Journal of Computer Science

Original Research Paper

A Deep Learning Framework for Classifying and Evaluating Yoga Exercises

Mehar Latif Malik and Arun Khosla

Department of Electronics and Communication Engineering, Dr. B R Ambedkar National Institute of Technology, Jalandhar, India

Article history
Received: 15-05-2022
Revised: 16-06-2022
Accepted: 22-06-2022

Corresponding Author:
Mehar Latif Malik
Department of Electronics and Communication Engineering,
Dr. B R Ambedkar National Institute of Technology,
Jalandhar, India
Email: meharmalik454@gmail.com

Abstract: Yoga is a type of physical activity that combines physical postures with breathing techniques. Cardiovascular disease is the leading cause of illness and mortality. Yoga's potential involvement in cardiac recovery can be incorporated into cardiovascular rehabilitation. Classifying and evaluating yoga exercises using a deep learning framework plays a significant role in postoperative recovery. Home based rehabilitation often suffers from a lack of patient adherence to prescribed exercise routines, resulting in longer treatment times and higher healthcare expenses. This study proposes a framework for classifying and evaluating the quality of yoga exercises. The ConvNet followed by the deep belief network model is used to create quality scores for input movements to evaluate the quality of yoga asanas. Additionally, various machine learning algorithms are deployed to identify the best performing algorithm. The proposed framework is tested using the COCO dataset and the publicly available dataset from KAGGLE. The model is implemented using a three stage pipeline. In the first stage, an estimator and detector model is used for feature extraction. In the second stage, restricted Boltzmann machines are used for dimensionality reduction in two steps: The forward pass and the backward pass. Finally, the ConvNet followed by the deep belief network is used to classify yoga postures. The quaternion data is treated as a multivariate Gaussian variable to create Markov random fields. To quantify yoga asanas, the Markov random fields network is triggered, and the resulting quaternion point is compared to standard pose data, increasing the likelihood that the yoga position is correct. The importance of our study is that it allows participants to watch their movement in real-time and to obtain a numerical estimate of the accuracy of their yoga stance. Using this method, we were able to achieve a precision of 99.99%, which was previously unattainable on this dataset. Our model is more robust than previous models in terms of performance.

Keywords: ConvNet, PCA, Yoga Asanas, Cardiovascular Rehabilitation, Performance Assessment

Introduction

Yoga is one of the most popular physical activities, however, there are only very few professional yoga instructors. For the vast majority of yoga newbies, self-study, such as systematically repeating a yoga video, is the only method to learn yoga. As a result, they have no idea whether their performance was satisfactory or not. The goal of appraising yoga stance is to statistically analyze yoga postures to accomplish yoga pose perception and pose effectiveness (Lei *et al.*, 2019; Li *et al.*, 2021); it may distinguish between separate moves by scrutinizing stance features. The most crucial part of yoga practice is good form, as an incorrect position could be harmful, therefore result in contusion (Swain and McGwin, 2016; Russell *et al.*, 2016; Wiese *et al.*, 2019). Nevertheless, all participants are not able to collaborate with an adept educator. Most yoga newbies practice yoga by autodidactism, including meticulously viewing a real time yoga practice. As a result, assessing yoga positions is critical for identifying yoga poses and informing participants (Yu and Huang, 2020). Yoga is gaining popularity in curative analysis, and a plethora of work has been offered for a variety of medicinal uses, particularly intervention for a good body image (Neumark-Sztainer *et al.*, 2018;

Halliwell *et al*., 2019), and psychiatric disorders (Sathyanarayanan *et al*., 2019). Yoga can cure numerous disorders without the need for medications (Patil *et al*., 2011). Among deep learning architectures, CNNs are the most frequently used for the field of vision. CNNs automatically learn a few representational characteristics (Sitaula *et al*., 2021). Yoga is made up of several asanas, each of which symbolizes a static bodily stance (Bromley *et al*., 1993).

Yoga position evaluation is an onerous undertaking. The foremost problem stems from the scarcity of a yoga position evaluation baseline because labeling at the image level is expensive. Another difficulty stems from a basic mismatch between the practitioner's stance and the ideal position stance. The clustered characteristics derived from deep characteristics from the models which have been trained in advance might be much more resilient than a unique piece of characteristics (Chen *et al*., 2020). Furthermore, skeleton knowledge may be sturdy enough to control this variability. The contrastive learning technique was developed to address these issues (Chen and He, 2021; Hu *et al*., 2021; Haresamudram *et al*., 2021).

The main concept is to use a discriminative learning strategy to acquire encoded characteristic depictions, wherein slightly closer pairs stay near together and diverse sample pairs stay far apart. It has been effectively validated in various computer vision usage, including picture categorization (Khaertdinov *et al*., 2021) as well as human activity recognition (Chen *et al*., 2018; 2014). Also, the approaches using the skeleton to obtain human body key points to determine the correct stance have been proffered (Chen *et al*., 2013).

We presented a framework for classifying and evaluating the quality of yoga for modeling various full body stances with increased precision and filling in the gaps in posture assessment. Our proffered approach doesn’t require any special hardware but a simple web camera. The model is implemented using a three stage pipeline. In the first stage, an estimator and detector model is used for feature extraction. In the second stage, restricted Boltzmann machines are used for dimensionality reduction. Finally, the ConvNet followed by the deep belief network is used to classify yoga postures. To quantify yoga asanas, the Markov Random Fields network is triggered, and the resulting quaternion point is compared to standard pose data, increasing the likelihood that the yoga position is correct. The suggested method detects the five most frequently practiced asanas in real time. Virabhadrasana II, Phalakasana, Vrikshasana, Utkata Konasana, and Adho Mukha Shvanasana are among these asanas. Additionally, we have added a new class labeled 'Unknown Pose' to denote if the posture does not fit into any of the current asanas.

Materials and Methods

The key points which are obtained consist of the mouth, nose, left wrist, right wrist, neck, and elbow (Table 1).

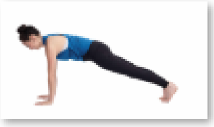
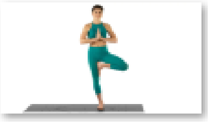
For training we used an image dataset for yoga posture categorization adopted from Kaggle, (Available online: Yoga pose image classification dataset | Kaggle) consisting of 45 categories and 1931 photos broadly categorized under 5 categories namely Virabhadrasana II, Phalakasana, Vrikshasana, Utkata Konasana, and Adho Mukha Shvanasana. A summary of these classifications is provided in Table 2. Additionally, for testing, we used an additional dataset from different subjects. Methods such as drawing a body map (Fig. 1) and using the human skeleton to acquire critical spots on the human body to establish the proper stance have been proposed. The photos in this collection were collected at a variety of different resolutions and against a variety of different backdrops. The overview of the 45 different types of yoga poses included in the dataset (Fig. 2).

