Yoga Pose Detection System

Prof. Anita Devkar[1]
Department of Information Technology
Pimpri Chinchwad College of Engineering
Pune, India
anita.devkar@pccoepune.org

Shrinivas Nagargoje[2]
Department of Information Technology
Pimpri Chinchwad College of Engineering
Pune, India
shrinivas.nagargoje19@pccoepune.org

Pranav Tapdiya[3]
Department of Information Technology
Pimpri Chinchwad College of Engineerin
g Pune, India
pranav.tapdiya19@pccoepune.org

Adesh Shinde[4]
Department of Information Technology Pimpri Chinchwad College of Engineering
Pune, India
adesh.shinde19@pccoepune.org

Om Shinde[5]
Department of Information Technology Pimpri Chinchwad College of Engineering
Pune, India
om.shinde19@pccoepune.org

Abstract

This paper presents a method that uses deep learning algorithms to recognize different yoga postures. A dataset containing six yoga asanas—namely Bhujangasana, Padmasana, Shavasana, Tadasana, Trikonasana and Vrikshasana—was presented to the public using 15 people (ten men and five women) using a standard RGB webcam. It is proposed to use a hybrid deep learning model that combines convolutional neural networks (CNN) and long-term memory (LSTM) for well-known real-time videos. The keyframes obtained from OpenPose are processed by a CNN layer to extract features, while an LSTM layer provides temporal prediction. To our knowledge, this is the first study that uses a deep learning pipeline to recognize yoga in videos. After selection estimation, the algorithm achieves a test accuracy of 99.04% for one frame and 99.38% for 45 movie frames. A model using temporal data uses information from previous frames to produce accurate and reliable results. Five teams of 12 people (five men and seven women) were used to test the system in real-time, and we achieved an accuracy rate of 98.92%. The results of the experiments suggest a qualitative assessment of the approach and a comparison with current best practices. A dataset containing six yoga asanas including Bhujangasana, Padmasana, Shavasana, Tadasana, Trikonasana and Vrikshasana has been created and is publicly available. 15 people (ten men and five women) use a standard RGB webcam. It is proposed to use a hybrid deep learning model that combines a convolutional neural network (CNN) to recognize OGs from video in real-time.

Keywords- DeepLearning, YogaPose detection, CNN, Openpose, Posenet etc.

1. Introduction

Yoga is a practice that dates back thousands of years and has its roots in India. It is intended to balance the mind and body. It was originally mentioned in the Rigida, and there are connections to its evolution that date back even further than 3000 B.C. It is described as the "remover of misery and destroyer of pain" in the Bhagavad Gita. The practice was severely affected by the previous pandemic because people lacked access to yoga teachers, and given the physical and mental health conditions around the world, it is important to incorporate yoga into everyday life. However, due to their limitations, children with severe chronic diseases find it difficult to adopt this lifestyle.

A study was done on 70 kids, ages 10 to 15, from a school that specializes in serving mentally challenged students. While the other half continued with their regular education, the other half of them were forced to perform a specific set of yogic training exercises for 5 hours a week for 12 weeks. Studies revealed that children who practiced yoga experienced a significant improvement in their IQ [1].This demonstrated that practicing yoga exercises can be a potent therapeutic tool in the conditional improvement of mentally handicapped children. Currently, there is an increase in the use of technology for tracking health, including the use of sensors to monitor health parameters like heart rate and even sleep and the use of cameras with machine learning in various sports to measure a player's and an object's speed, such as in football, the javelin throw, the disc throw, etc.We will be employing Mediapipe's Blazepose, a high-fidelity body pose model that can allow monitoring of demanding activities like dance, yoga, etc., to gather the data and carry out the pose estimation.

2. Literature Survey

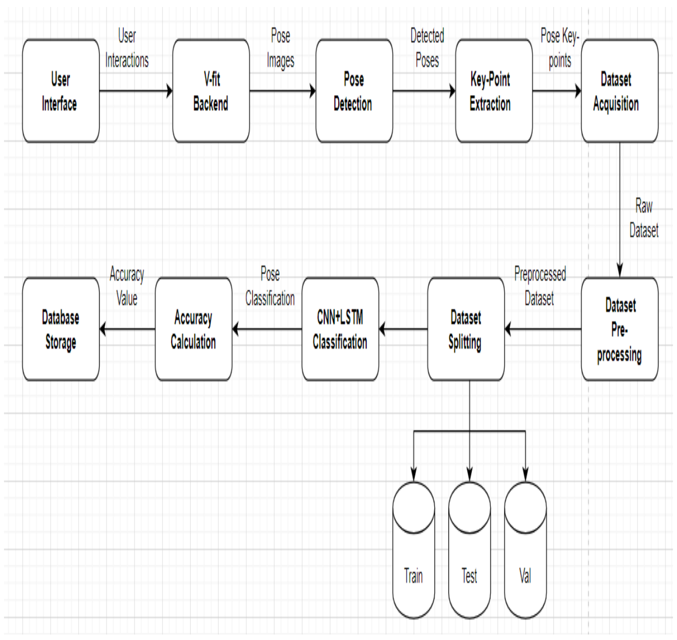
Human posture estimation has been done using various works that have been published in the literature. [1]–[7]We focus on estimating the position of one individual for our specific project. To date, little research has been done on yoga posture recognition. We reviewed a number of online study publications to go into greater detail. Many people have tried to advance this field of study.A 2020 study called Infinity Yoga Tutor focuses primarily on identifying yoga postures and correcting incorrect posture. [1] Keypoint identification is done using the OpenPose Library, which accurately recognizes the basic keypoints of the human body for pose detection. After the keypoint is detected, another position estimate is made. The models are trained using more than 100 iterations on computers with powerful graphics and lots of RAM. [1] Recent developments in computer vision and machine learning have enabled automated techniques to identify human poses. Researchers have developed a number of techniques to identify poses, including a combination of many machine learning (ML) algorithms, the use of 3D data, and the use of depth data.

