Enhancing The Evolution And Analysis Of Body Posture In Different Self Learning Activities Processes

Prof. Roshni Narkhede[1], Saurabh Sonalkar[2], Dhiraj Yadav[3]
Computer Engineering Department[1,2,3]
Nutan Maharashtra Institute of Engineering and Technology, Pune, Maharashtra[1,2,3]

***Abstract*—** This work presents a novel approach to accurately assess Yoga poses using advanced deep learning algorithms. Our proposed system utilizes pose detection techniques to enhance the understanding and proficiency of Yoga practitioners. By employing multi-stage pose detection using a PC camera, we can accurately identify Yoga poses in real-time. Additionally, we introduce an innovative scoring algorithm applicable to various poses, ensuring comprehensive assessment capabilities. Our system's performance is evaluated across different Yoga poses and environmental conditions, demonstrating its robustness. Furthermore, we introduce a sophisticated deep learning model for real-time Yoga recognition, leveraging linear regression to extract features from key points detected in each frame using OpenPose.

***Keywords*—** Yoga disguise Recognition, Deep Learning Algorithms, Linear Retrogression, Real- time videotape Analysis, Pose Discovery System.

I. INTRODUCTION

Human pose estimation presents a formidable challenge in computer vision, involving the localization of human joints in images or videos to create a skeletal representation. Automatically detecting a person's pose in an image is complex due to factors such as scale, resolution, illumination variations, background clutter, clothing variations, and interactions with the surroundings.[2] Exercise and fitness have garnered significant interest in this field as people increasingly prioritize their health amidst busy schedules.[4] The University of Rochester Medical Center lists physical activity, nutrition, obesity, tobacco use, HIV/AIDS, mental health, injury, violence, environmental quality, and access to healthcare as the top ten health concerns.[3] Therefore, maintaining a healthy lifestyle is crucial, encompassing healthy eating, physical activity, weight management, andstress management.

Exercise, including calisthenics, strength training, balance exercises, cardio, and yoga, contributes to healthy physical conditioning.[3] While some may prefer exercising under the guidance of an instructor or at a facility, many opt for self-directed routines using instructional materials or online resources due to time constraints. However, improper exercise techniques can pose risks.

Thus, it is essential to provide proper guidance for individuals engaging in self-directed exercise to reap the benefits and improve overall health. Yoga, with its spiritual, physical, and mental benefits, involves intricate postures. However, correct form is paramount, as incorrect postures can be counterproductive or even harmful.[1] This underscores the importance of having an instructor to oversee sessions andcorrect posture.

Given the limited access to instructors for some individuals, an artificial intelligence-based system can beemployed to identify yoga poses and provide feedback to improve form.[7] This approach explores various methods for yoga pose classification, aiming to enhance individual performance and maximize the benefits of yoga practice.[5]

.

I. LITERATURE SURVEY

The exploration of yoga pose detection in computer vision and human-computer interaction has garnered significant interest. This examination delves into various methodologies, techniques, and advancements aimed at accurately detecting and categorizing yoga poses from images or videos.[6] Understanding the latest developments in this field is crucial for the development of robust systems that can assist practitioners in refining their yoga practice, facilitating remote instruction, and improving healthcare operations.

The first paper, titled "An Overview of the Fashionability of Yoga," highlights the increasing popularity of yoga and its associated physical, mental, and spiritual benefits. However, practicing yoga without proper guidance can lead to health issues such as strokes or nerve damage. Therefore, it is essential to ensure correct posture during yoga sessions. The proposed system aims to identify and visually guide practitioners in real-time, using a vision-based approach.[3] The Continuous Yoga Instructor system captures practitioner movements using a mobile camera, streaming at 1280 x 720 resolution and 30 frames per second to the detection system[8].

The second paper discusses the integration of yoga into people's lives worldwide and the need for scientific analysis of yoga postures. Recognizing yoga postures in real-time poses a significant challenge due to limited datasets and computational constraints.[1] To address this, a large dataset comprising 5500 images of ten different yoga poses was created. A pose estimation algorithm generates a skeletal representation of the human body, enabling joint angle extraction for use in machine learning models. Training on 80% of the dataset and testing on the remaining 20% achieved an accuracy of 99.04% using a Random Forest Classifier[2].