Table 1: Key points used

No.	Key points	No.	Key points
0	Mouth	11	Left knee
1	Right ear	12	Nose
2	Left ear	13	Right eye
3	Right shoulder	14	Left eye
4	Left shoulder	15	Right wrist
5	Neck	16	Left wrist
6	Right elbow	17	Right foot
7	Left elbow	18	Left foot
8	Right hip		
9	Left hip		
10	Right knee		

DOI: 10.3844/jcssp.2023.229.241

Table 2: Details of each yoga pose

S. no.	Asana	Posture
1	Phalakasana (plank)	
2	Shvanasana (downdog)	
3	Vrikshasana (tree)	
4	Utkata konasana (goddess)	
5	Virabhadrasana (warrior 2)	

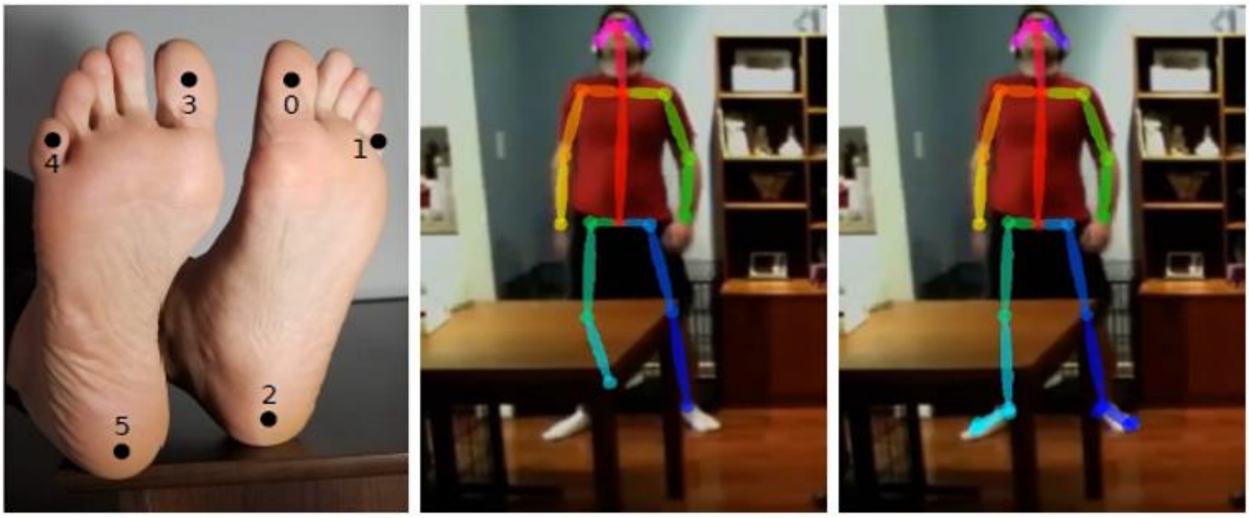


Fig. 1: Body map capture for obtaining key points

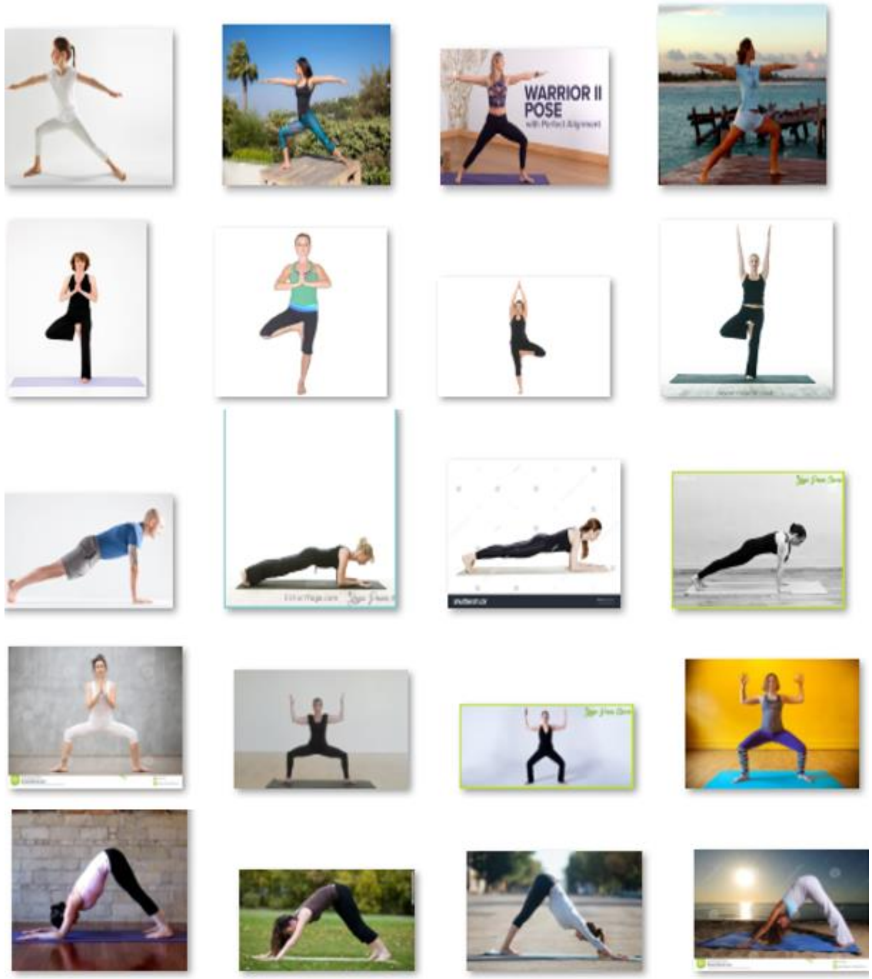


Fig. 2: A summary of 45 kinds of yoga positions included in the dataset

Methodology

This research proposes a framework for categorizing and evaluating the quality of yoga workouts. The quality of movement is evaluated, and a scoring procedure is used. A deep learning model is used to provide quality scores for input motions for judging the quality of yoga asanas. Furthermore, multiple machine learning algorithms are used to find the best-performing model. The proposed framework is validated using the COCO dataset and a publicly available dataset from KAGGLE, which combines a diverse population with 1931 photos and 45 categories. The proposed model is built in three stages, the first of which is an estimator and detector model used for feature extraction. For training, the estimator uses a heat map. There are two ways to use detector. The foremost is in box type mode and the second is in alignment and positioning mode, where scale along with the angle is calculated. Secondly, restricted Boltzmann machines are implemented in two steps i.e., forward pass and backward pass. RBMs are composed of two layers, visible and hidden units. Every exposed unit is linked to every hidden unit. RBMs have no output nodes and have a bias unit that is coupled to all the visible and hidden units. In the forward pass, RBMs take the inputs and transform them into a collection of integers that encodes the inputs. RBMs combine each input with its weighting and a single overall bias. The output is sent to the hidden layer by the algorithm. RBMs take that collection of integers and transform those to generate the recreated inputs in the backward pass. Every exposed node extracts a minimal characteristic from a learning item in the database. The output of the node, or the intensity of the signal traveling through it, is determined by the outcome of these two procedures, which are supplied into an activation function. Lastly, ConvNet followed by Deep Belief Network (DBN) intends to categorize yoga postures. The Greedy learning algorithm is used to pre-train DBN. For learning the top-down, generative weights, the greedy learning method employs a layer by layer approach. These generative weights define the relationship between

DOI: 10.3844/jcssp.2023.229.241

variables in one layer and variables in the layer above. On the top two hidden layers, numerous steps of Gibbs sampling in DBN are conducted. The top two hidden layers define the RBM, thus this stage is effectively extracting a sample from it. The sample is obtained from the visible units using a single pass of ancestral sampling over the rest of the model. A single, bottom-up pass can be used to infer the values of the latent variables in each layer. In the lowest layer, greedy pretraining begins with an observed data vector. It then oppositely fine-tunes the generating weights. The quaternion data is treated as a multivariate Gaussian variable to build Markov random fields. To quantify yoga asanas, the Markov random fields network is triggered, and the resulting quaternion point is compared to standard pose data.

In Markov random field likelihood function p over variables $y_{1,}$ y_n formed by an undirected graph U, the nodes of which correlate to variables y_i. Where C is the set of cliques (i.e., completely connected subgraphs) of U, and each factor f is a non-negative function over the parameters in a clique, the probability p has the form shown in Eq. 1:

$$p(y_1,...y_n)\frac{1}{X}\prod_{f\in S}\emptyset_f(y_f) \tag{1}$$

where:

$$X=\sum_{y_1,...,y_n}\prod_{f\in S}\emptyset_f(y_f)$$

The probability function of the Gaussian variable pertaining to each body part is employed to assess the yoga stance.

Skeleton Extraction