They have other methods.Hash-based learning is used by Miss Yoga, a smartphone yoga assistant, to extract human poses using pressure sensors.[2] For pose detection, the widely used Posenet library, which is comparable to OpenPose, is used. Cobra, Tree, Mountain, Lotus and other publicly available datasets are used to train the model. With respect to SVM, KNN and Random Forest, the accuracy of the model is poor. [2]Girija Chiddarwar presented the use of MS Bing browser to get images for training 82 different poses. [3] A hierarchical designation is used for yoga poses based on the body configuration of the pose. Poses are categorized based on different body positions like standing, sitting, balancing, inversion etc. This category makes it easy to identify yoga poses.Utilizing advances in machine learning and deep learning, it replaces the previous skeleton-based Pose determination method. A dataset of 400 to 900 photos for each yoga pose was used by Yash Aggarwal and Yash Shah in their study implementing machine learning to identify yoga poses. Each image has been resized to 500 500 resolution for accurate results. [4]

In the post Yoga Pose Recognition Using Real-time Human Joint Point Detection Using Microsoft Kinect, Yoga Pose Detection Using MS Kinect and Its Cross Validation Against Real Data. However, the system is only able to detect three main positions and provides less accuracy when many positions are involved. [6] Santosh Kumar Yadhav presented the use of OpenPose, LSTM and CNN to detect yoga poses. OpenPose is a library that helps detect human poses using keypoints. LSTM helps analyze changes over time and remember patterns. Using LSTM in the system makes the system more robust and minimizes large-scale errors. CNN helps keep focus on patterns that are similar and record them for better position detection. [7]

3 .Proposed System

Figure 1. *System architecture*



Real-time frames from a video series are provided using the mentioned technique. The result would be the anticipated yoga posture, along with any advice for modifying the angle or pose. The system's three main stages are Key Points extraction, Pose prediction, and Pose correction. The key points extraction phase locates and extracts the locations of significant keypoints based on the user's position[20]. The posture prediction step, which also decides whether or not the position is appropriate, defines the model architecture. The final stage, stance correction, gives the user additional input for stance correction and displays the similarity percentage to the actual pose. Figure 1 depicts the suggested system design with each of the aforementioned phases.

4 . Methodology

4.1 Openpose:

OpenCV is an open-source library that can be used with a wide range of algorithms and frameworks and that offers a common infrastructure for computer vision and machine learning activities. For pose estimation, pose detection, and collecting the 3D coordinates of the pose in our scenario.

4.2 Pandas

A quick and effective indexing and data manipulation Python package. Pandas is a programme that can read and write data in a variety of forms, including CSV, text files, etc. Pandas were utilized in our project.

4.3 Scikit-Learn

Numerous pre-built machine learning algorithms, such as random forest, logistic regression, and linear regression, are available in the Scikit-learn Python toolkit. In our investigation, the stances were categorized using four built-in machine-learning algorithms.

4.4 Streamlit

An open-source app framework called Streamlit is used to build GUI online applications. In our project, we deployed our machine learning model using streamlit.

5. Workflow

5.1 Data Collection

Consequently, we have gathered particular photos from two datasets, Yoga-82. A New Dataset for Classifying Human Pose and Yoga Pose Fineness Dataset: a collection of images showing popular yoga stances.Pose requirements have been retained, while non-required postures have been deleted.The following poses are performed by different people in the dataset, and most of them are seen in images taken from different perspectives and at a distance of 4 to 5 meters from the subject. Figure 2 below shows some images from the dataset before processing.

Figure 2. *Images from Yoga-82 dataset*

In order for the final version to function well in every scenario and provide improved accuracy, images had to be captured in a variety of settings, including both open and closed locations.

