internet of Things (IoT) technology with the use of a deep CNN and three wireless sensor network (WSN) nodes equipped with infrared sensors. This system was capable of recognizing 26 different yoga poses, achieving an impressive average F1-score of 0.9854 with one WSN node and 0.9989 with three WSN nodes.

Computer vision is a rapidly expanding field and Human Pose Estimation (HPE) is one of its most promising aspects. and provide a more visual and intuitive representation of a user's movements compared to wearable device-based or Kinect-based solutions [13]. HPE focuses on identifying and assessing human body parts using technology that can detect key points in images and videos [14]. The skeleton-based model is commonly used due to its versatility, and accurate feature extraction is crucial for HPE accuracy [15]. Kothari [16] created a deep learning technique, specifically convolutional neural networks, for categorizing yoga poses in pictures. The dataset she used for her work consisted of 1000 images spread across 6 different yoga poses, and she achieved 85% accuracy. The scope of her work was limited to only six yoga poses. Lin et al.[17] developed a self-practice yoga system that tracks body posture during practice, utilizing the OpenPose[18] to identify twenty four body keypoints with 5.5 fps speed. Yamao and Kubota[19] developed a human pose recognition system using the PoseNet [20] model. The PoseNet model was capable of detecting 18 body joints. The system was tested and the accuracy of pose recognition was evaluated by measuring the movement of body parts. Jo and Kim put forward a model to estimate errors in a yoga posture by leveraging MoveNet [21]. Thunder has the ability to provide key point coordinates for 17 unique joints in the body. AlphaPose [22] can estimate the 17 (Alphapose-COCO) keypoints used for pose estimation of the Yoga performer [23].

Previous significant studies have utilized image datasets as only a limited number of publicly accessible yoga video datasets exist [5]. However, as a yoga asana involves a sequence of movements, an image dataset alone may not be suitable for training a model in a realistic scenario, as it does not account for intermediate steps. To address this issue, Yadav et al. [24] proposed a yoga recognition system capable of classifying 6 different asanas. The system utilized OpenPose to identify 18 keypoints on the human body, which were then fed into a hybrid deep neural network model. A video dataset of 6 yoga asanas was created, consisting of 88 total videos, resulting in training and validation accuracy of 99.34% and 99.41% respectively, and a testing accuracy of 99.04% frame-wise and 99.38% after polling 30 frames. The model was also tested in real-time with an accuracy of 98.92%. Jain et al. [25] proposed a 3DCNN-based yoga pose detection system, which was an altered version of the C3D architecture originally developed for recognizing human actions and scored 99.39 and 90.50 testing accuracy on test set and in real-time respectively.

3 Proposed Methodology

The methodology for the proposed hybrid model of CNN-GRU with Mediapipe pose estimation is outlined in this section. The section is divided into sub-sections.

3.1 Dataset preparation

A video dataset of yoga poses is essential for yoga action recognition. For this study, we have acquired a publicly available dataset [24] consisting of 6 yoga postures: Bhujangasana, Padmasana, Tadasana, Trikonasana, Shavasana, and Vrikshasana shown in Fig. 1. The dataset was created with ten males and five females total 15 participants using normal RGB webcam. All participant perform all 6- asanas in common indoor environment. This dataset contains 88 total no. of videos with single view recorded at 30 fps.

Fig. 1. Six yoga pose of collected video dataset

3.2 Body Keypoints detection and data generation

In this work, we have utilized Mediapipe [26] for body keypoint detection. When compared to other pose estimation methods such as OpenPose, PoseNet, AlphaPose, and MoveNet, Mediapipe offers the advantage of being easily integrated into our everyday platforms without the need for powerful hardware and consuming less computational resources.

The Mediapipe pose estimation extracts 33 keypoints, each representing a specific body part and joint location in terms of X, Y, and Z coordinates. The keypoints are depicted in Fig. 2. and Fig. 3 represents how Mediapipe generates keypoint for frame of videos. The Mediapipe library generates 3D coordinates with Z denoting depth, and We obtain the X and Y coordinates of the joint locations from each frame of the video and save them in a JSON file. After processing, the X and Y values are separated and

arranged in vectors X and Y respectively. The data is then split into training, validation, and test sets with a ratio of 80:10:10 for their respective use in the model.