The estimator and detector model is used for feature extraction. For training, the estimator uses a heat map. There are two ways to use detector. The first is in box mode, and the second is in alignment mode. In the second stage, for dimensionality reduction Restricted Boltzmann Machines are implemented in two steps i.e., forward pass and backward pass. Lastly, ConvNet followed by deep belief network intends to categorize yoga postures. The quaternion data is treated as a multivariate Gaussian variable to build Markov random fields. To quantify yoga asanas, the Markov random fields network is triggered, and the resulting quaternion point is compared to standard pose data. The probability function of the Gaussian variable pertaining to each body part is employed to assess the yoga stance. Consider $p_{x\text{-}gem}$ to be the quaternion rotations. Before beginning the training, it is important to do a posture calibration. When the calibration is complete, $p_{scaled,x\text{-}gem}$ will be recorded for the quaternion data points. Local coordinate systems are fixed and the expression is expressed in O-xyz. Additionally, O_1-$x_1y_1z_1$ and O_2-$x_2y_2z_2$ comparative to O_{ref}-$x_{ref}y_{ref}z_{ref}$ have been considered beforehand with $p_{ini1\text{-}interface}$ and $p_{ini2\text{-}interface}$ values given in Eq. 2 and 3 respectively:

$$p_{ini1-interface}=\left(0,\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2},0\right) \tag{2}$$

and:

$$p_{ini2-interface}=\left(\frac{1}{2},-\frac{1}{2},-\frac{1}{2},\frac{1}{2},\right) \tag{3}$$

$p_{x(interface)}$ is quaternion obtained in real time and is obtained from Eq. 4 and 5 respectively:

$$p_{x(interface)}=p_{ini1-interface}\otimes p^{-1}_{scaled,x-gem}\otimes px-gem$$

$$\forall x\in\{0,1,6,7,8,9,10\} \tag{4}$$

$$p_{x(interface)}=p_{ini2-interface}\otimes p^{-1}_{scaled,x-gem}\otimes px-gem$$

$$\forall x\in\{2,3,4,5\} \tag{5}$$

Model

We propose three stages deep learning pipeline for the classification of a given image to the corresponding yoga pose. Figure 3 depicts a block schematic of the proposed system for assessing yoga asanas. The skeletal joint coordinates are processed through dimensionality reduction, which is then utilized to train a NN model followed by a deep belief network for performance assessment and scoring mapping to produce movement quality scores for yoga asanas. The trained ConvNet model followed by a deep belief network creates movement scores for input data.

Dimensionality reduction of collected data is a key stage in analyzing different movements because it suppresses insignificant, redundant, or highly linked variables. Restricted Boltzmann machines are used for Dimensionality reduction. RBMs are composed of two layers: Units that are visible and units that are hidden. Every exposed unit is linked to every hidden unit. RBMs have no output nodes and have a bias unit that is coupled to all the visible and hidden units. Forward pass and backward pass are the two steps of implementation. In the forward pass, RBMs take the inputs and transform them into a collection of integers that encodes the inputs. RBMs combine each input with its weighting and a single overall bias. The output is sent to the hidden layer by the algorithm. RBMs take that collection of integers and transform those to generate the recreated inputs in the backward pass. Every exposed node extracts a minimal characteristic from a learning item in the database. The output of the node, or the intensity of the signal traveling through it, is determined by the outcome of these two procedures, which are supplied into an activation function. The framework includes feature extraction, dimensionality reduction methods, performance measurements, scoring algorithms, and deep learning models. This study highlights the use of deep learning models for evaluating yoga exercises. The merits of NNs for this task stem from their ability to describe human movements hierarchically at several temporal and spatial abstraction levels. These models improve the ability to "understand" the degrees of hierarchy and the intricate spatiotemporal relationships in human movement data. The deep learning model utilized in this case is CNN followed by the deep belief network. They are widely used to solve pattern recognition difficulties and intend to categorize yoga postures. The quaternion data acquired for yoga asanas evaluation is treated as a multivariate Gaussian variable.

