RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt24212

Yoga Poses Identification and Classification Based on Convolutional Neural Network and Transfer Learning with Media Pipe

S.V. Shri Bharathi [a], T. Parasuraman [b], S. Akila [c,*], R. Ramakrishnan [b], K. Shivaraju [b], S. Krishnakumar [b], S.A. Sreedev [b], C. Vijayalakshmi [b], C. Vijayalakshmi [d]

[a] Department of Data Science and Business Systems, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai-603203, Tamil Nadu, India

[b] Department of Physical Education and Sports Sciences, Hindustan Institute of Technology and Science, Chennai-603103, Tamil Nadu, India.

[c] Department of Physical Education and Sports, Central University of Tamil Nadu, Thiruvarur-610005, Tamil Nadu, India

[d] Department of Statistics and Applied Mathematics, Central University of Tamil Nadu, Thiruvarur-610005, Tamil Nadu, India.

* Corresponding Author Email: akilaphyog@gmail.com

DOI: https://doi.org/10.54392/irjmt24212

Received: 12-12-2023; Revised: 17-03-2024; Accepted: 24-03-2024; Published: 30-03-2024

Abstract: Yoga is an ancient Indian discipline that promotes mental and physical well-being. It's become popular due to the stress of modern life. There are many ways to learn yoga, including studios, private instructors, and online resources. Many students of yoga struggle to identify their own mistakes when learning on their own. This article proposes a new approach for the effective identification and classification of different yoga poses using deep learning algorithms. The Media-pipe library is used to extract user-relevant features from 85 videos featuring 15 yoga practitioners doing 6 different poses. In the study, results from many deep learning models are compared, both with and without extracting features. Several different learning models achieved their best performance when fed skeletonized pictures to a neural network for training. Results from several models are compared in order to demonstrate the beneficial effect of skeletonization. With a validation accuracy of 99.9% on non-skeletonized images, Mobile-Net with CNN outperforms CNN, LSTM and SVM by a wide margin. Skeletonized images are used by the proposed model MobileNet, which achieves an accuracy result of 99.9%.

Keywords: Media Pipe, Deep Learning, Yoga Pose, Activity Recognition, Classification

1. Introduction

Today's fast-paced society has made it common for people to work out alone at home. The Physical practised known as yoga dates back about 5,000 years [1, 2]. Due to its various health benefits, yoga has become more popular among individuals of all ages [3, 4]. When it comes to yoga, the most difficult part is achieving the right poses. Yoga asanas have the potential to cause serious harm if not performed properly. Injuries and deformities of the body's framework are possible results of an incorrect position. Hence, it's understandable why many seek out the services of a teacher or trainer to check in on their development and make necessary adjustments to their asanas. In light of the fact that not everyone has access to yoga studios and instructors—especially in the midst of the current COVID pandemic—a system that uses deep learning and other forms of artificial intelligence may be developed to categorize and correct yoga postures. This sort of method may be used to categorize the many yoga asanas and provide the practitioner with useful feedback. A lot of progress has been made in the realms of science and technology in recent years. Everyone's lives have become simpler and more comfortable thanks to technological developments. Healthcare and related fields will benefit enormously from this expansion. Developments in computing power have improved every facet of civilization, from object detection to posture detection [4]. Human posture detection has come a long way in the last several decades. Recent advancements in deep learning and the rising processing power of current computers have facilitated progress in posture identification [5, 6]. Using Media Pipe, image processing, deep learning, and the transfer learning method, the authors of this research want to develop a system capable of identifying yoga poses in still images.

Around 40% of children and adolescents are overweight or obese. The above details make it clear how crucial it is to regularly engage in physical activity.

Regular exercise not only aids in weight loss, but also keeps us mobile, improves blood flow, and keeps us in excellent physical shape and mental equilibrium. A similar level of physical fitness to what we enjoyed when we regularly attended gyms. Joining a gym, yoga class, or training with a personal trainer is an investment that is out of reach for many people. Self-training is an alternative since it provides a recorded yoga sequence but no feedback to the practitioner. If we don't get accurate information on our postures, we risk injuring ourselves, which is why our initiative is so important. It is possible to ascertain a person's location at critical junctures by using pose estimation methods. This would allow us to provide feedback on estimated or evaluated human body poses. In our project, we do this by comparing the camera's incoming picture to information already saved in the database. If they're the same, we know the input image's posture is accurate; otherwise, we may use that information to guide corrections. Human posture estimation has been one of computer vision's most challenging challenges since the field's inception. In an effort to pinpoint the best effective method for real-time human posture recognition, several different strategies have been explored. Increases in computational power have led to a dramatic improvement in deep learning models, which are now the standard method for estimating body poses. Some examples of where gait analysis is put to use include biometric identification, video surveillance anomaly detection, an exercise/yoga tracking system, animation and frame interpolation, and more.

The challenge with yoga, nevertheless, is that it is just as important to practice it appropriately as it is with any other exercise, as any incorrect posture during a yoga session can be counterproductive at worst. This necessitates the presence of a trainer who can keep an eye on things and make adjustments as needed throughout the session. Given that not everyone can afford or find a yoga teacher, AI-based apps that can recognize postures and provide individual feedback on how to improve form are promising options [7]. Recent years have seen tremendous performance increases in human posture estimation because of deep learning [8]. Instead of manually dealing with the relationships between structures, deep learning algorithms provide an easier way of mapping the structure. Lift up, swiss ball hamstring curls, push up, cycling, and walking were the five exercise positions identified using deep learning [9]. Yet a more recent use is in yoga positions [7].

The ancient Indian practice of yoga is gaining popularity across the globe for its purported health benefits to the body, mind, and soul [8]. The astonishing curative benefits of yoga on a wide range of human illnesses, including respiratory disorders, heart ailments, musculoskeletal diseases, and deep learning applications in healthcare [9-13], are contributing to yoga's rising profile in the medical community. Yet, there is a generational divide when it comes to knowledge of yoga's benefits, which has contributed to the rise of several modern health problems that might be easily avoided if people made yoga a regular part of their lives. The lack of accessible teachers is a key contributor to the widespread misunderstandings about yoga, which in turn discourages individuals from practicing it. Real-time self-learning assistance that can identify different yoga postures using activity recognition techniques is one solution to the problem of not having easy access to the proper coaching that is essential to the form's desired level of popularity.