II. PROPOSED METHODOLOGY

The proposed Yoga Posture Detection and Correction System comprises four main components: Keypoints Detection using OpenPose, Keypoints Detection using Mask RCNN, Higher Probability Prediction & Comparison, and an Android Trainer Application. The system workflow involves capturing and streaming the practitioner's movements in real-time to the system through a media streaming server. The system then

utilizes a pose estimation library, either OpenPose or Mask RCNN, to detect the key joints of the practitioner. These keypoints are then passed to the yoga pose detection module.Using the keypoints data, the system predicts the practitioner's yoga pose before they reach the final phase of the asana, allowing for real-time guidance. The results and accuracy values are then transferred to the Android application via a data channel.

A. Dataset:

A comprehensive dataset for yoga postures is available, consisting of six poses: Bhujangasana (Cobra Pose), Padmasana (Lotus Pose), Shavasana (Corpse Pose), Tadasana (Mountain Pose), Trikonasana (Triangle Pose), and Vrikshasana (Tree Pose). The dataset includes videos of fifteen individuals (ten males, five females), recorded at 1366 x 768 resolution and 30 frames per second using a standard webcam. The videos were recorded in typical indoor lighting conditions, with a distance of 4 meters from the camera. To facilitate efficient streaming, the videos were resized to 1280 x 720 resolution. The total duration of the dataset is approximately 1 hour and 6 minutes, comprising roughly 118,950 frames, with an average duration of 45 seconds per video.

B. Keypoints Discovery using OpenPose:

The incoming video stream is processed using OpenPose to detect 25 keypoints in the body, including ankles, eyes, elbows, hips, knees, nose, neck, shoulders, and wrists. The detected keypoints are then used to generate point maps and confidence charts, followed by a greedy algorithm to refine the results. The output, in JSON format, contains the body part locations for each frame, which are then passed to the higher probability prediction and comparison component.

C. Keypoints Discovery using Mask RCNN:

Similarly, the incoming video stream is processed using Mask RCNN for human detection and keypoints extraction. The keypoints are obtained by applying convolutional neural networks (CNNs) and region proposal networks (RPNs) to generate region proposals and predict the presence of a person in each region. These regions are then aligned and fed into a fully connected network (FCN) to generate bounding boxes for each person. The generated keypoints are refined using non-maximum suppression before being passed to the higher probability prediction and comparison component.

Higher Probability Prediction & Comparison:

This component utilizes the JSON data obtained from the pose estimation modules to predict the practitioner's pose and provide feedback on their performance relative to the original asana requirements. The aim is to provide feedback with minimal latency to ensure a seamless practitioner experience.

III. IMPLEMENTATION DETAILS

The proposed system classifies the Yoga pose into four situations so that the learner intimately can understand the evaluation result. The four classes are "perfect", good", "not good", and "bad". Each class is named by comparing the average angle difference for all the joints. Result value = toatal angle difference/ total number of common The result value is counterplotted with the range function of classifying the performance. The range is from 0 to 9 and there are at most 360 degrees. According to human's geste, they're interested in the overall performance result and they will try to ameliorate the result's position. The pose bracket is defined with the threshold given in