Table 1. Description of dataset

Sr. No.	Asanas(poses)	No. of Images
1.	Ardha Chakrasana(Half-Wheel Pose)	385
2.	Bhujangasana(Cobra Pose)	355
3.	Padahastasana(Hand to foot Pose)	515
4.	Setu Bandhasana(Bridge Pose)	375
5.	Tadasana(Palm Tree Pose)	385
6.	Sava Asana(Corpse Pose)	385
7.	Dhanurasana(Bow Pose)	380
8.	Trikonasana(Triangle Pose)	420

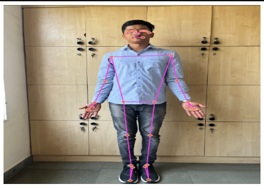
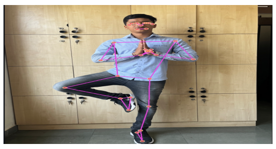
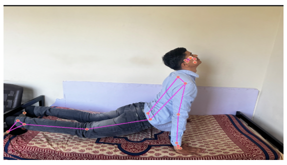
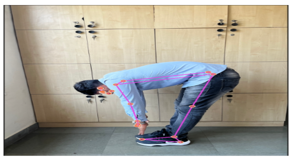
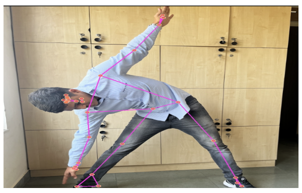
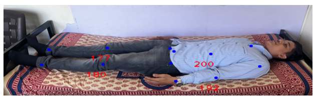
5.2 Pose prediction and Keypoint extraction

Making a real-time pose tracing system utilizing computer vision is difficult because it might be time-consuming to identify just the human body doing the stance among all the other items in the backdrop. From RGB video frames, the Openpose posture landmark as shown in table 2 model can infer 33 3-D landmarks on the body. The camera will use Open pose to record the person & yoga positions as they are being performed.When Open Pose estimates the pose of the real-time input stream, landmarks are superimposed on the feed and we get the output depicted in table 3, which also conducts keypoint/landmark extraction. Once extracted, landmarks are loaded into a CSV file as x, y, and z coordinates.

Table 2. Key Points used

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

Table 3.Asanas and corresponding key point

No	YogaAsan	Pose
1	Tadasana	
2	Vrikshasana	
3	Bhujangasana	
4	Padahastasana	
5	Trikonasana	
6	Savasana	

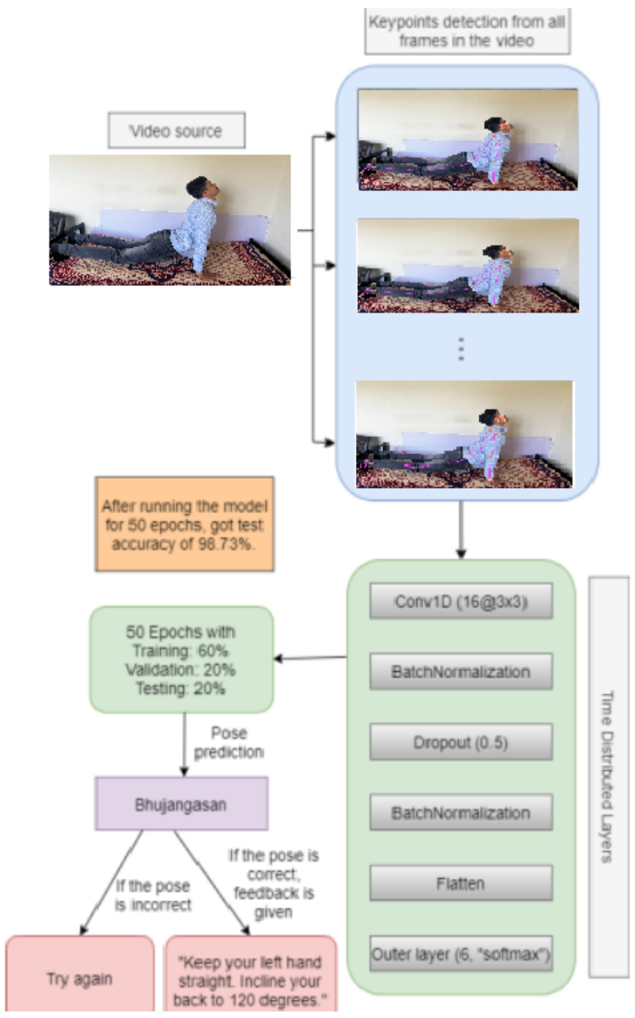
The next step is to preprocess the data after the landmarks have been recorded and stored in the CSV file. By determining whether or not there are any null values in our dataset and by addressing the missing values in our dataset, we will clean up our data set. After eliminating unnecessary points from the table, the values of the coordinated x and y coordinates are normalized to fit into our system. Rows that contain null entries are eliminated from the table.

5.4 Model:

5.4.1 CNN

One-dimensional, single-layer CNN and 3-dimensional 16 filters are trained using OpenPose keys. An 18 x 2 input display shows 18 points with X and Y coordinates

***Figure 3.** Working of CNN*


Keypoints detection from all frames in the video
Video source
After running the model for 50 epochs, got test accuracy of 98.73%.
Conv1D (16@3x3)
BatchNormalization
Dropout (0.5)
BatchNormalization
Flatten
Outer layer (6, "softmax")
Time Distributed Layers
50 Epochs with Training: 60% Validation: 20% Testing: 20%
Pose prediction
Bhujangasan
If the pose is incorrect
If the pose is correct, feedback is given
Try again
"Keep your left hand straight. Incline your back to 120 degrees."