This is the outline of the paper. In Part 2, we will examine the literature about the different yoga classification schemes. The paper's approach for using current methods is described below in section 3. Section 4 presents the findings and analysis. Section 5 contains some closing observations and the report's projected future scope.

2. Related works

Many fields, including robotics and computer science, have made use of the ability to recognize human actions. Randomized trees (random forests) are used to monitor human actions using sensors, as shown in references [14, 15]. Human activity identification is accomplished with the use of hidden Markov models and identified body parts in reference [16]. An accuracy of 97.16 percent was reached using this strategy for recognising six common household tasks. This technique is employed by monitoring services at smart homes. [17] Sensing devices that detect noises are employed in conjunction with external ambient sounds for human activity detection, and 96% accuracy is attained.

Several different approaches and tactics have been devised and applied for real-time human monitoring, each with its own set of benefits and drawbacks. The proposed study intends to expand upon existing research in the field of computer vision for human posture assessment [18], using their results wherever appropriate. It was also shown [19, 20]. that LSTM neural networks operate well and contribute significantly to certain tasks. The goal of this work was to develop a deep learning-based system capable of accurately identifying yoga poses and acting as a stand-in for a personal trainer by giving the user helpful, actionable feedback. Researchers in [21] estimated yoga postures using a combination of ML and DL. A Support Vector Machine (SVM), a Convolutional Neural Network (CNN), and a Convolutional Neural Network with Long Short-Term Memory (CNN-LSTM) were used to create a framework and their respective performance was compared (LSTM). In terms of accuracy, the research discovered that the hybrid CNN-LSTM model was the most effective. An approach termed tf-pose estimation was developed in [22] to identify the user's skeleton. The results of our tests on six widely used

machine learning models are shown below. We put Decision Tree, Naive Bayes, SVM, KNN, and the other prominent ML models (Random Forest, Logistic Regression, and Naive Bayes) to the test. When deployed to a dataset consisting of 10 unique yoga poses and about 5,000 photos, the Random Forest classifier produced a success rate of 99.04%.

To create a self-training system, Chen *et al.* [22] first established a features-based method for recognizing Yoga activities. The user's body contour is extracted and a body map is captured with the help of a Kinect. Rapid skeletonization using a star skeleton was utilised to get a human stance descriptor. The research in [23] develops a Kinect-based computer-assisted self-training method for correcting poor posture. To this end, it has considered the tree pose, the warrior III pose, and the downward dog. The total accuracy, however, is just 82.84 percent. Using Kinect and Ada boost classification, the authors of [25] offer a Yoga identification system for six asanas, claiming an accuracy of 94.78%. Nevertheless, they are using a camera with a depth sensor, which may not be readily accessible to people. Indian traditional dance and Yoga postures may be identified from photographs with the use of image recognition methods like convolutional neural networks (CNNs) and stacked autoencoders (SAEs), as shown by the work of Mohanty *et al.* [26]. Nevertheless, they have only tested how well they work with static photos, not moving ones. In order to aid in correcting postures while performing Yoga, Chen et al. [24] suggested a Yoga self-training system that would use a Kinect depth camera to monitor 12 different asanas. Nonetheless, individual models are built for each asana and laborious feature extraction is used.

Deep learning is an area that has a massive amount of unrealized potential in the classification of human poses, and a significant amount of research is already being carried out using this method. A CNN-based pose estimation approach is proposed by Kim *et al.* [25] as a solution to the issue of information loss and the drifting of joints that may occur during the process of posture estimation. Deep learning-based models are used in Jose et.al [26] 's work; nonetheless, the small dataset reportedly produces subpar results. An enhanced version of the Mask Region Convolution Neural Network architecture was suggested by Wang et al. [27] in order to detect yoga movements. Despite this, the research is carried out on a limited dataset, which is insufficient to verify whether or not the model is accurate. With the use of a camera and some video data, a simple convolutional neural network was trained in the study [28] to identify 22 distinct yoga postures. Long *et al*. [29] construct a yoga posture coaching system using a number of different transfer learning models and base it on a number of different assessment measures. The research employs around twice the number of photos in the dataset to verify the conclusion, in comparison to the previous study [27]. In their paper [28-30] Liaqat *et al.* present a hybrid model that makes use of both machine learning and deep learning for the initial and final layers. The dataset used in the research only includes three fundamental positions, and the model's accuracy is poor when applied to more complicated stances. The authors provide a technique for the identification of a practitioner's posture that makes use of a deep learning convolution neural network, also known as a DLCNN. Before feeding the photos into the DLCNN network, the model performs an analysis on them using OpenPose. Cao *et al.* [31] offer a technique to simultaneously recognize the stance in the presence of many people in a video. Finding the associated body part of a person may be accomplished with the help of the suggested technique by making use of a nonparametric presentation. In addition, Narayanan *et al*. [32] offer a model for the classification of yoga postures that is based on deep learning. In this project, skeletal elements of the human posture captured in the photo frame are extracted with the help of the Open Pose architecture and stored in a NumPy array for further usage. As a model, it employs a straightforward neural network structure that consists of two hidden layers in addition to an output layer.

3. Methodology

In this research, a deep learning-based yoga pose estimation technique that is shown in algorithm 1 is suggested to recognize accurate yoga poses and offer feedback to enhance the yoga posture. Initially, the image of a yoga practitioner performing an asana was captured by a camera and fed separately to the four deep learning architectures, which then estimate the pose performed by the practitioner by comparing it with the pretrained model. If it does not match any of the five asanas, an error was shown.

The goal of this methodology is to help people improve their yoga posture. The model receives its input in the form of movies or still photos, and frames are taken at regular intervals from videos. These frames are then passed to the Keras multi-person posture estimation algorithm in order to extract critical points. The calculation of the 12 joint vectors begins with these crucial points. The angles that are formed between the x-axis and each of these 12 joints are calculated, one for each joint. The categorization model uses these angles to determine where the posture falls within the spectrum of six different yoga poses. These angles are then compared to a grid consisting of 12 different angles that make up the categorized posture. This array contains the average angles of 12 different joints that were taken from the dataset. The differences, accordingly, are computed for each angle, and recommendations are shown for each angle.