To quantify yoga asanas, the Markov random fields network is triggered, and the resulting quaternion point is compared to standard pose data, raising the likelihood that the yoga position is more similar to the proper stance. The softmax layer provides the likelihood of each yoga in each instance. This number is thresholded to detect instances in which the user is not practicing yoga, and the impact of pooling has been evaluated. CNN layer containing 16 filters of size 3 × 3 and ReLU activation is used. For faster convergence, batch normalization is performed on the CNN output.

This is succeeded by a dropout layer, which lowers a fragment of the weight on a randomized basis to prevent overfitting. This output is thresholded to detect whether the user is practicing yoga or not. Figure 4 represents the structured system architecture of CNN and Fig. 5 represents the architecture of the deep belief network. Our technique tries to recognize a performer's yoga asanas automatically from images and live videos. The procedure can be broken down into four distinct steps. To begin, data collection is carried out, which may be a real time operation running concurrently with detection or previously recorded instances. Second, estimator and detector are used to locate the joint locations, followed by bipartite matching and parsing. This is followed by dimensionality reduction using restricted Boltzmann machines. Lastly, ConvNet followed by deep belief network intends to categorize yoga postures.

The quaternion data is treated as a multivariate Gaussian variable to build Markov random fields. To quantify yoga asanas, the Markov random fields network is triggered, and the resulting quaternion point is compared to standard pose data, raising the likelihood that the yoga position is more similar to the proper stance. The importance of our study is that it will allow participants to watch their movement in real-time, as well as gain a numerical estimate accuracy of their yoga stance.