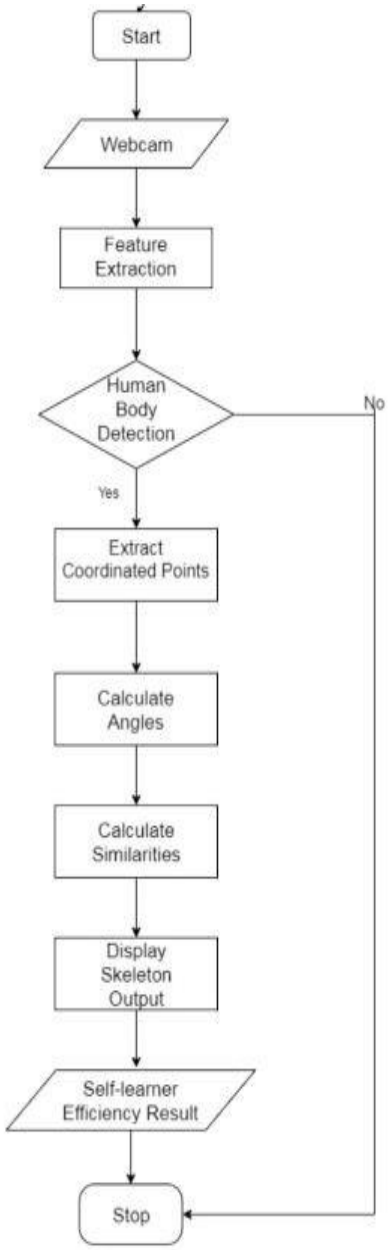
Performance results range

Class Level	Ranges
Perfect	0:9
Good	10:18
Fair	19:27
Not Good	28:36
Bad	37:360

IV. RESULTS AND DISCUSSIONS

Several models underwent training using keypoints data obtained from both OpenPose and Mask RCNN pose estimation modules to determine the optimal module for the system. It was found that the model trained with keypoints derived from OpenPose performed commendably overall, exhibiting the least latency when tested with real-time feeds. However, it was noted that this model encountered challenges in distinguishing between Tadasana and Vrikshana poses on occasion, possibly due to the similarities in the movements leading to both poses. Notably, this model achieved a high accuracy of 99.87% on the training data and 99.91% on unseen test data. Despite achieving satisfactory accuracies, individual models designed for the head, torso, and legs exhibited higher computational complexity, resulting in longer prediction times compared to other models.

Figure 1 : 24 Joints of human skeleton



Flowchart : Yoga Pose Assessment System Body

Angle Difference

Upon calculating the angle of each specified joint for both the images of the instructor and the learner, the difference between them is determined by the formula:

Difference = | instructor angle - learner angle |

- *Assessment Result Output*

  To ameliorate the understanding of a tone-learner, a grade colour display is a better way to show the angle difference value is assigned to each common with the corresponding colour value. The system maps the difference value to the colour value. Specifically, red is used to represent the large angle difference as the wrong result, and green is to represent the small difference as the correct result, as shown in Figure.

Figure 2 : Yoga Posture

Then, the gradient colour of this segment is determinedby the following equation:

$color\ value$ =[($d1$+$d2$) /2] $threshold$where,
d1, d2 = difference values of two consecutive jointsthreshold = 255 / 45

If the angle difference is zero, the colour is green. If the difference is over 45 degree, the colour turns red. Figureshows the example for one segment, and Figure shows the example for the whole body.

Figure 3 : Real life complex scenes for tree pose

V. ALGORITHM

Creating a yoga pose detection and analysis algorithm using Convolutional Neural Networks (CNNs) involves several steps. Here's an algorithm outline:

1. Data Collection and Preprocessing:
 - Gather a dataset of yoga images or videos with labeled poses. Ensure diversity in poses, backgrounds, lighting conditions, and individuals.
 - Preprocess the data by resizing images to a consistent resolution, normalizing pixel values, and potentially augmenting the dataset with techniques like rotation, flipping, and brightness adjustments to increase variability.

2. Dataset Labeling:
 - Label each image or video frame with the corresponding yoga pose(s) present in it. This can be done manually or using automated tools if available.

3. Dataset Splitting:
 - Split the dataset into training, validation, and testing sets. Typically, you might use a split like 70% for training, 15% for validation, and 15% for testing.

4. CNN Architecture Design:
 - Design a CNN architecture suitable for pose detection. This architecture should take input images as input and output the probabilities of each pose class.
 - Consider architectures like ResNet, MobileNet, or custom architectures tailored to your specific requirements.

5. Training:
 - Train the CNN model on the training dataset using techniques like stochastic gradient descent (SGD) or Adam optimization.
 - Utilize techniques like data augmentation to increase the variability of the training data and prevent overfitting.
 - Monitor the model's performance on the validation set and adjust hyperparameters as needed.

6. Testing and Evaluation:
 - Evaluate the trained model on the testing set to assess its performance in detecting yoga poses.
 - Calculate metrics such as accuracy, precision, recall, and F1 score to quantify the model's performance.
 - Visualize the model's predictions on sample images or videos to gain insights into its strengths and weaknesses.