Google Mediapipe is able to conduct pose estimation in all sorts of platforms like desktop, iso, android with a relevantly low latency when compared to other pose estimation methods like OpenPose, PoseNet, AlphaPose. It can be integrated into all the platforms that we use on a regular basis and doesn't need powerful hardware [26].

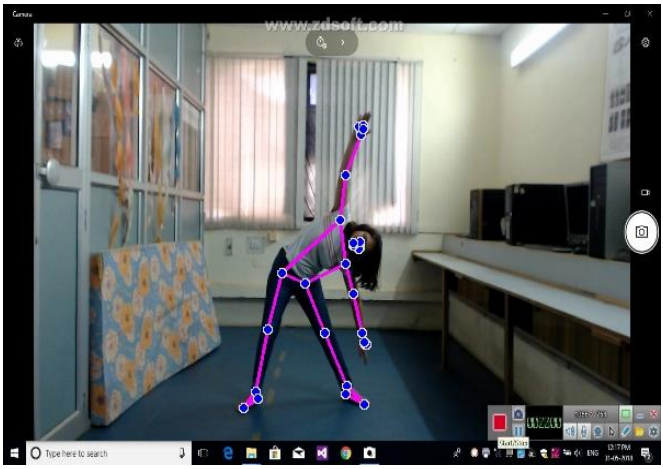
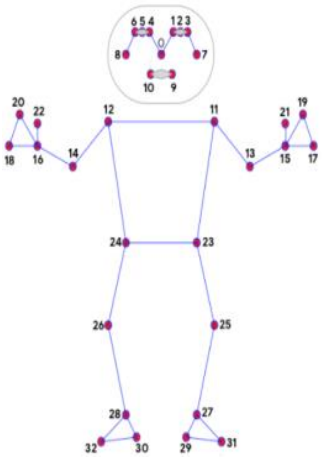


Fig. 2. Extraction of Keypoints of body parts using Mediapipe for Trikonasana video



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

Fig. 3. List and body location of 33 keypoints Pose

3.3 Model Design

A combination of CNN and GRU, known as a hybrid model, was created using the high-level neural network API Keras and powered by TensorFlow with python programming. The CNN was utilized for spatial feature extraction while the GRU, a type

of RNN, was employed to process the temporal information from the sequence of video frames.

The proposed model features three Conv1D layers serving as feature extractors, each with 32, 256, and 100 filters of size 3 and utilizing the 'relu' activation function. To improve training efficiency, Batch Normalization layers were added to normalize the activations. To prevent over-fitting and simplify the model's complexity, MaxPooling1D and Dropout layers were incorporated. The output from the feature extractor was then transformed into a 1D array through the use of a Flatten layer. A GRU layer with 50 hidden units was added to model the sequential data, followed by a Dense layer with 6 neurons and the 'softmax' activation function for asana classification. The complete model architecture is depicted in Fig. 4.

Layer (type)	Output Shape	Param #
time_distributed (TimeDistr ibuted)	(None, 30, 33, 32)	224
time_distributed_1 (TimeDis tributed)	(None, 30, 33, 32)	128
time_distributed_2 (TimeDis tributed)	(None, 30, 16, 32)	0
time_distributed_3 (TimeDis tributed)	(None, 30, 16, 32)	0
time_distributed_4 (TimeDis tributed)	(None, 30, 16, 256)	24832
batch_normalization_1 (Batc hNormalization)	(None, 30, 16, 256)	1024
time_distributed_5 (TimeDis tributed)	(None, 30, 8, 256)	0
time_distributed_6 (TimeDis tributed)	(None, 30, 8, 256)	0
time_distributed_7 (TimeDis tributed)	(None, 30, 8, 100)	76900
batch_normalization_2 (Batc hNormalization)	(None, 30, 8, 100)	400
time_distributed_8 (TimeDis tributed)	(None, 30, 8, 100)	0
gru (GRU)	(None, 30, 50)	127800
time_distributed_10 (TimeDi stributed)	(None, 30, 24)	1224