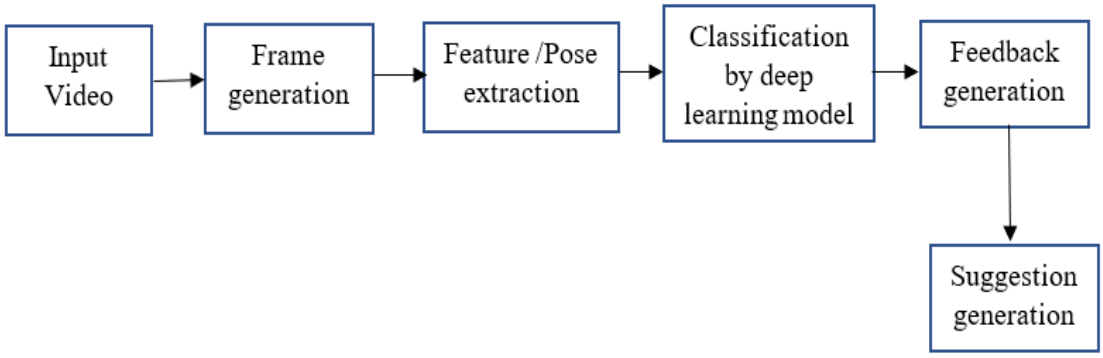


Figure 1. Schematic Representation of Suggested Methodology

As a feedback output, the direction in which joints should be rotated is determined by the sign of the difference and may either be clockwise or anticlockwise.

Figure 1 provides a schematic representation of the suggested methodology, and the accompanying text provides more explanations of each stage. The suggested system receives individual frames from video sequences that are played in real time. The end result would be a recommended yoga position that also included instructions on how to fix the appropriate angle and posture for that pose. The system is comprised of three basic components: the first is key point extraction, the second is position prediction, and the third is posture correction. During the phase of feature extraction, the goal is to detect and retrieve the location of essential key points in response to the user's current position. During the phase known as posture prediction, the structure of the model is created, and it is decided whether or not the stance is appropriate. The next step is known as "position correction," and it is when the user is given more guidance on how to improve their stance, as well as a visual depiction of the degree to which their posture fits the ideal pose.

3.1 Dataset

This project's data set is freely accessible as part of the Open-Source library. With a combined running time of 1 hour, 6 minutes, and 5 seconds, the 88 movies in this collection are certainly impressive. The videos were captured at 30 frames per second (frames per second). Every one of these clips was shot inside, at a distance of four meters. Each person has done their own unique version of yoga to help create a dataset that can be utilized to create an accurate yoga position identification algorithm. The typical video lasts between 45 and 60 seconds.

Training the 3D CNN model using the six-pose Yoga datasets of Yadav *et al.* [7] similarly used a similar approach to preparing the datasets. In order to train the model, we collected 4930 Yoga posture clips from 60% of the movies at random and used their assessment technique to determine the remaining 40%. 1643 clips were taken from an additional 20% of the movies in the sample and utilized for hyperparameter tweaking. In the end, 1644 clips were taken from the remaining 20% of Yoga video content, and the trained model's performance was assessed based on those results. Figure 2 shows the sample yoga poses from the dataset.

3.2 Pre-processing

Each category of yoga asanas has its own unique data structure in the yoga asanas dataset. Aligning the data makes it more suitable for the next steps in the deep learning pipeline. The data goes through three distinct pre-processing phases as seen in the picture. Because of the diverse origins of the data, their size will vary.

As a first stage of preparation, resizing to 100x100 pixels is applied to data with varying dimensions. After that, we use a Gaussian filter to smooth out the data while maintaining its original sharpness and detail. Finally, the Histogram Equalization is applied to the final picture to normalise the intensity distribution and boost the image's contrast.

First, we offer an algorithmic pipeline for Yoga position detection that makes use of multiple open-source tools, including OpenCV and NumPy, during the database construction stage. Yoga stance frames collected by a camera are processed in real time to meet the input requirements of the developed 3D CNN model. Nevertheless, this phase is handled offline during the construction of the database for the pre-recorded Yoga posture videos. Here, we take each of the Yoga videos in the dataset and rip out many 16-frame vignettes showing various positions. In addition, as the resolution of the input Yoga stance videos might vary, we first extract an input picture with dimensions of 650 × 650 × 3 pixels from each frame of the video clips, and then we scale the recovered image to a frame resolution of 112 ×112 × 3 pixels. Including temporal and spatial jittering in this way also helps reduce overfitting during network training. Normalizing the input data is a common pre-processing step in deep learning used to make the intensity levels of picture pixels consistent across the board.