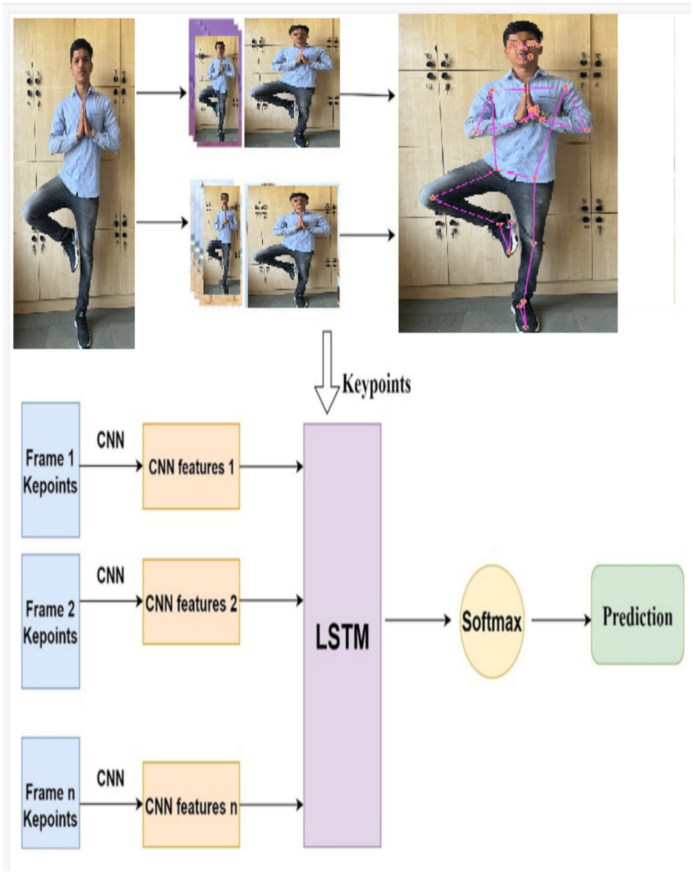
Ensemble regularization is applied to the output of CNN layers to speed up model convergence. In addition, we add a layer that avoids overfitting by randomly removing some weights. The activation function used to extract features in the keyframes of each frame is the corrected Linear Unit (ReLU). The final output is then smoothed, showing the possible poses in the state of entropy for all classes, before smoothing with softmax activation and dense layering with six units.

The loss function used to develop the model is cross-entropy, sometimes called softmax loss. This is done to ensure that the softmax activation is evaluated in the coupled solid layer. yoga position is divided into several categories, so use categorical cross-entropy as the mean of the loss function for multi-class classification. The learning rate is controlled using the Adam optimizer with an initial learning rate of 0.0001. A total of 100 training periods were used to train this model.

5.4.2 CNN + LSTM

CNN and LSTM are employed in deep learning models [22]. LSTM uses temporal data to make predictions while CNN analyzes frames and finds patterns. Features derived from CNN layer keypoints are used by LSTM cells to interpret the variation in features across multiple frames. CNN input is formed as follows: 45 frames in each frame, 18 keypoints and 2 X and Y coordinates for each keypoint represented by 45 x 18 x 2 numbers. The output from the CNN is fed to the LSTM as sequences. Convolution layers of 45 frames (1.5 seconds) of data dispersed in time.This is done to ensure that the softmax activation is evaluated in the coupled solid layer. yoga position is divided into several categories.

Figure 4. *Working of CNN + LSTM*

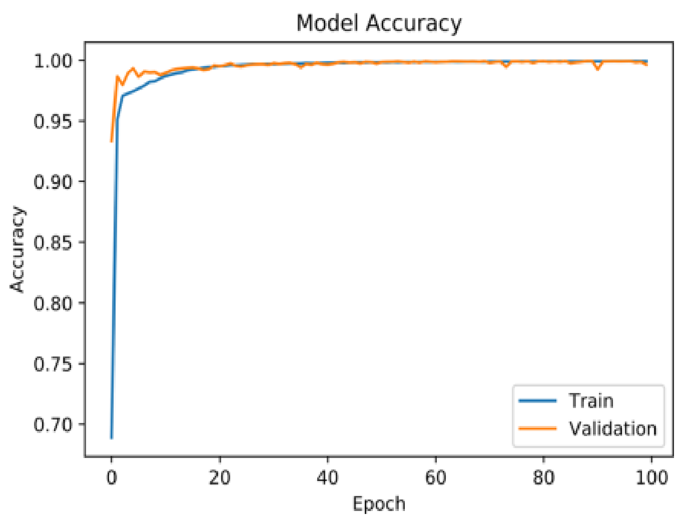


Because it is advantageous for operations involving motion, a time-dispersed layer is used. A one-dimensional vector representing the CNN output is then propagated to an LSTM layer with 20 units and an error bias of 0.5. The LSTM processes the entire yoga pose from its inception and uses the sequential nature of the input video data to detect temporal changes in the features identified by the CNN. As with Model 2, the rest of the CNN architecture remains unchanged. The schematic representation of the model is in Fig. 4

6. Training

First, CNN and LSTM models are applied to the model to predict posture. When the two are combined, an optimal collection of features filtered by CNN and long-term data dependence produced by LSTM is acquired. Has a Keras-based 00.....model Theano backend. Display of the output of a fully connected layer when Softmax is activated.