Total params: 232,532
Trainable params: 231,756
Non-trainable params: 776

Fig. 4. Model Architecture

3.4 Model Training

The model was trained using 50 epochs with a batch size of 16, utilizing the training data and being evaluated on the validation data. The training was performed on a Google Colab system that features a Python 3 Google Compute Engine backend with a GPU, 15.0 GB of RAM, and a System RAM of 12.7 GB. The system only utilized 3.2 GB and 4 GB of memory during the training process.The performance of the model was measured through the use of the categorical cross-entropy loss function and a

Softmax activation function in the fully connected layer. The model was compiled with a adam optimizer and a learning rate of 0.0001. In order to prevent overfitting, an EarlyStopping callback was employed to halt training if there was no improvement in validation accuracy for 15 epochs. A ModelCheckpoint callback was also utilized to preserve the model's weights each time the validation accuracy improved.

Fig.5 and Fig.6 illustrate the changes in training and validation accuracy and losses throughout the model training process. At each epoch, the accuracy and loss of the model are calculated for both the training and validation datasets. The figures reveal that the training accuracy rapidly improves during the first eight epochs and stays at a high level throughout the following epochs, with a minimal loss for both the training and validation sets. Additionally, it can be seen that the validation accuracy closely follows the training accuracy, suggesting that the model is well-generalized. As applied EarlyStopping, the training was terminated after 44 epochs, resulting in remarkable results with an average training time of approximately 12 seconds per epoch. The training accuracy reached 99.95% and validation accuracy was 99.79%, demonstrating excellent performance.

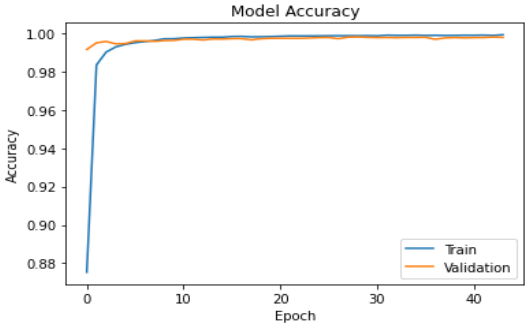


Fig. 5. Depicts the training and validation accuracy of model

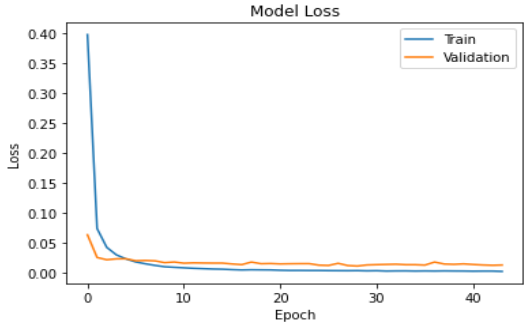


Fig. 6. Depicts the training and validation loss of model

3.5 Model Performance Evaluation

After getting a training and validation accuracy of 99.95% and 99.79%, respectively for designed hybrid model of CNN and GRU the model was tested for evolution on test dataset and in real time. The proposed model is capable of processing temporal data to provide predictions on sequences. This is achieved by considering yoga asanas as an action, including the intermediate steps involved in both forming and leaving an

asana. Polling for 30 frames was applied with window size of 25. The polling correctly identifies the intermediate steps as part of the yoga pose action. The most frequently occurring asana among the 30 frames is considered the predicted asana. In the end, the trained model showed a testing accuracy of 99.95% on the test set of the dataset with all correct prediction of all asana except 2 wrong prediction for vrikshasana among 854 test cases. The testing results using polling are depicted in the form of a confusion matrix and its normalized version in Fig. 7 and Fig. 8 respectively on test dataset.