Figure 2. Sample Yoga poses from the dataset

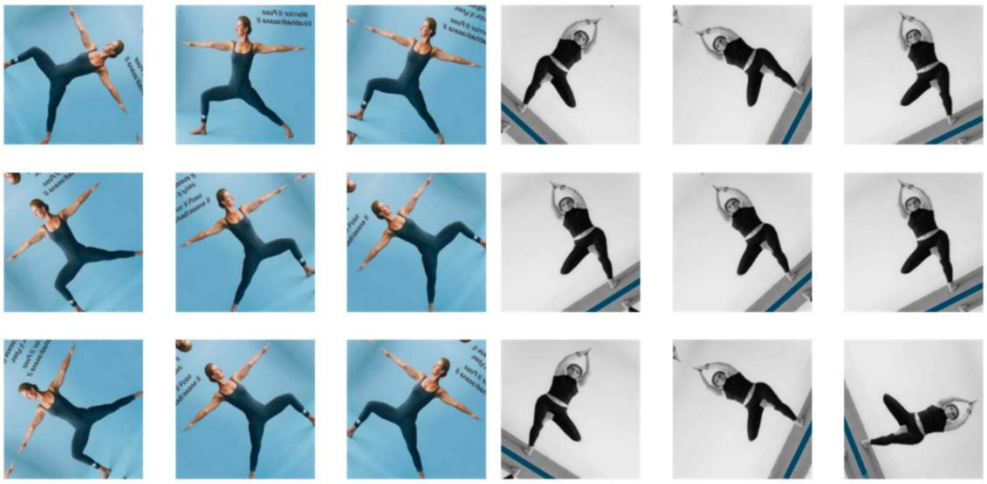


Figure 3. Augmented images from the dataset

The convergence qualities of neural networks may be improved by doing this procedure before training and testing the network topologies. Improved recognition accuracy and generalizability may also be achieved in inference by normalising test input data with the derived parameters from the training set. This is why, before training our proposed neural network design, we standardized the frame rates of the Yoga posture videos. This was accomplished using a two-stage process: first, the frames were split by a factor of 255 to put the pixel intensities in the range of 0-1, and then the two-step process was applied. After that, we subtracted the channel-wise pixel intensities from the channel-wise mean intensities calculated on the whole dataset to get a mean normalization of the scaled frames. Through this process, we can ensure that the network is being trained with consistent input frames from all of the videos.

In addition, we apply a variety of on-the-fly data augmentation tactics as shown in the figure.3 to enhance the model's generalization capacity and get over the limits imposed by overfitting while training parameter-heavy 3D CNN models on a limited sample Yoga position dataset. Then, we randomly flip the horizontal orientation of each frame in the training set's video clips.

To create the rotated pictures, step two involves randomly rotating each frame by an angle between -11 and 11. (empirically determined). Frames are then randomly magnified, sheared, and moved in both the horizontal and vertical planes in the third stage. After the pre-processing phase, we acquire 3608 clips of 16 frames for training, testing, and validation sets, divided 80:10:10, correspondingly, utilizing clips with different subjects for each.

3.3 Proposed Methodology

The first step is to extract key points from each frame of the movie, and then save those key points in JSON format thereafter. This step is the beginning of the process. Shoulders, elbows, wrists, knees, and other parts of a person's body are all examples of key points that are important in the development of a yoga posture. Key points are also known as anchor points. For the purpose of key points extraction, we made use of the Media Pipe library, which is a Google-developed cross-platform library that offers outstanding machine learning solutions that are already prepared for use in computer vision applications. At this step, a highly tuned and pre-trained CNN model is used for high-fidelity body posture tracking. This model determines 33 3D landmarks and background segmentation masks on the full body based on RGB video frames. The Media pipe library provides three coordinates—X, Y, and Z—where Z represents the depth of a two-dimensional value. Figure 4 shows the effect of using the Media Pipe library to extract the key points from the data.

After converting the movies to the JSON format, the dataset is being divided into three parts: the train dataset, the validation dataset, and the test dataset. The split ratio that is being used is 64:16:20, and each test case is made up of a series of 45 frames with an overlap size of 36 frames. These frames include the coordinates of all 33 key points. The following is an example of how the input shape of a single test case should look: (45, 33, 2). The combined numbers of samples used for training, validation, and testing come to 7063, 1832, and 2202. At this phase, methods for machine learning are put to use in order to successfully construct data structures within the context of the particular application. The contribution resides not only in the efficient creation of the system but also in the application of the assessing procedures.

3.4 MobileNetV2

When it comes to solving the categorization issue, researchers have turned to a Deep Neural Network called MobileNetV2. TensorFlow's ImageNet pre-trained weights were used. The first set of layers is then frozen to preserve previously learned characteristics. Then, we add additional trainable layers, which are then trained on the amassed dataset to learn the characteristics that distinguish a face with a mask from an unmasked one. The simulation is then tweaked and the weights are stored. If you can find a model that has already been trained, you can save time and money by using it. You can also benefit from the model's biased weights without having to discard any previously learned features. The following layers and operations are used in the MobileNetV2 deep learning model, which is based on a Convolutional Neural Network. Figure 5 displays the internal organization of MobileNetV2. The Convolutional Neural Network's base layer. When we say "convolution," we're referring to a mathematical process wherein we combine two functions to get a third. In order to extract features from a picture, it uses a sliding window method. The creation of feature maps is aided by this. Convolution of mainly two matrices, the input image matrix X and the convolutional kernel Y, yields the output matrix Z as

$$Z(t) = (X * Y)(a) = \int_{-\infty}^{\infty} X(t) * Y(t-a)dt \quad (1)$$

Pooling procedures, when applied, may allow for a decrease in the size of the input matrix while maintaining the majority of the matrix's characteristics. This can result in quicker computation times. By excluding some random neurons with a bias from the model, the dropout layer mitigates the risk of overfitting, which might occur during the training process. These neuronal connections may be found at both the visible and unseen levels of the brain.

Figure 4. Sample key points extracted from the data

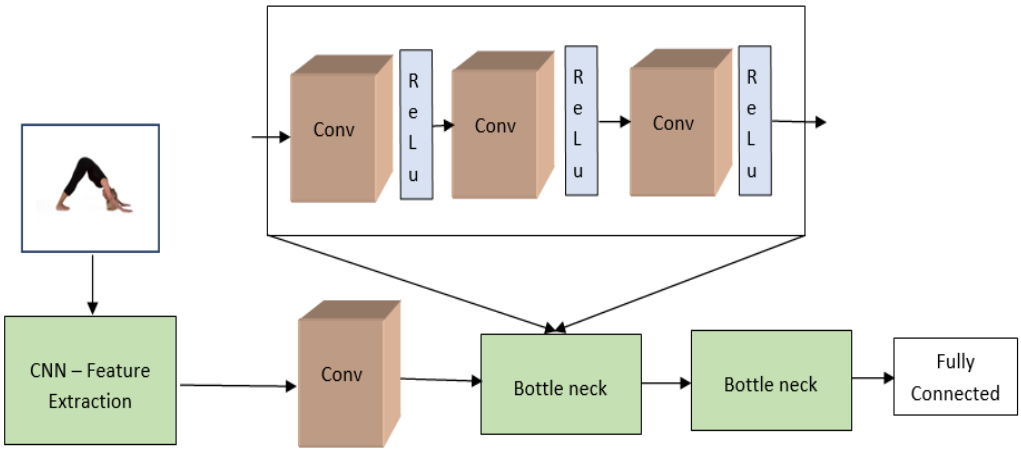


Figure 5. MobileNetV2 architecture for Yoga pose estimation

Altering the dropout ratio in a neuron population may cause a change in the probability that a neuron will be lost.