***Figure 5**. Graph of Model Accuracy Vs epoch*



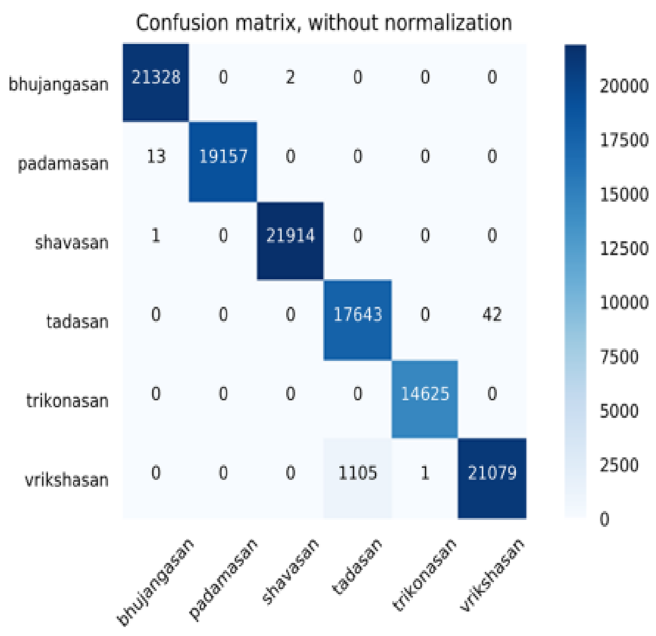
The categorical cross-entropy loss function is used to measure it because it is an appropriate method. The Adam optimizer controls the learning rate using a 0.0001 starting learning rate, a beta of 0.9, and no decay.

7. Experimental results

7.1 Confusion matrix

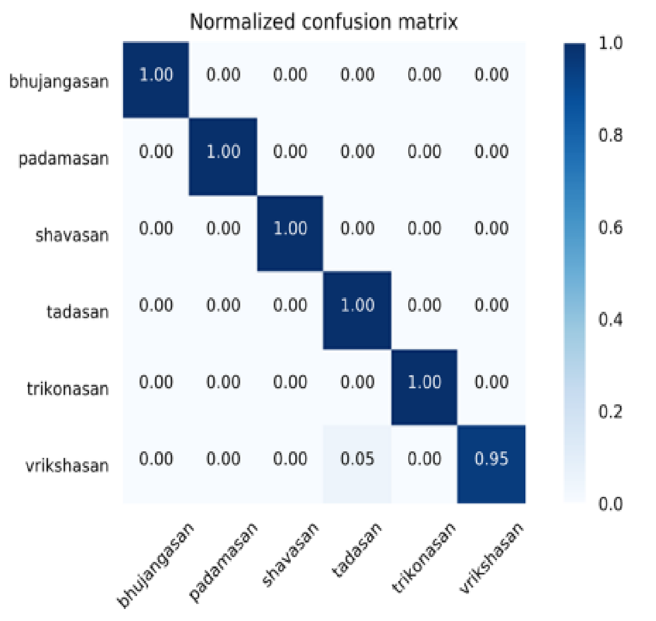
A confusion matrix is a crucial performance indicator that is used to calculate metrics like recall, accuracy, specificity, sensitivity, etc. For categorization problems, the predictions' outcomes are also summarised. [38]. Its four key values are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). [32]. After 100 training cycles, the system achieves a score of 99.34%. 100% accuracy was achieved while using training data, and 99.41% accuracy when using validation data. The system obtains a test accuracy of 99.04% for every frame. The accuracy of the model is illustrated in Figure 5.This suggests high correlation. A model's performance is assessed using a statistic called precision.

***Figure 6.** Confusion matrix without normalization*



	bhujangasan	padamasan	shavasan	tadasan	trikonasan	vrikshasan
bhujangasan	21328	0	2	0	0	0
padamasan	13	19157	0	0	0	0
shavasan	1	0	21914	0	0	0
tadasan	0	0	0	17643	0	42
trikonasan	0	0	0	0	14625	0
vrikshasan	0	0	0	1105	1	21079

***Figure 6.** Confusion matrix with normalization*



	bhujangasan	padamasan	shavasan	tadasan	trikonasan	vrikshasan
bhujangasan	1.00	0.00	0.00	0.00	0.00	0.00
padamasan	0.00	1.00	0.00	0.00	0.00	0.00
shavasan	0.00	0.00	1.00	0.00	0.00	0.00
tadasan	0.00	0.00	0.00	1.00	0.00	0.00
trikonasan	0.00	0.00	0.00	0.00	1.00	0.00
vrikshasan	0.00	0.00	0.00	0.05	0.00	0.95

The stronger the blue color is on the diagonal, the more closely the predicted and actual values correlate. A strong model can be inferred from the fact that the majority of the diagonal values are near to 1. This suggests high correlation. A model's performance is assessed using a statistic called precision. It is described as the percentage of samples accurately categorized as positive out of all samples that are truly positive [40]. Overall accuracy for the algorithm is 98.92%, and each individual asana has accuracy above 97%. These results are in line with those obtained for the test set, indicating a successful result in actual circumstances

8, Overall result

As demonstrated in table 3, a highly precise frontend was produced after combining all three phases to track the user in real time for pose prediction and correction. Figures 7 and 8 demonstrate the frontend's overall view and the outcome of pose prediction and correction, respectively.