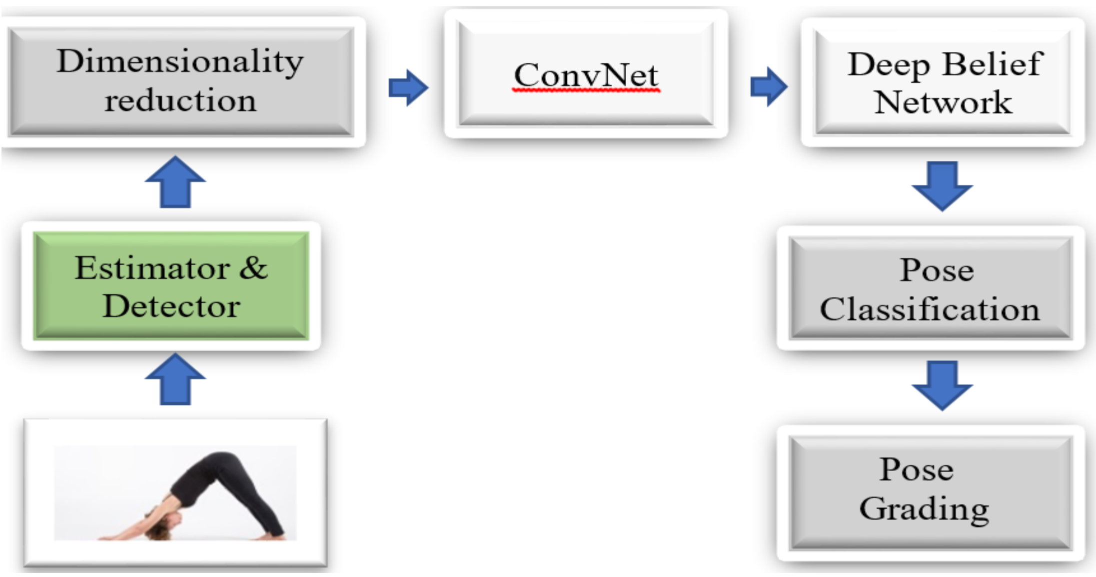


Fig. 3: Schematic of the proposed system for assessing yoga asanas

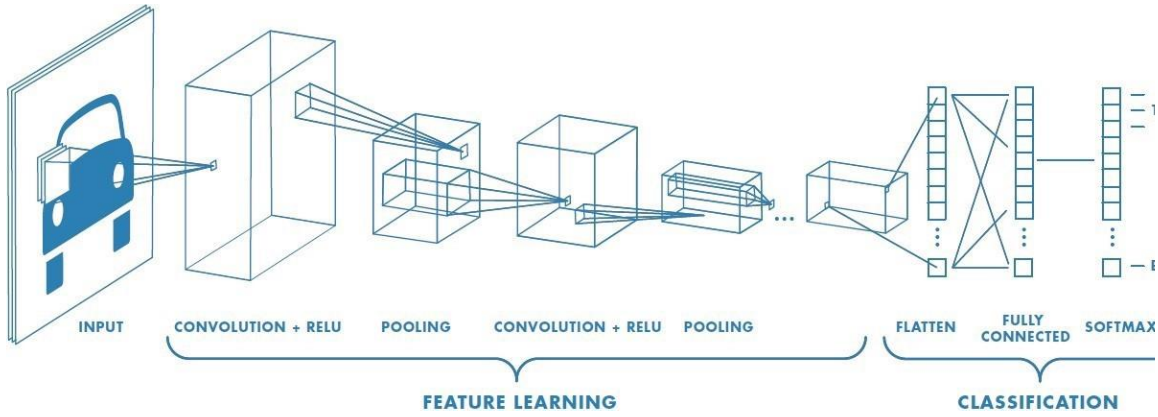


Fig. 4: The structured system architecture of CNN

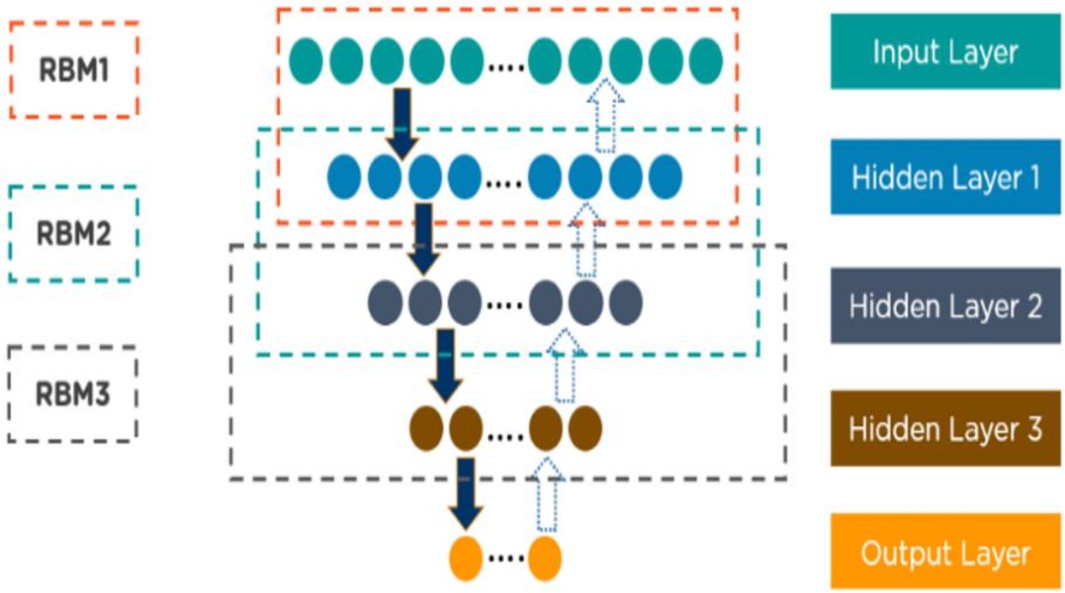


Fig. 5: Deep belief network

Training

Our objective is to accurately classify and assess the yoga asanas. To begin, key point characteristics are retrieved. With estimator and detector and recording the joint orientation data, followed by dimensionality reduction, asanas are predicted using the proposed model. The categorical cross-entropy loss function is utilized because it is well-suited for evaluating the output of a fully connected layer when Softmax activation is used. Adam optimizer with an initial learning rate of 0.0001 and no decay is utilized. The joint estimation model is passed as a regression problem and trained to optimize the mean square error loss function trained with the COCO dataset. This model is trained until convergence and validated with a validation dataset.

On a PC with an Intel i7 7700HQ processor, 64GB RAM, and an Nvidia GTX 960M GPU, the model was trained for 100 epochs. After 100 epochs of training, the proffered method achieved 99.99% on validation data. The training takes approximately 21 sec for each epoch, which is quite fast given the minimal inputs and concise design. Figure 6 and 7 illustrate the evolution of the precision and loss function, correspondingly, throughout training. Initially, the training and testing precision grows rapidly, with the validation accuracy remaining greater than the training accuracy. For future testing, the weights of the best suitable model with the highest accuracy rate are retained. Training and validation loss have reduced uniformly, and training was terminated when the difference in loss between epochs was less than 0.001. This condition was sufficient to conclude that model has converged to local minima.

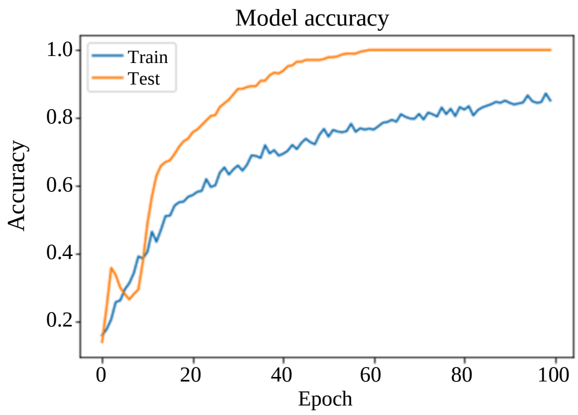


Fig. 6: Model accuracy over the epochs

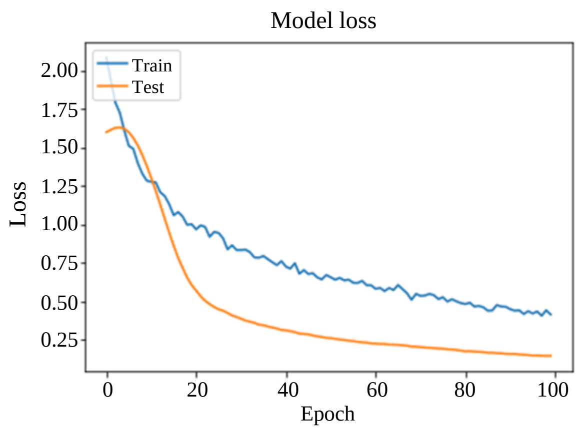


Fig. 7: Model loss over the epochs

Results

The final stage displays real-time prediction results for a diverse range of people using the webcam and all conceivable variants using the thresholding algorithm. The model predicts yoga poses if the confidence of the model is more than 0.8 for prediction otherwise the model outputs 'Unknown Pose' when the confidence value is less than the desired threshold. This threshold is a hyperparameter and was chosen empirically after a test run with different values. After 100 epochs of training, the proffered method achieved 99.99% on validation data.

Apart from Virabhadrasana for test situations, the performance is much above and close to ideal in the majority of asanas. Shvanasana was incorrectly classified as Phalakasana. Similarly, Vrikshasana is occasionally misclassified as Utkata Konasana. This discrepancy could be explained by the fact that both asanas performed have the first stages of creation extremely similar. This can be rectified with the addition of more data for training and adding regularization using a deeper classification model but these are beyond the scope of this project and will be explored in future works. Moreover, various machine learning algorithms are deployed to obtain the best-performing algorithm which is compared to our Proposed Model. Our proposed model outperforms all other methods employed. Table 3 lists all the algorithms that were employed and Fig. 8 gives a graphical representation of the accuracies obtained. Yoga poses were classified and assessed using the most accurate model as shown in Fig. 9.