3.5 Efficient Net

We included a global average pooling2d layer into the EfficientNetB4 model for transfer learning to help reduce the risk of overfitting by decreasing the number of parameters. We have incorporated three dense layers within the inner layers of the network, each accompanied by a dropout layer and activated by the RELU function. To mitigate overfitting, a random dropout rate of 30% was implemented. The suggested automated detection system is made up of a single dense layer with multiple output units engaged by a softmax activation function for classification tasks and three target output units triggered by the same function for multi-class recognition. The proposed model's layers are laid out in sequence, together with their output shapes, the number of parameters (weights) in each layer, and the overall number of parameters (weights). There are a total of 170,603 parameters.

The whole stack of programs and libraries that will be utilized to carry out the planned work is freely available to the public. All that's required for readers to replicate the findings is a copy of the Google Colab Notebook and the GPU run-time type. As Google donates this program for use in research with a 12 GB Tesla K80 GPU, there are no financial barriers to entry for anyone interested in utilising it. Transfer learning may be used to image classification issues with the help of the EfficientNet Models, which are pre-trained, scaled CNN models. In May of 2019, Google AI released the model to the public through GitHub. Keras's Image Data Augmentor is a specialised picture data generator that allows for the addition of augmentation modules. Furthermore, you may find it on the Google AI-maintained GitHub repository. To round things off, Google AI has created the Augmentations library, which can be downloaded from the Github repositories. In conclusion, as it is open source and free, there are no licensing restrictions on using the programme. The algorithm 1 depicts the procedure for yoga pose identification and correction.

Algorithm 1. Algorithm for yoga pose identification and correction
Input: Video $V_D = \{V_1, V_2, \ldots\ldots\ldots V_i\}$
Output: Pose estimation and correction
1. **Load** the input video V_D
2. **Extract** the frames from video **For** each i in V_D $F_i = \sum_{i=1}^{n} f_i$ **End for**
3. **Apply** augmentation on each frame
4. **for** each frame in F_i Extract the feature or pose from frames using CNN
5. **pass** the features to MobileNet $Z(t) = (X * Y)(a) = \int_{-\infty}^{\infty} X(t) * Y(t-a)dt$
6. **calculate** the angle $\emptyset = tan^{-1}\left(\frac{a_2-a_1}{b_2-b_1}\right)$ $cos\theta = \frac{X \cdot Y}{\|\|X\|\|Y\|\|}$
7. **Generate** the feedback for the identified pose

3.6 Pose Correction

After the similarity % (using cosine similarity) has been determined, it will be shown to the user once the projected posture has been assessed as accurate with regard to the specified pose. Next, we'll explain the six yoga positions that comprise the dataset, and we'll show you how we figured out the most crucial angles and developed guidelines for each one. A threshold is established for each regulation, which represents the greatest departure from the ideal stance that the user is permitted. The user receives both visual and auditory alerts if they go above the predetermined limit. By taking the tangent inverse of the positive X-slope, we may get the angle between two control points. Given the two

coordinates of the reference points, the equation represents the formula for calculating the angle.

$$\emptyset = tan^{-1}\left(\frac{a_2-a_1}{b_2-b_1}\right) \quad (2)$$

The user provides input in the form of text, which is then translated into voice using the text-to-speech converter (Pyttsx3), which may be used both online and offline. The cosine similarity is also presented to the user, which is a metric that compares two vectors by computing the cosine of angles between them. This metric's value may go as low as 1 or as high as +1. The similarity score is multiplied by 1 to the positive side if it is between 1 and 0. The resemblance is then determined by the score, which may vary from -1 to +1. Cosine similarity is defined by the formula provided in the below equation (3),

$$cos\theta = \frac{X \cdot Y}{|||X||Y|||} \quad (3)$$

Two vectors, X and Y, in a higher dimensional space. The cosine similarity between the user's posture's landmarks and the reference pose is determined here. By comparing the two, you can see how similar the stance is to the real one. Keypoints are first adjusted to determine a common scale for all users, since their distances from the camera may naturally vary.

4. Result and Discussion

A model's performance may be measured in terms of its precision, which is a metric. To put this into a more concrete context, it is defined as the percentage of samples that are properly identified as positive given the total number of samples that are, in fact, positive. The model's results show that it has a precision of 0.99. The mathematical model for it may be found in below equations (3),

$$Precision(P) = \frac{T_{ps}}{T_{ps}+F_{ps}} \quad (4)$$

$$Accuracy(A) = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

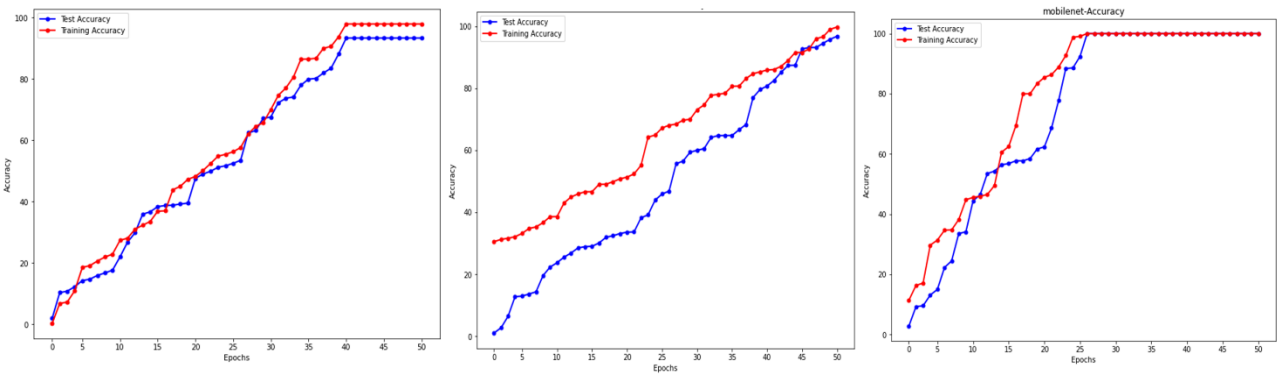
$$Error\ Rate = \frac{FP+FN}{TP+FP+FN+TN} \quad (6)$$