The model is also tested in real-time on live camera streams captured by a 1080p HD Logitech webcam. A group of six individuals, consisting of three males and three females, performed all six asanas. The system was equipped with a 64-bit Windows 10 operating system, an Intel i5-8300H CPU, 8GB of RAM, and an Nvidia GeForce GTX 1050 GPU-8GB.

3.6 Model Performance Evaluation

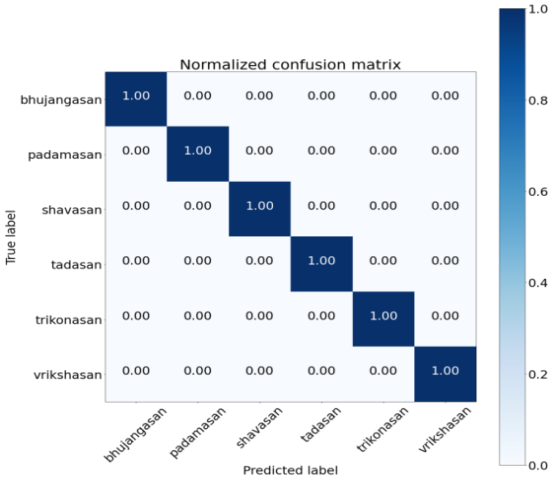


Fig. 7. Normalised Confusion matrix for testing accuracy

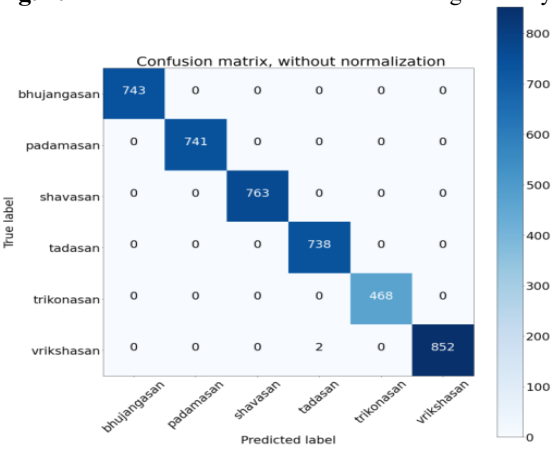


Fig. 8. Confusion matrix for testing accuracy

The testing was conducted using the Jupyter Lab IDE. The results of the real-time tests are presented in Table 1. Mediapipe achieved a frame rate of approximately 38 fps and took approximately 20ms to generate 30 input frames, making it suitable for real-time applications. Model achieved test accuracy of 99.27 in real time.

Table 1. Test Result in Real Time on proposed Hybrid Model

Sr. No	Asana	Total Cases	Correct Cases	Accuracy (%)
1	Bhujangasana	139	137	98.56%
2	Trikonasana	133	132	99.25%
3	Tadasana	132	132	100.00%
4	Shavasana	121	120	99.17%
5	Padmasanan	122	122	100.00%
6	Vrikshansana	144	142	98.61%
	Total	**791**	**785**	99.27%

Table 2. Comparative Analysis with Earlier Works On Same Dataset

Re-frence	Model	Key points	Op-timizer	Epoch	Train Time/ epoch	Training Accuracy	Validation Accuracy	Testing Accuracy		Speed In real Time
								Test Dataset	Real time	
Sk Yadav et al. [24]	Open-pose +CNN + LSTM	18	adam	100	22 Sec	99.34	99.41	99.38	98.92	5.6 fps
S. Jain et al. [25]	3DCNN	NA	SGD	100	200 Sec	99.96	99.39	99.39	90.5	20 fps
Pro-posed	Mdei-apipe+CNN + GRU	33	adam	44	13 Sec	99.95	99.79	99.95	99.27	38 fps

4 Result Discussion and analysis

The proposed hybrid model was trained on a collected dataset of six yoga poses, resulting in training and validation accuracy of 99.95% and 99.79%, respectively. This efficient generalization of the model was demonstrated through its performance evaluation. The model was tested on a test dataset and in real-time, both of which resulted in high accuracy levels of 99.95% and 99.27%, respectively. The Comparative analysis represented in Table. 2 between our proposed Yoga pose recognition hybrid model, which combines CNN and GRU, and Our proposed method demonstrated superiority in recognition accuracy and computational efficiency through the use of cutting-edge techniques.