Table 3: Algorithms employed to compare with ConvNet

Algorithms	Accuracy (%)
Logistic regression	88.49
Ridge classifier	88.49
Random forest	92.33
Gradient boosting	92.65
K nearest neighbor	85.62
Proposed ConvNet model	99.99

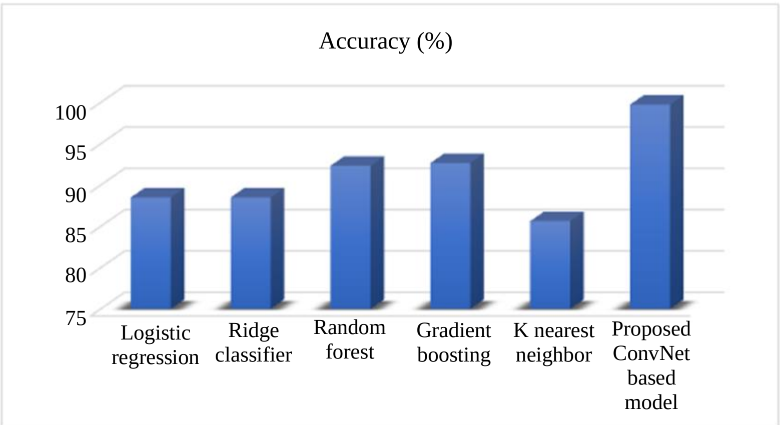


Fig. 8: Graphical representation of accuracies

Fig. 9: Classification and evaluation of yoga pose

Prediction in Real Time Via Thresholding

Real time predictions can be made using any RGB webcam. Numerous individuals were involved in the data gathering and real time prediction process. All participants performed asanas indoors at 2-3 m from the camera. A threshold was used to the softmax output to infer whether or not the user was executing one of the asanas. A threshold value of 0.80 has been determined to be desirable, as a less value resulted in false positives in the 'Unknown Pose' situation, but a greater value resulted in true negatives when asana was carried out. Thus, estimator and detector's net resolution options can be changed by the user for rapid but less accurate results. Figure 10 represents the prediction of yoga asanas in real-time.

Performance Metrics

Table 4 represents the performance metrics of our model.

Classification Precision

It is the ratio of several accurate predictions and the total number of samples in the input. It works if each class has an equivalent number of observations:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Fig. 10: Prediction of yoga asanas in real-time

Confusion Matrix

The confusion matrix produces a matrix that highlights the overall performance of the model. Figure 11 represents the confusion matrix of our model.

There are four important terms to remember:

True Positive: Instances in which we anticipated YES and the real outcome was likewise YES
True Negative: When we anticipated NO and the real output was NO
False Positive: When we anticipate YES but the real result was NO
False Negative: When we anticipated NO and the real result was YES

The matrix's precision may be measured by taking the average of the values spanning over the "major diagonal," i.e.:

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ sample}$$

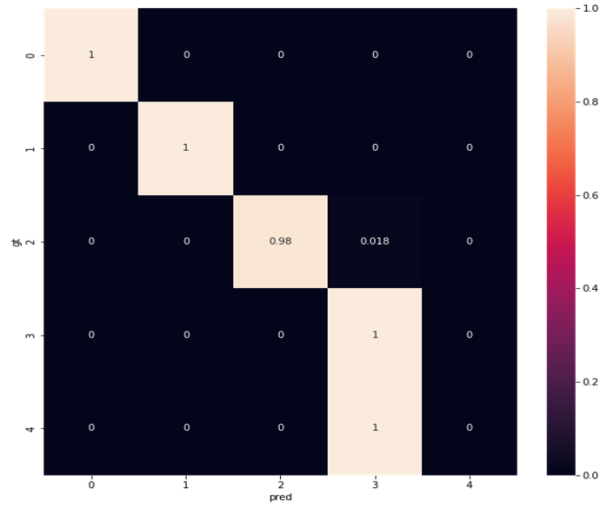


Fig. 11: Confusion matrix for the proposed model

Area Under Curve

One of the most used measures for assessment is Area Under Curve (AUC). The AUC of a classifier is the probability that it would score a randomly picked positive example higher than a randomly selected negative example. Let us first define fundamental words before delving into AUC:

True Positive Rate (Sensitivity): TPR is defined as:

$$Accuracy=\frac{True\,Positive}{(False\ Negative+True\,Positive)}$$

True negative rate (specificity): TNR is defined as:

$$Accuracy=\frac{True\,Negative}{(False\ Positive+True\,Negative)}$$

False Positive Rate (FP): FP is defined as:

$$Accuracy=\frac{False\ Positive}{(False\ Positive+True\,Negative)}$$

The false positive rate and true positive rate each have values between 0 and 1. FPR and TPR are calculated at various threshold values (0.00, 0.02, 0.04...., 1.00), and a graph (Fig. 12) is obtained. AUC is defined as the area under the curve of a plot of false positive rate vs true positive rate at various positions in [0, 1].

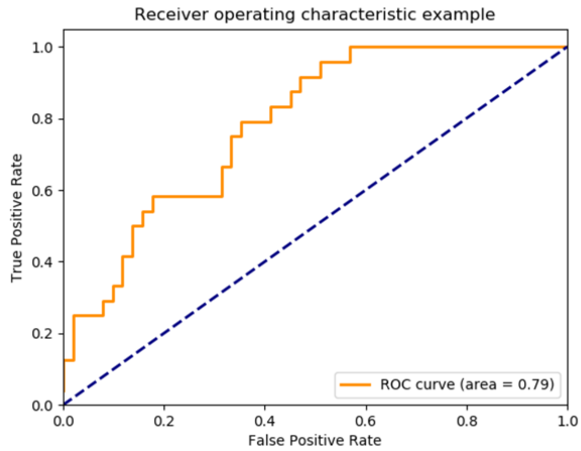


Fig. 12: AUC for false positive rate vs true positive rate

F1-Score

The F1-score is used to determine whether or not a test is correct. The F1-score is defined as the harmonic mean of accuracy and recall. F1-score has a range of [0, 1]. It represents the classifier's exactness and resilience. F1-score attempts to strike a compromise between precision and recall:

$$F1=2*\frac{Recall*Precision}{(Recall+Precision)}$$

Precision

It is defined as a ratio of correct positive outcomes and positive results predicted:

$$\frac{True\,Positive}{(True\,Positive+False\,Positive)}$$

Recall

It is defined as the ratio of the number of correct positive outcomes and a total number of relevant instances:

$$\frac{True\,Positive}{(True\,Positive+False\,Negative)}$$

Table 4: Performance metrics of the proposed model

Performance metrics					
Precision	[10	1	1.0000	0.4076900	0]
Recall	[10	1	0.9816	1.0000000	0]
F1-score	[10	1	0.9907	0.5792397	0]
Average	[79	59	109.0000	53.0000000	75]