$$F\ Measure = \frac{2+Recall*Precision}{Recall*Precision} \quad (7)$$

T_{ps} represents the true positive and F_{ps} represents the False positive. We have employed not only recognition accuracy but also precision, recall, and F1-score to evaluate the built deep learning architecture as precisely as possible. The accuracy of a prediction is measured by how many positive observations were really right out of the total number of predictions. Correctly predicted positive observations as a percentage of total observations in the actual class is called "recall." The model was running for a total of 50 iterations, and the results showed that the training accuracy was 99.99%.

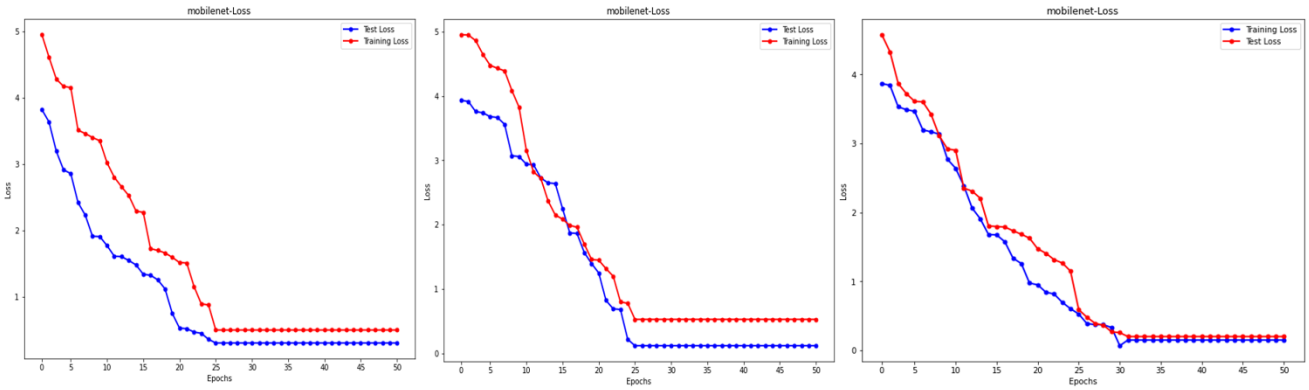
The graphical depiction of both model accuracy and model loss may be seen in Figure 7 and 8.

After training for a total of 100 epochs, the system achieves an accuracy of 100% on the training examples and 99.9% on the validation data respectively. The overall test reliability for the system is 99.84%, as measured for each frame. These curves, also known as information absorption expectations, are often used in models that learn gradually over time, such as neural networks. They discuss evaluations of information gathering and approval processes, which offer us an idea of the model's capacity for learning and summing up. To have a lower score indicates better model execution as implied by the model misfortune bend. The accuracy curve of the model indicates an increasing score (exactness), which means that a higher score indicates a more successful model implementation. When the preparation and approval misfortune both falls, stabilise, and have a small gap between their final misfortune values, we may say that the model's fit is good. In contrast, a well-fitting model exactness curve has a base hole between the final exactness values and an upward trend in both the training and approval precision.



(a) MobileNetV2 (b) EfficientNet (c) CNN+MobileNet

Figure 6. Accuracy of the models



(a) MobileNetV2 **(b)EfficientNet** **(c) CNN+MobileNet**

Figure 7. Loss of the models proposed in this paper

Table 1. Performance metric of the CNN + Mobilenet

	Precision	Recall	F1-Score
Chair	0.98	0.98	0.98
Cobra	0.99	0.99	0.99
Downdog	1.00	1.00	1.00
Goddess	1.00	0.99	1.00
Tree	1.00	0.99	0.99
Warrior	0.99	1.00	1.00
Accuracy			1.00
Macro Avg	1.00	0.99	1.00
Weighted Avg	1.00	0.99	1.00

Table 2. Accuracy comparison of the models

	Accuracy	Recall	F1-Score
CNN	0.95	0.94	0.95
LSTM	0.96	0.96	0.97
SVM	0.85	0.84	0.85
VGG16	0.96	0.96	0.97
MobileNet	0.97	0.97	0.98
EfficientNet	0.98	0.97	0.97
CNN + MobileNet	1.00	0.99	1.00

A combination of NumPy, OpenCV, and PIL. The MobileNet model was employed with a magnet of training weights for the transfer learning phase. After down-sampling all the photos to 100x100, 4608 features were taken from them. There is a complete separation of the dataset into a training set and a testing set. In order to create a reliable classifier, data from the training set must be utilised, while data from the test set must be used to measure the classifier's performance. We achieved 99.9% accuracy in our tests, and the confusion matrix we generated by comparing the true labels to the ones we predicted is shown in figure 6, while the computed values for other metrics, including precision, recall, and fi score, are provided in table 1. In Table2, we can see that the suggested transfer learning architecture outperforms the conventional Architectures and other deep learning algorithms trained using CNN Moment.