Table 3*.Accuracy table*

Pose gradingModules	Accuracy(%)
Through Images	95-99
Through Real-time	90-98

Figure *7. Frontend*

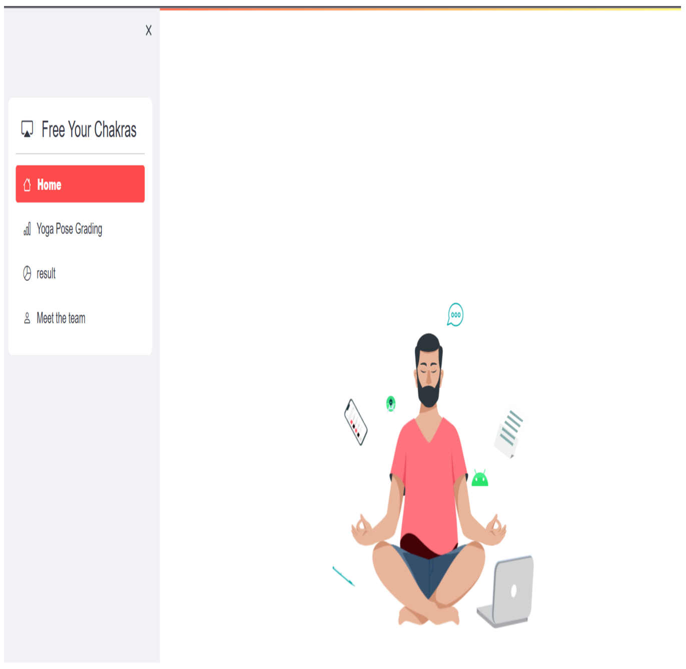
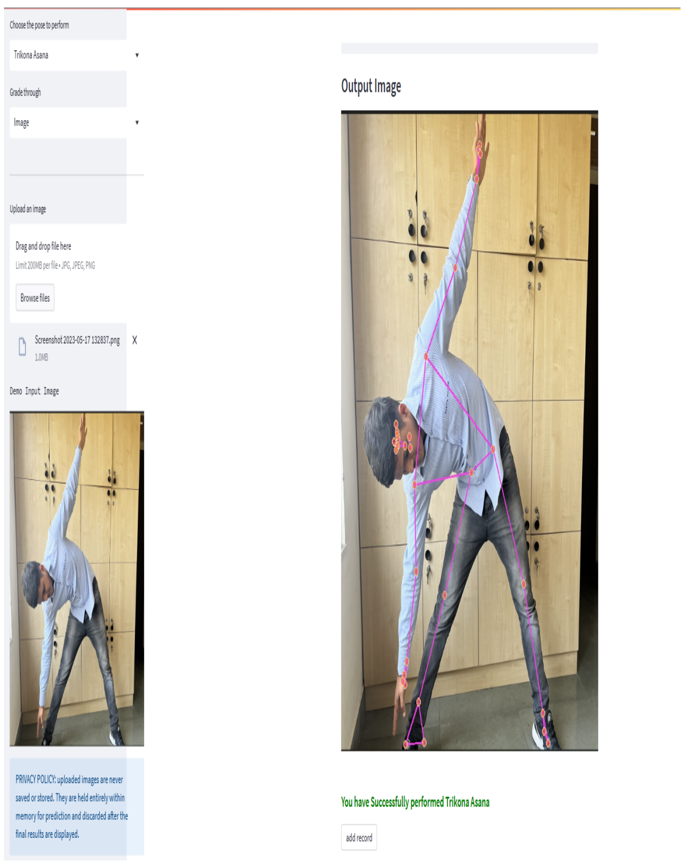