DOI: 10.3844/jcssp.2023.229.241

Discussion

The study presents a unique methodology for assessing yoga asanas using deep learning. The system includes performance indicators, scoring systems, and deep learning model. The amount of consistency in captured rehabilitative motions is quantified using common measures. NNs are trained for each workout with the inferred outputs being quality ratings for inputs consisting of exercise repetitions. Restricted Boltzmann machines are used for dimensionality reduction. RBMs are composed of two layers, visible and hidden units. Every exposed unit is linked to every hidden unit. RBMs have no output nodes and have a bias unit that is coupled to all the visible and hidden units. In the forward pass, RBMs take the inputs and transform them into a collection of integers that encodes the inputs. RBMs take that collection of integers and transform those to generate the recreated inputs in the backward pass. Every exposed node extracts a minimal characteristic from a learning item in the database. The output of the node, or the intensity of the signal traveling through it, is determined by the outcome of these two procedures, which are supplied into an activation function. Lastly, ConvNet followed by deep belief network intends to categorize yoga postures. The quaternion data is treated as a multivariate Gaussian variable to build Markov random fields. To quantify yoga asanas, the Markov random fields network is triggered, and the resulting quaternion point is compared to standard pose data, raising the likelihood that the yoga position is more similar to the proper stance. Moreover, various machine learning algorithms are deployed to obtain the best performing algorithm. Compared to model less techniques, probabilistic approaches, have enhanced capacity to manage the inherent variability and measurement uncertainty in human movement data. The importance of our study is that it will allow participants to watch their movement in real-time, as well as gain a numerical estimate accuracy of their yoga stance. Using this method, we were able to reach a precision of 99.99%, which was previously unattainable. In future work, we will seek to solve the study's weaknesses, focusing on a thorough validation of the framework of yoga asanas done by learners and labeled by a panel of experts who will award quality scores. We will validate the suggested technique by measuring muscle activation.

Table 5 compares a few existing methods to our suggested solution. Our accuracy was greater than the model proposed by Yadav *et al*. (2019). It should be noted that we modeled and tested a greater number of postures. Additionally, they have not utilized the yoga identification system to analyze and lead the yoga practice poses. AdaBoost algorithm (Trejo and Yuan, 2018), and Open Pose (Qiao *et al*., 2017) were widely employed in yoga. They replicated only a few yoga postures without body folding and the accuracy of recognition was lower than that of our proposed strategy. In the Star skeleton template (Chen *et al*., 2018), two cameras were utilized to distinguish 12 yoga positions, and visual input was employed to provide feedback. Replication of motion (Luo *et al*., 2011) was a similar approach for recognizing and evaluating Yoga postures. Our proposed approach shows the greatest precision as compared to previous approaches (Huang *et al*., 2014). Recently, a CNN-LSTM hybrid was developed and used for tasks such as sentiment classification (Wang *et al*., 2016), text classification (Zhou *et al*., 2015), cardiac diagnosis (Oh *et al*., 2018), anti-spoofing of faces (Xu *et al*., 2015).

Outcomes of Posture Recognition

To build the classifier, 70% of the data is selected for training and 30% of the data is used for testing. 45 categories broadly classified into 5 are used throughout the training period. The proposed detection method was implemented using cumulative probability. Certain postures were too similar to others, which slightly reduced the accuracy of the detection.

Table 5: Comparative study of several ways for recognizing and evaluating yoga

Technique	Detector	Accuracy of posture classification	Posture grading
Deep learning (Yadav *et al*., 2019)	RGB webcam	98.92%	No
AdaBoost algorithm (Trejo and Yuan., 2018)	A depth sensor based camera	94.78%	No
Open pose (Qiao *et al*., 2017)	An RGB camera	Not applicable	No
Template star skeleton (Chen *et al*., 2018)	Two cameras	94.30%	Yes
Motion replication (Luo *et al*., 2011)	16 IMUs and 6 tactors	Not applicable	No
Star skeleton (Huang *et al*., 2014)	Kinect	99.33%	No
Proposed model	Webcam	99.99%	Yes

Findings from the Posture Evaluation

In training data, 30% of the database contained data on yoga postures performed both conventionally and in an intentionally non-standard manner. During the training stage, the network was trained using the quaternion data from the training set. The evaluation stage included testing with both conventional and non-standard postures. The probabilities of nonstandard components were virtually all less than 0.3, whereas probabilities of standard portions were almost uniformly greater than 0.5. The recognition outcome demonstrates that the two stage classifier is capable of effectively discerning yoga postures and obtaining high accuracy. The proposed recognition approach has the potential to efficiently reduce noisy outcomes using cumulative probability, and enhanced precision in recognition.

Conclusion

This research proffers a full-body posture modeling and assessment method to recognize and assess yoga asanas. Movement quality is assessed and a scoring method is employed to translate these statistics into an overall movement quality score. Furthermore, a variety of Machine Learning algorithms are employed to determine which method is the most effective when compared to the Proposed Model. The proposed model is executed in three stages; the first stage involves the deployment of an estimator and detector model for feature extraction. Secondly, for dimensionality reduction, Restricted Boltzmann machines are implemented in two steps i.e., forward pass and backward pass. Lastly, ConvNet followed by deep belief network intends to categorize yoga postures. The significance of our study is that it will enable participants to observe their movements in real-time while also obtaining a numerical evaluation of how precise their yoga stance is and how far they are from the ideal pose. We can achieve a precision of 99.99% with this strategy, which was previously unreachable. Since it was trained on a large dataset, our model is more robust in terms of performance than previous models. It's worth noting that our solution eliminates the need for any special hardware for yoga posture recognition and can be implemented using input from a standard RGB camera thereby reducing the hardware cost. Additionally, the system can be deployed on a portable device to provide real-time prediction and self-training. This study provides an example of how activity recognition systems can be used in practical applications. A related approach may be used to recognize postures in a variety of contexts, including surveillance, athletics, healthcare, and image categorization.

Acknowledgment

We would like to express our gratitude to NIT Jalandhar for giving us the chance to conduct this research and for their unwavering support.

Funding Information

The authors have not received any funding for the report.

Author's Contributions

Mehar Latif Malik: Assess, research, analyze, and modeling.

Arun Khosla: Review, discussion, and supervision.

Ethics

The article is authentic and contains unpublished material. The corresponding author affirms that no ethical concerns exist, and all authors have read and endorsed the article.