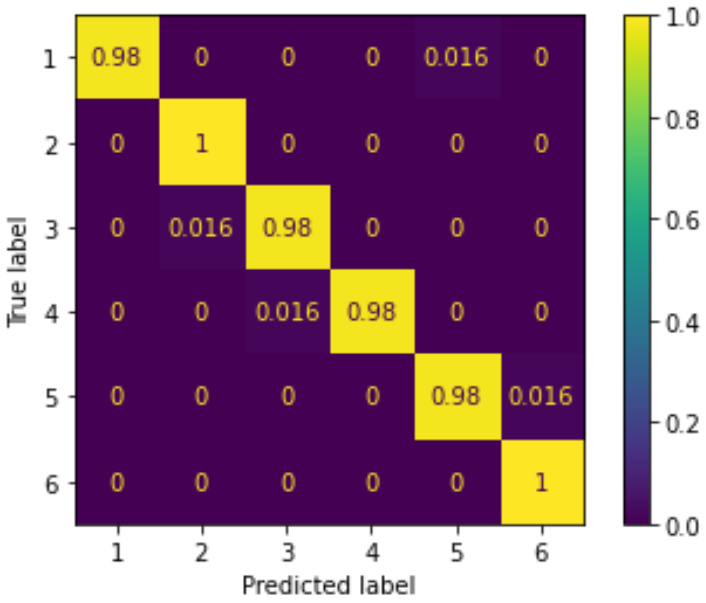


Figure 8. confusion matrix

Results of the model's predictions on the test set of the internally developed Yoga position dataset are shown in Figures 8 in units of the normalised confusion matrices. High density along the confusion matrix's diagonal indicates that the model accurately predicted the majority of Yoga posture videos.

It is crucial to provide the user with feedback so that they may learn from their mistakes. This aids the user in acquiring the proper posture for the yoga position and perfecting their practise. The user receives immediate visual and auditory feedback on their performance. The user will be alerted when their deviation is more than the predefined value. The user may then alter his or her yoga posture accordingly after seeing the correction. The notification may appear on the screen or be played over the headphones. To avoid the necessity for the user to tilt their head to view the on-screen text. Also, the user may be facing away from the screen due to their yoga position, making it difficult or impossible to comprehend the information being shown. The user may get the feedback message about their posture using a Bluetooth-connected headset or speaker. Users' degrees of pliability vary; some may be less flexible than others. Thus, a threshold parameter that may be adjusted by the user is provided for dealing with this issue. To meet their own needs, users may adjust the threshold value. To allow for a variance of around 20 degrees in either way, a new user may set the threshold to, say, 20 degrees. In order to get the most out of their practise, experts might reduce the angle down under 10 degrees. As a result, this function makes it possible for users of all experience levels to gradually and steadily increase their physical flexibility for the practise of yoga.

5. Conclusion and future work

Developments in science and technology provide the way for interdisciplinary exploration. Artificial intelligence, machine learning, and computer vision are just a few examples of the cutting-edge technologies used to power today's practical, constantly evolving, and oftentimes live-streaming applications. The mind-body practise of yoga is gaining popularity worldwide. We have suggested a computerized yoga posture identification scheme from images or videos as a starting point for developing a system to help humans practise yoga with a virtual trainer. When developing this system, we tried using state-of-the-art image classification techniques like convolutional neural networks (CNN), but the dataset was too small. We have used a CNN classifier and transfer learning with a MobileNet architecture and primarily targeted MobileNet weights to improve upon previous results. The outcomes were encouraging, with an 99.9% rate of correct predictions. Plenty of untapped potentials exists in this space. Yoga asanas can be validated not just through visual inspection, but also through video analysis of the corresponding movements.

Six yoga asanas are presently categorised using the suggested models. There are many different yoga asanas, making it difficult to develop a position estimate model that works for all of them. More yoga positions done by people both inside and outdoors would enrich the current dataset. Model accuracy is dependent on the accuracy of OpenPose posture estimation, which may fail in situations when several persons or body parts overlap. This system may be used to train itself and provide predictions in real time on a portable device. This paper illustrates the feasibility of activity recognition in real-world settings. Pose identification in domains as

diverse as sports, surveillance, healthcare, etc., might benefit from a method similar to this. There is a lot of room for exploration in the realm of multi-person posture estimation, which is a novel issue in and of itself. Several situations need for more than just one person's stance to be estimated; for example, pose estimation in crowded settings requires the monitoring and identification of the poses of many different individuals. Multi-person pose estimate is difficult because of the many aspects, such as backdrop, illumination, overlapping figures, etc., that have already been mentioned in this study.

References

[1] I.N. Sukarsa, Asana yoga meditation as a spiritual development ananda marga ashram denpasar (perspectives theology hindu). Vidyottama Sanatana: International Journal of Hindu Science and Religious Studies, 2(2), (2018) 301-306. https://doi.org/10.25078/ijhsrs.v2i2.632

[2] Puja Yatham, Supritha Chintamaneni and Sarah Stumbar, Lessons From India: A Narrative Review of Integrating Yoga Within the US Healthcare System, Cureus, 15(8) (2023) e43466. https://doi.org/10.7759/cureus.43466

[3] S. Jain, A. Rustagi, S. Saurav, R. Saini, S. Singh, Three-dimensional CNN-inspired deep learning architecture for yoga pose recognition in the real-world environment. Neural Comput Appl 33(12), (2021) 6427–6441. https://doi.org/10.1007/s00521-020-05405-5

[4] N. Zeng, P. Wu, Z. Wang, H. Li, W. Liu, X. Liu, A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. IEEE Transactions on Instrumentation and Measurement, 71, (2022) 1–14. https://doi.org/10.1109/TIM.2022.3153997

[5] S. Malek, S. Rossi, Head pose estimation using facial-landmarks classification for children rehabilitation games. Pattern Recognition Letters, 152, (2021) 406–412. https://doi.org/10.1016/j.patrec.2021.11.002

[6] A. Lamas, S. Tabik, A.C. Montes, F. Pérez-Hernández, J. García, R. Olmos, F. Herrera, Human pose estimation for mitigating false negatives in weapon detection in video-surveillance. Neurocomputing, 489, (2022) 488-503. https://doi.org/10.1016/j.neucom.2021.12.059

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

[8] U. Rafi, B. Leibe, J. Gall, I. Kostrikov, An Efficient Convolutional Network for Human Pose Estimation. In BMVC, 1, (2016) 1-11. http://gall.cv-uni-bonn.de/download/jgall_posecnn_bmvc16.pdf

[9] S. Haque, A. Rabby, M. Laboni, N. Neehal, S. Hossain, ExNET: deep neural network for exercise pose detection. Recent Trends in Image Processing and Pattern Recognition, 1035, (2019). https://doi.org/10.1007/978-981-13-9181-1_17

[10] I. Stephens, Medical Yoga Therapy, Children. 4(2), (2017), 12. https://doi.org/10.3390/children4020012

[11] S. Newcombe, The Development of Modern Yoga: A Survey of the Field. Religion Compass, 3, (2009) 986–1002. https://doi.org/10.1111/j.1749-8171.2009.00171.x

[12] C. Woodyard, Exploring the therapeutic effects of yoga and its ability to increase quality of life. International journal of yoga, 4(2), (2011) 49-54.