The proposed model represents an improvement over previous studies [24,25] performed on the same dataset established by Yadav et al [24]. The training time of the proposed model is significantly shorter compared to the state of the art and using Mediapipe it operates at a speed of approximately 38 fps, making it suitable for real-time applications

5 Conclusion and Future work

This paper presents a computer vision based yoga pose recognition model that has the capability of recognizing yoga poses in videos as an action recognition task, and it is also capable of real-time prediction. A hybrid model of Mediapipe-CNN-GRU is designed, where Mediapipe extracts the 33-body key points as features, after which time distributed CNN used for spatial feature extraction and GRU for temporal prediction. The system was able to generalize well overall the six asanas Bhujangasana, Padmasana, Tadasana, Trikonasana, Shavasana, and Vrikshasana and proven to be highly efficient for yoga pose recognition as an action with high accuracy.

The model has undergone testing for six yoga poses, and for future work, a larger and more diverse dataset with a greater number of asanas can be created and the model can be tested on it. Integrating a feedback mechanism into the model to give users immediate feedback on the accuracy of their posture is another option. The model has the potential to be implemented on a portable device for real-time prediction assistance during self-training.

References

1. M. Vrigkas, C. Nikou, and I.A. Kakadiaris, "A review of human activity recognition methods," Frontiers in Robotics and AI, vol. 2, p. 28, 2015. doi: 10.3389/frobt.2015.00028.
2. S.-R. Ke, H. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi, “A review on video-based human activity recognition,” Computers, vol. 2, no. 2, pp. 88–131, Jun. 2013, doi: 10.3390/computers2020088.
3. H.Saini, S.Singh and Rathore V."Effect of yoga on perceived stress level of college students", Jagannath University Research Journal , Vol.2, Issue no.2, pp-62-65, November, 2021.
4. Q. Lei, J.-X. Du, H.-B. Zhang, S. Ye, and D.-S. Chen, “A survey of vision-based human action evaluation methods,” Sensors, vol. 19, no. 19, p. 4129, Sep. 2019, doi: 10.3390/s19194129.
5. H. Saini, R. Bagoria, and P. Arora, “Image and video datasets for yoga pose estimation: a review,” International Journal of Scientific Research and Engineering Trends, vol. 8, no. 6, pp. 1-10, Nov.-Dec. 2022.
6. S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network-a deep learning approach," Procedia Computer Science, vol. 132, pp. 679-688, 2018. doi.: 10.1016/j.procs.2018.05.069
7. M. S. Ansari, V. Bartoš, and B. Lee, "GRU-based deep learning approach for network intrusion alert prediction," Future Generation Computer Systems, vol. 128, pp. 235-247, Jan. 2022. doi:10.1016/j.future.2021.09.040