Figure 8.*Upload image for detection*


Choose the pose to perform

Trikona Asana

Grade through

Image

Upload an image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

Screenshot 2023-05-17 132837.png 1.0MB X

Demo Input Image

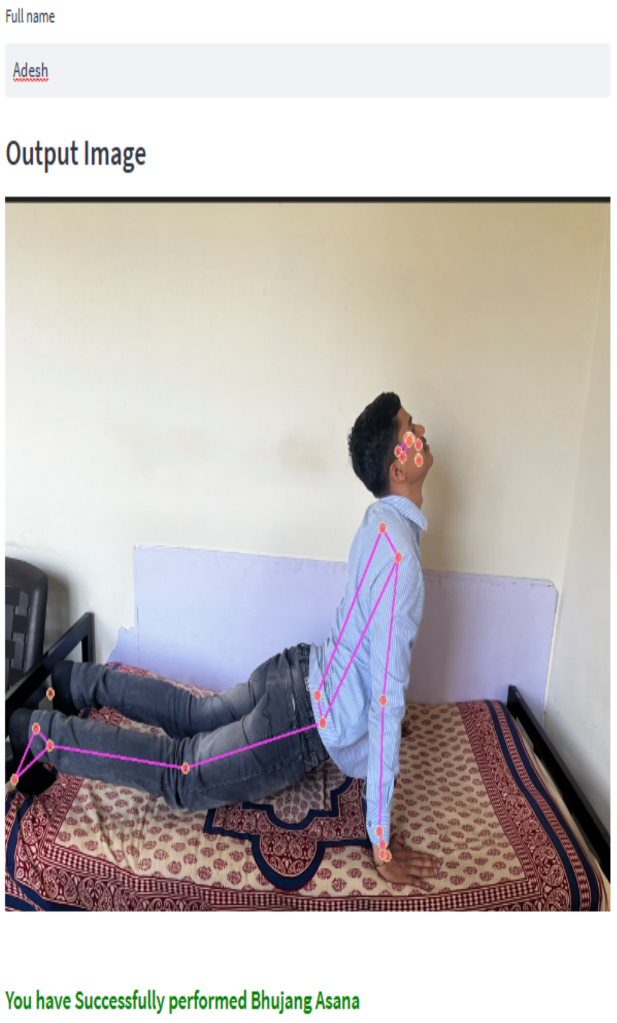
PRIVACY POLICY: uploaded images are never saved or stored. They are held entirely within memory for prediction and discarded after the final results are displayed.

Output Image

You have Successfully performed Trikona Asana

add record

Figure 9 *.Pose detected successfully*

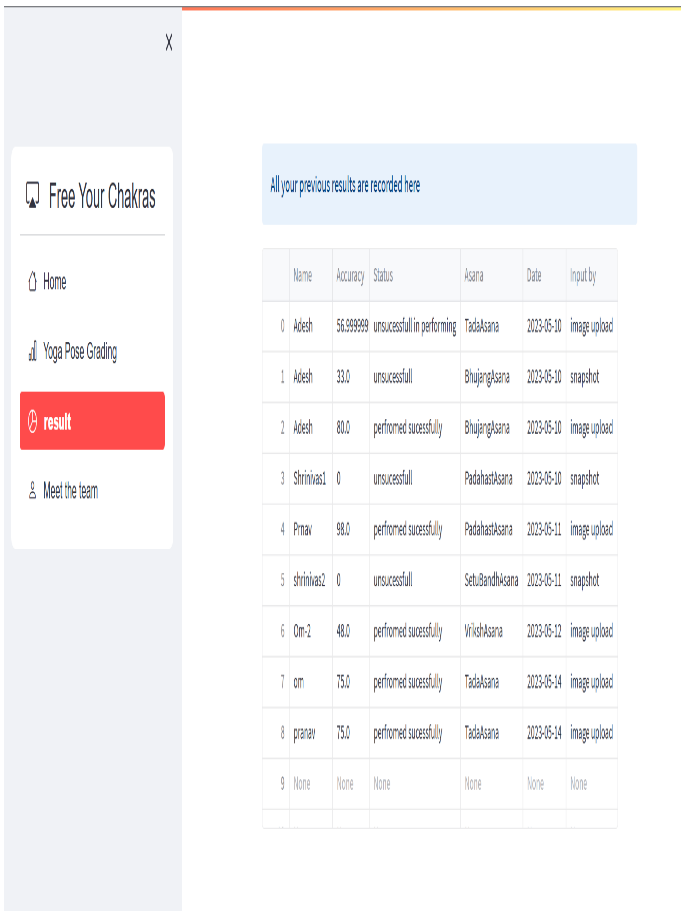

Full name

Adesh

Output Image

You have Successfully performed Bhujang Asana

Figure 10.Accuracy added to database



All your previous results are recorded here

	Name	Accuracy	Status	Asana	Date	Input by
0	Adesh	56.999999	unsucessfull in performing	TadaAsana	2023-05-10	image upload
1	Adesh	33.0	unsucessfull	BhujangAsana	2023-05-10	snapshot
2	Adesh	80.0	perfromed sucessfully	BhujangAsana	2023-05-10	image upload
3	Shrinivas1	0	unsucessfull	PadahastAsana	2023-05-10	snapshot
4	Prnav	98.0	perfromed sucessfully	PadahastAsana	2023-05-11	image upload
5	shrinivas2	0	unsucessfull	SetuBandhAsana	2023-05-11	snapshot
6	Om-2	48.0	perfromed sucessfully	VrikshAsana	2023-05-12	image upload
7	om	75.0	perfromed sucessfully	TadaAsana	2023-05-14	image upload
8	pranav	75.0	perfromed sucessfully	TadaAsana	2023-05-14	image upload
9	None	None	None	None	None	None

9.Conclusion

In this study, we have proposed a useful method for real-time yoga tracking. It initially locates the user's key points using the OpenPo library, where relative coordinates are recorded and saved in model.pkl. Then, a 45-frame sequence created from the real-time data is provided to the model. The model, which combines CNN and LSTM, initially applies CNN to detect useful properties before using LSTM to observe the occurrence of a series of frames. After calculating the probability of each pose for the current frame sequence the output by the softmax layer is the highest probability yoga trend. After 45 frames of polling the result of each frame is established and output to the user.