[13] S. Dash, B.R. Acharya, M. Mittal, A. Abraham, A. Kelemen, (2020) Deep Learning Techniques for Biomedical and Health Informatics. Springer Nature, Switzerland

[14] C.C. Hsieh, B.S. Wu, C.C. Lee, A distance computer vision assisted yoga learning system. Journal of Computers, 6(11), (2011) 2382–2388.

[15] M.T. Uddin, M.A. Uddiny, (2015) Human activity recognition from wearable sensors using extremely randomized trees. in Proceedings of the 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), IEEE, London, UK, https://doi.org/10.1109/ICEEICT.2015.7307384

[16] A. Jalal, N. Sarif, J.T. Kim, T.S. Kim, Human activity recognition via recognized body parts of human depth silhouettes for residents monitoring services at smart home. Indoor and Built Environment, 22(1), (2013) 271–279. https://doi.org/10.1177/1420326X12469714

[17] Y. Zhan, T. Kuroda, Wearable sensor-based human activity recognition from environmental background sounds. Journal of Ambient Intelligence and Humanized Computing, 5(1), (2014) 77–89. https://doi.org/10.1007/s12652-012-0122-2

[18] R. Josyula, S. Ostadabbas, (2021) A review on human pose estimation. arXiv preprint. https://doi.org/10.48550/arXiv.2110.06877

[19] A. Kanavos, F. Kounelis, L. Iliadis, C. Makris, Deep Learning Models for Forecasting Aviation Demand Time Series. Neural Computing and

Applications, 33, (2021) 16329–16343. https://doi.org/10.1007/s00521-021-06232-y

[20] A. Lyras, S. Vernikou, A. Kanavos, S. Sioutas, P. Mylonas, Modeling Credibility in Social Big Data using LSTM Neural Networks. In Proceedings of the 17th International Conference on Web Information Systems and Technologies (WEBIST), 1, (2021) 599–606. https://doi.org/10.5220/0010726600003058

[21] Y. Agrawal, Y. Shah, A. Sharma, Implementation of Machine Learning Technique for Identification of Yoga Poses. In Proceedings of the 9th IEEE International Conference on Communication Systems and Network Technologies (CSNT), IEEE, India. https://doi.org/10.1109/CSNT48778.2020.9115758

[22] S. Kothari, Yoga Pose Classification Using Deep Learning. Thesis, San Jose State University, San Jose, CA, USA, 2020.

[23] H.T. Chen, Y.Z. He, C.C. Hsu, C.L. Chou, S.Y. Lee, B.S.P. Lin, (2014) Yoga posture recognition for self-training. Lecture notes in computer science, 8325, 496–505.

[24] H.T. Chen, Y.Z. He, C.L. Chou, S.Y. Lee, B.S.P. Lin, J.Y. Yu, (2013) Computer-assisted self-training system for sports exercise using kinects. IEEE International Conference on Multimedia and Expo Workshops (ICMEW), IEEE, USA. https://doi.org/10.1109/ICMEW.2013.6618307

[25] Y. Kim, D. Kim, A CNN-based 3D human pose estimation based on projection of depth and ridge data. Pattern Recognition, 106, (2020) 107462. https://doi.org/10.1016/j.patcog.2020.107462

[26] J. Jose, S. Shailesh, Yoga asana identification: a deep learning approach. IOP Conference Series: Materials Science and Engineering, 1110(1), (2021) 012002. https://doi.org/10.1088/1757-899X/1110/1/012002

[27] H. Wang, Neural network-oriented big data model for yoga movement recognition. Computational Intelligence in Image and Video Analysis, 2021, (2021) 4334024. https://doi.org/10.1155/2021/4334024

[28] J. Kutálek, K. Kutálek (2021) Detection of Yoga Poses in Image and Video. Brno Fcaulty University of Information and Technology, 1-10.

[29] C. Long, E. Jo, Y. Nam (2022) Development of a yoga posture coaching system using an interactive display based on transfer learning. The Journal of Supercomputing, 78(4), 5269–5284. https://doi.org/10.1007/s11227-021-04076-w

[30] S. Liaqat, K. Dashtipour, K. Arshad, K. Assaleh, N. Ramzan, A hybrid posture detection framework: integrating machine learning and deep neural networks. IEEE Sensors Journal, 21(7), (2021) 9515–9522. https://doi.org/10.1109/JSEN.2021.3055898

[31] Z. Cao, G. Hidalgo, T. Simon, S.E. Wei, Y. Sheikh, OpenPose: realtime multi-person 2D pose estimation using part afnity felds. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1), (2021)172–186. https://doi.org/10.1109/TPAMI.2019.2929257

[32] S.S. Narayanan, D.K. Misra, K. Arora, H. Rai, (2021) Yoga pose detection using deep learning techniques. In Proceedings of the International Conference on Innovative Computing & Communication (ICICC), SSRN. https://dx.doi.org/10.2139/ssrn.3842656

Acknowledgement

One of the authors, S. Akila, gratefully acknowledges the financial support provided by the Indian Council for Social Science Research (ICSSR), Government of India, for facilitating the major research project (ICSSR)[02/44/2022-23/ICSSR/RP/MJ/OBC].

Authors Contribution Statement

S.V. Shri Bharathi – Conceptualization, methodology, software, formal analysis, writing-original draft; T. Parasuraman – Formal analysis, writing-original draft; S. Akila – Conceptualization, validation, supervision, writing-original draft; R. Ramakrishnan – writing – review editing; K. Shivaraju– writing – review editing; S. Krishnakumar – writing – review editing; S. Sreedev– writing – review editing; C. Vijayalakshmi– writing – review editing; C. Vijayalakshmi– Data curation, writing – review editing. All the authors read and approved the final version of the manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

Data obtained during the study are available from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes

About the License