8. Z. Wu, J. Zhang, K. Chen, and C. Fu, “Yoga posture recognition and quantitative evaluation with wearable sensors based on two-stage classifier and prior bayesian network,” Sensors, vol. 19, no.23, p. 5129, Nov. 2019, doi: 10.3390/s19235129.
9. I. Rodriguez-Moreno, J. M. Martinez-Otzeta, B. Sierra, I. Rodriguez, and E. Jauregi, "Video activity recognition: state-of-the-art," Sensors, vol. 19, no. 14, p. 3160, Jul. 2019. doi: 10.3390/s19143160.
10. H.-T. Chen, Y.-Z. He, C.-C. Hsu, C.-L. Chou, S.-Y. Lee, B.-S.P. Lin, "Yoga posture recognition for self-training," in International Conference on Multimedia Modeling, Springer, 2014, pp. 496–505.
11. E.W. Trejo and P. Yuan, “Recognition of yoga poses through an interactive system with Kinect device,” in Proceedings of the 2018 2nd International Conference on Robotics and Automation Science, pp. 12–17, 2018. https://doi.org/10.1109/icras.2018.8443267
12. M. Gochoo, T.-H. Tan, S.-C. Huang, T. Batjargal, J.-W. Hsieh, F.S. Alnajjar, and Y.-F. Chen, “Novel IoT-based privacy-preserving yoga posture recognition system using low-resolution infrared sensors and deep learning,” IEEE Internet of Things Journal, vol. 6, no. 4, pp. 7192–7200, 2019.
13. Rishan, B. De Silva, S. Alawathugoda, S. Nijabdeen, L. Rupasinghe, and C. Liyanapathirana, "Infinity yoga tutor: yoga posture detection and correction system," in 2020 5th International Conference on Information Technology Research (ICITR), Dec. 2020, pp. 1-6..
14. R. Josyula and S. Ostadabbas, "A review on human pose estimation," arXiv preprint arXiv:2110.06877, Oct. 2021.
15. S. Dubey and M. Dixit, "A comprehensive survey on human pose estimation approaches," Multimedia Systems, pp. 1-26, Feb. 2022.
16. S. Kothari, “Yoga pose classification using deep learning,” Ph.D. thesis, San Jose State University, 2020.
17. C. H. Lin, S. W. Shen, I. T. Anggraini, N. Funabiki, and C. P. Fan, "An OpenPose-based exercise and performance learning assistant design for self-practice yoga," in 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), Piscataway, NJ, USA, 2021, pp. 456-457, doi: 10.1109/GCCE51884.2021.9501763.
18. M. Noori, B. Wallace, M. Z. Uddin, and J. Torresen, "A robust human activity recognition approach using OpenPose, motion features, and deep recurrent neural network," in Image Analysis: 21st Scandinavian Conference, SCIA 2019, Norrköping, Sweden, June 11–13, 2019, Proceedings, Cham, Switzerland, 2019, pp. 299-310, doi: 10.1007/978-3-030-21278-3_25.
19. K. Yamao and R. Kubota, "Development of human pose recognition system by using raspberry pi and posenet model," in 2021 20th International Symposium on Communications and Information Technologies (ISCIT), Tokyo, Japan, 2021, pp. 41-44, doi: 10.23919/ISCIT52477.2021.9461238.
20. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in Proceedings of the IEEE international conference on computer vision, Santiago, Chile, 2015, pp. 2938-2946, doi: 10.1109/ICCV.2015.334.
21. Jo and S. Kim, "Comparative analysis of OpenPose, PoseNet, and MoveNet models for pose estimation in mobile devices," Traitement du Signal, vol. 39, no. 1, pp. 119-124, 2022, doi: 10.18280/ts.390109.
22. J. Zhao, Y. Cao, and Y. Xiang, "Pose estimation method for construction machine based on improved AlphaPose model," Engineering, Construction and Architectural Management, pp. 1-13, 2022, doi: 10.1108/ECAM-01-2022-0036.

23. M. Chasmai, N. Das, A. Bhardwaj et al., "A view independent classification framework for yoga postures," SN Computer Science, vol. 3, no. 6, p. 476, 2022, doi: 10.1007/s42979-022-01376-7.
24. S.K. Yadav, A. Singh, A. Gupta, and J.L. Raheja, "Real-time yoga recognition using deep learning," Neural Computing & Applications, vol. 31, no. 12, pp. 9349–9361, 2019,doi: 10.1007/s00521-019-04232-7
25. S. Jain, A. Rustagi, S. Saurav, R. Saini, and S. Singh, "Three-dimensional CNN-inspired deep learning architecture for yoga pose recognition in the real-world environment," Neural Computing & Applications, pp. 1-15, 2020. doi:10.1007/s00521-020-05405-5
26. Y. Zhang, "Applications Of Google Mediapipe Pose Estimation Using A Single Camera", California State Polytechnic University, Pomona. [Available online: https://scholarworks.calstate.edu/downloads/n009w777f]