References

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1993). Signature verification using a" siamese" time delay neural network. *Advances in Neural Information Processing Systems*, 6. https://doi.org/10.1142/S0218001493000339

Chen, H. T., He, Y. Z., & Hsu, C. C. (2018). Computer-assisted yoga training system. *Multimedia Tools and Applications*, 77(18), 23969-23991. https://doi.org/10.1007/s11042-018-5721-2

Chen, H. T., He, Y. Z., Chou, C. L., Lee, S. Y., Lin, B. S. P., & Yu, J. Y. (2013, July). Computer-assisted self-training system for sports exercises using kinects. *In 2013 IEEE International Conference on Multimedia and Expo Workshops* (ICMEW) (pp. 1-4). https://doi.org/10.1109/ICMEW.2013.6618307

Chen, H. T., He, Y. Z., Hsu, C. C., Chou, C. L., Lee, S. Y., & Lin, B. S. P. (2014, January). Yoga posture recognition for self-training. *In International Conference on Multimedia Modeling*. Springer, Cham. (pp. 496-505). https://doi.org/ 10.1007/978-3-319-04114-8_42

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. *In International conference on machine learning* (pp. 1597-1607). PMLR. https://doi.org/10.48550/arXiv.2002.05709

Chen, X., & He, K. (2021). Exploring simple siamese representation learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 15750-15758). https://doi.org/10.48550/arXiv.2011.10566

Halliwell, E., Dawson, K., & Burkey, S. (2019). A randomized experimental evaluation of a yoga-based body image intervention. *Body Image*, *28*, 119-127. https://doi.org/10.1016/j.bodyim.2018.12.005

Haresamudram, H., Essa, I., & Plötz, T. (2021). Contrastive predictive coding for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2), 1-26. https://doi.org/10.1145/3463506

Hu, X., Li, T., Zhou, T., Liu, Y., & Peng, Y. (2021). Contrastive learning based on the transformer for hyperspectral image classification. *Applied Sciences*, *11*(18), 8670. https://doi.org/10.3390/app11188670

Khaertdinov, B., Ghaleb, E., & Asteriadis, S. (2021, August). Contrastive self-supervised learning for sensor-based human activity recognition. *In 2021 IEEE International Joint Conference on Biometrics* (IJCB) (pp. 1-8). https://doi.org/10.1109/IJCB52358.2021.9484410

Lei, Q., Du, J. X., Zhang, H. B., Ye, S., & Chen, D. S. (2019). A survey of vision-based human action evaluation methods. *Sensors*, 19(19), 4129. https://doi.org/10.3390/electronics11030453

Li, J., Hu, Q., Guo, T., Wang, S., & Shen, Y. (2021). What and how well you exercised? An efficient analysis framework for fitness actions. *Journal of Visual Communication and Image Representation*, 80, 103304. https://doi.org/10.1016/j.jvcir.2021.103304

Luo, Z., Yang, W., Ding, Z. Q., Liu, L., Chen, I. M., Yeo, S. H., ... & Duh, H. B. L. (2011, March). "Left Arm Up!" Interactive yoga Training in Virtual Environment. *In 2011 IEEE virtual reality conference* (pp. 261-262). https://doi.org/10.1109/VR.2011.5759498

Neumark-Sztainer, D., Watts, A. W., & Rydell, S. (2018). Yoga and body image: How do young adults practicing yoga describe its impact on their body image?. *Body Image*, 27, 156-168. https://doi.org/10.1016/j.bodyim.2018.09.001

Oh, S. L., Ng, E. Y., San Tan, R., & Acharya, U. R. (2018). Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats. *Computers in Biology and Medicine*, 102, 278-287. https://doi.org/10.1016/j.compbiomed.2018.06.002

Patil, S., Pawar, A., Peshave, A., Ansari, A. N., & Navada, A. (2011, June). Yoga tutor visualization and analysis using SURF algorithm. *In 2011 IEEE Control and System Graduate Research Colloquium* (pp. 43-46). https://doi.org/ 10.1109/ICSGRC.2011.5991827

Qiao, S., Wang, Y., & Li, J. (2017, October). Real-time human gesture grading based on OpenPose. *In 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics* (CISP-BMEI) (pp. 1-6). https://doi.org/10.1109/CISP-BMEI.2017.8301910

Russell, K., Gushue, S., Richmond, S., & McFaull, S. (2016). Epidemiology of yoga-related injuries in Canada from 1991 to 2010: *A case series study. International Journal of Injury Control and Safety Promotion*, 23(3), 284-290.

Sathyanarayanan, G., Vengadavaradan, A., & Bharadwaj, B. (2019). Role of yoga and mindfulness in severe mental illnesses: A narrative review. *International Journal of Yoga*, 12(1), 3. https://doi.org/ 10.4103/ijoy.IJOY_65_17

Sitaula, C., Xiang, Y., Aryal, S., & Lu, X. (2021). Scene image representation by foreground, background, and hybrid features. *Expert Systems with Applications*, 182, 115285. https://doi.org/10.1016/j.eswa.2021.115285

Swain, T. A., & McGwin, G. (2016). Yoga-related injuries in the United States from 2001 to 2014. *Orthopaedic Journal of Sports Medicine*, 4(11), 2325967116671703. https://doi.org/10.1177/2325967116671703

Trejo, E. W., & Yuan, P. (2018, June). Recognition of Yoga poses through an interactive system with a Kinect device. *In 2018 2nd International Conference on Robotics and Automation Sciences* (ICRAS) (pp. 1-5). https://doi.org/10.1109/ICRAS.2018.8443267

Wang, J., Yu, L. C., Lai, K. R., & Zhang, X. (2016, August). Dimensional sentiment analysis using a regional CNN-LSTM model. *In Proceedings of the 54th annual meeting of the association for computational linguistics* (volume 2: Short papers) (pp. 225-230). https://doi.org/10.18653/v1/P16-2037

Wiese, C., Keil, D., Rasmussen, A. S., & Olesen, R. (2019). Injury in yoga asana practice: Assessment of the risks. *Journal of Bodywork and Movement Therapies*, 23(3), 479-488. https://doi.org/10.1016/j.jbmt.2018.09.151

Xu, Z., Li, S., & Deng, W. (2015, November). Learning temporal features using LSTM-CNN architecture for face anti-spoofing. *In 2015 3rd IAPR Asian Conference on Pattern Recognition* (ACPR) (pp. 141-145). https://doi.org/10.1109/ACPR.2015.7486482

Yadav, S. K., Singh, A., Gupta, A., & Raheja, J. L. (2019). Real time yoga recognition using deep learning. *Neural Computing and Applications*, 31(12), 9349-9361. https://doi.org/10.1007/s00521-019-04232-7

Yu, N., & Huang, Y. T. (2020). Important factors affecting user experience design and satisfaction of a Mobile health app-A case study of daily yoga app. *International Journal of Environmental Research and Public Health*, 17(19), 6967. https://doi.org/10.3390/ijerph17196967

Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A C-LSTM neural network for text classification. *arXiv preprint arXiv*:1511.08630. https://doi.org/10.48550/arXiv.1511.08630