7. Pose Analysis:
 - Once a pose is detected, perform further analysis to provide insights into the quality and correctness of the pose.
 - Calculate metrics such as alignment, balance, symmetry, and pose difficulty based on the detected poses and key points.

8. Deployment and Continuous Improvement:
 - Deploy the trained CNN model in a real-world application environment, such as a mobile app or web platform.
 - Gather feedback from users and iteratively improve the model based on their experiences and suggestions.
 - Continuously update the model with new data and advancements in CNN architectures and training techniques to enhance performance and accuracy over time.

By following this algorithm, you can develop a robust yoga pose detection and analysis system using Convolutional Neural Networks.

VI. CONCLUSION

In conclusion, our approach represents a well-thought-out integration of advanced neural network layers, specifically leveraging the time-distributed Convolutional Neural Network (CNN) layer to discern patterns among key points within a single frame, while simultaneously employing the Long Short-Term Memory (LSTM) layer to memorize patterns observed in recent frames. This combined use of CNN and LSTM has proven effective in capturing both spatial and temporal information, thereby enhancing the system's ability to recognize and understandsequential movements inherent in Yoga poses.

The inclusion of LSTM in our model serves a dual purpose – not only does it facilitate the memorization of patterns across frames, but it also helps minimize errors arising from false key point detections. Additionally, the integration of polling for denoising further strengthens the system's robustness. By incorporating these mechanisms, our model becomes proficient at handling the sequential nature of Yoga images, leading to more accurate and reliable pose detection. This emphasis on sequential frames acknowledges the dynamic and continuous nature of Yoga movements, ensuring that the model can effectively adapt to the fluidity inherent in Yoga practice.

In essence, the synergy between the time-distributed CNN layer and LSTM, along with denoising through polling, signifies a comprehensive strategy to optimize the system's performance. This strategic combination not only enhances accuracy but also fortifies the system against potential disruptions caused by false detections. As we explore future applications of this technology, the robustness achieved through these methodologies positions our system as a promising tool not only for pose detection in Yoga but also for broader applications in movement analysis, fitness tracking, and rehabilitation exercises.

ACKNOWLEDGMENT

This document is supported by NMIET, Pune(410507) We would like to express our sincere gratefulness to everyone who gave us precious opinions toward the completion of the forum report of "Enhancing the Evolution and Analysis of Body Posture in Different Self Learning Activities Processes ", which is part of the class of this forum. We thank the cooperating departments for furnishing precious support and prerequisites for system development. We're veritably thankful to

Professor Roshani Narkhede for guiding us in the right way. giving her time whenever demanded, giving her knowledge and participating their experience in carrying out this design. We'd like to thank you for correcting our dubieties by furnishing them.

REFERENCES

1) V. Akuhota and S. F. Nedler, ”Core Strengthening,” American Academy of Physical Medicine and Rehabilitation , 2004.

2) R. Szeliski, ”Computer Vision: Algorithms and Applications,” Springer, 2010.

3) G. Bradski and A. Kaehler, ”Learning OpenCV,” O’Reilly, 2008.

4) Z. Cao, T. Simon, S.-E. Wei and Y. Sheikh, ”Realtime MultiPerson 2D Pose Estimation using Part Affinity Fields,” The Robotics Institute, Carnegie Mellon University, 2017.

5) P. Ganesh, ”Human Pose Estimation : Simplified,” Towards Data Science, 26 March 2019. [Online]. Available: https://towardsdatascience.com/humanpose estimationsimplifie d6cfd88542ab3. [Accessed 3 April 2020].

6) T. Amert, N. Otterness, M. Yang, J. H. Anderson, and F. D. Smith, “GPU scheduling on the NVIDIA TX2: Hidden details revealed,” in Proc. IEEE Real-Time Syst. Symp. (RTSS), Dec. 2017, pp. 104–115.

7) B. Schoettle, “Sensor fusion: A comparison of sensing capabilities of human drivers and highly automated vehicles,” Univ. Michigan, Ann Arbor, MI USA, Tech. Rep. SWT-2017-12, 2017.

8) D. Feng et al., “Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges,” IEEE Trans. Intell. Transp. Syst.,early access, Feb. 17, 2020