10.References

1. Wrigkas, M., Nikou, C. and Kakadiaris, I.A. (2015). Overview of human mental function. Frontiers in Robotics and Artificial Intelligence, 2. doi: 10.3389 / to 2013.

2. Kothari, Shruti, "Yoga Pose Classification Using Deep Learning" (2020). Graduate School. .DOI: https://doi.org/10.31979/etd.rkgu-pc9k

4. Newcombe, S. (2009). The Evolution of Modern Yoga: A Field Study. Kompas Agama, 3(6), 986–1002. doi:10.1111/j.1749-8171.2009.00171.x

5. Woodyard, C. (2011). Explore the healing effects of yoga and its ability to improve quality of life. International Journal of Yoga, 4 (2), 49.

doi: 10.4103 / 0973-6131.85485

6. Wrigkas, M., Nikou, C., Kakadiaris, I. (2015). Review of human activity recognition methods. https://doi.org/10.3389/frobt.2015.0

7. Xian-Ru Ke, Hoang Le Uyen Tuk, Ong-Jin Lee, Jenq-Neng Hwang, Jang-He ooo, Kyoung-Ho Choi (2013). A review of video-based human activity recognition. Computer. 2. 88–131.

doi: 10.3390/komputer2020088

9.Alzahrani, M., Kammon, S. (2016). Recognition of human activity: challenges and stages of the process. International Journal of Innovative

Research in Computer and Communication Engineering. 4 (5). 1111–1118.

10.Gupta, Saurabh (2021). Human Activity Recognition (HAR) is based on deep learning using wearable sensor data. International Journal of Information Management Information. 1 (2). doi: https://doi.org/10.1016/j.jjimei.2021.100046

11. Kadbhane, S., Dater, K., Jagdale, T., Dhongade, S., & Jagtap, G. (2021) Yoga pose recognition. International Journal of Advanced Research in Computer and Communication Engineering, 10 (1), 143-147. doi: 10.17148/IJARCCE.2021.10128

12. Haq, S., Rabbi, A.S., Laboni, M.A., Neehal, N., & Hossain, S.A. (2019). ExNET: Deep Neural Networks for Pose Detection. Communications in Computer and Information Science New Trends in Image Processing and Pattern Recognition, 186-193. doi:10.1007/978-981-13-9181-1_17

13. Agrawal, Y.., Shah, Y.. and Sharma, A. (2020). Implement machine learning methods to identify yoga postures. 2020 IEEE 9th International Conference on Communication Systems and Networking Technologies (CSNT), 40–43.

doi: 10.1109/CSNT48778.2020.9115758

14. Anilkumar, A., K.t., A., Sajan, S., & K.a., S. (2021) Designing an approximate yoga control system. SSRN Electronic Journal.

doi: 10.2139/ssrn.3882498

.

17. Zou, J., Li, B., Wang, L., Li, Y., Li, X., Lei, R., & Sun, S. (2019). An intelligent fitness trainer system based on the assessment of the human condition. Signal and Data Processing, Network and Computer, 593-599.

doi: 10.1007/978-981-13-7123-3_69

18. Adadav, Santosh and Singh, Amitojdeep and Gupta, Abhishek and Raheja, et al. (2019). Real-time yoga recognition using deep learning. Neural Computing and Applications. 31.https://link.springer.com/article/10.1007/s00521-019. 10.1007 / s00521-019-04232-7.

19. Vivek Anand Thoutam, Anugrah Srivastava, Tapas Badal, Vipul Kumar Mishra, G.R. Sinha, Aditi Sakalle, Harshit Bhardwaj, Manish Raj, "Evaluating and Feedbacking Yoga Using Deep Learning", Computational Intelligence and Neuroscience, Vol. 2022, Article number 4311350, 12 pages, 2022. https://doi.org/10.1155/2022/4311350

23. Al-Saffar, A.A., Tao, H., & Talab, M. (2017). A review of deep convolutional neural networks in image classification. International Conference on Radar, Antennas, Microwaves, Electronics and Telecommunications (ICRAMET).

doi: 10.1109 / executive.2017.8253139

24. Shiranthika, C., Premakumara, N., Chiu, H., Samani, H., Shyalika, C., & Yang, C. (2020) Human activity recognition using CNN and LSTM. 2020 5th International Conference on Information Technology Research (ICITR).

doi: 10.1109/icitr51448.2020.9310792.

25. Ullah, A., Ahmad, J., Mohammad, K., Sajjad, M., & Baik, S.W. (2018). Motion recognition in

video sequences using deep bipartite LSTM and CNN features. IEEE Introduction, 6, 1155-1166.

doi: 10.1109 / access.2017.2778011

28. Thakkar, V., Tewary, S., & Chakraborty, C. (2018). Batch regularization in convolutional neural networks - a comparative study with CIFAR-10 data. 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT). doi: 10.1109/eait.2018.8470438

30. Szandala, T. (2020). A review and comparison of commonly used activation functions for deep neural networks. Bio-inspired Neurocomputational Studies in Computational Intelligence, 203-224.

doi: 10.1007/978-981-15-5495-7_11.

32. Rusiecki, A. (2019). Category cross-section cut into deep study with volume label. Electronic letters, 55(6), 319–320. doi: 10.1049/el.2018.7980