NamastAI

EECS E6895: Advanced Big Data and AI

A1: Deep Video Understanding (Visual + Knowledge)

Balachander Sathianarayanan
Electrical Engineering Department
Columbia University
BS3507

***Abstract*—This project aims to develop an innovative yoga pose estimation system utilizing advanced computer vision techniques. The system will accurately analyze and grade yoga poses in real-time, providing personalized feedback based on alignment, balance, and form. Leveraging pose detection algorithm like Movenet to detect keypoints, the system addresses challenges such as varying lighting conditions and occlusions, ensuring robust performance across diverse environments. The frames are processed in real time, and can be run at 30 FPS. Then a score is calculated for each joint on how well you do a particular yoga pose with respect to a professional yoga performer's pose. Ultimately, the project seeks to revolutionize the yoga experience by making it more accessible, engaging, and effective.**

I. Introduction

In today's fast-paced world, the pursuit of holistic well-being has become increasingly important. Yoga, an ancient practice originating from India, offers a pathway to achieving physical, mental, and emotional balance. Central to the practice of yoga is the alignment of body and mind, cultivated through the performance of various postures or asanas.

While the benefits of yoga are well-established, practitioners often face challenges in achieving proper posture alignment and form. This is where technology can play a transformative role. By harnessing the power of machine learning and real-time video analysis, our project aims to provide practitioners with immediate feedback on their yoga poses, facilitating a deeper understanding of body alignment and enhancing the overall yoga experience.

This project integrates several key components, including WebRTC capture for real-time video communication, Streamlit deployment for creating a user-friendly web application interface, and the MoveNet Lightning model for accurate pose estimation. By leveraging these technologies, users can receive personalized feedback on their yoga poses, enabling them to refine their practice and deepen their mind-body connection.

Through this innovative application of technology, we seek to bridge the gap between traditional yoga practice and modern advancements, empowering practitioners to embark on a journey of self-discovery and transformation. Join us as we explore the intersection of technology and wellness, reimagining the way we approach yoga and holistic well-being in the digital age.

II. Related Work

Deep learning has revolutionized pose estimation with the development of highly accurate and efficient models. Convolutional Neural Network (CNN) has been adopeted as the backbone architecture for many pose estimation systems because of Hierarchical Feature Extraction, Translation Invariance, Parameter Sharing and availabilty of good Pre-trained Models.

One of the challenges in pose estimation is handling occlusions and complex body poses. These issues have been addressed by developing multi-person pose estimation models capable of detecting and tracking multiple individuals simultaneously. The following are the popular pose estimation models.

PoseNet operates [1] as a real-time pose estimation model, leveraging convolutional neural networks (CNNs) to detect and track human body poses within input images or video frames. By analyzing the spatial relationships between pixels, PoseNet identifies key body landmarks, including the nose, shoulders, elbows, wrists, hips, knees, and ankles. This process involves multiple stages of convolutional and pooling operations to extract features that inform the likelihood of keypoints existing at specific locations in the image. Through iterative refinement and optimization, PoseNet adjusts its predictions based on contextual information and spatial dependencies, ultimately producing a set of keypoints representing detected human poses. With its ability to provide real-time pose estimation,

MoveNet, developed by Google [2], is a lightweight pose estimation model optimized for real-time performance on various devices. It employs efficient convolutional neural networks (CNNs) to detect and track human body poses accurately. MoveNet utilizes a single-stage architecture, allowing for fast inference without sacrificing accuracy. The model operates by analyzing input images or video frames to identify key body landmarks, known as keypoints, including joints and limbs. It can simultaneously detect multiple poses, making it suitable for applications like fitness tracking, gesture recognition, and augmented reality. MoveNet's lightweight design and real-time capabilities make it versatile for deployment in resource-constrained environments and interactive experiences.

OpenPose is a popular pose estimation framework [3] that operates by analyzing input images or video frames to detect and track human body poses. It utilizes a multi-stage deep

learning architecture, typically based on convolutional neural networks (CNNs), to identify key body landmarks, known as keypoints, such as joints and limbs. OpenPose processes the input data to infer the spatial relationships between keypoints, enabling accurate estimation of human poses. It employs techniques such as heatmap regression and part affinity fields to refine pose estimations and handle occlusions.

BlazePose is a pose estimation model developed by Google [4], [5], designed for real-time performance and accuracy. It utilizes a lightweight architecture based on efficient convolutional neural networks (CNNs) to detect and track human body poses quickly and accurately. BlazePose operates by analyzing input images or video frames to identify key body landmarks, known as keypoints, such as joints and limbs. The model employs advanced techniques to handle occlusions and refine pose estimations and is invariant to changes in scale and rotation, allowing it to accurately estimate poses across a wide range of viewing angles and distances, ensuring robust performance in diverse environments.

Overall [6]–[10] MoveNet and BlazePose are both optimized for real-time performance and use lightweight architectures, making them suitable for deployment on various devices. PoseNet and OpenPose offer comprehensive pose estimation frameworks with multi-stage architectures, providing detailed insights into human body poses and interactions. OpenPose excels in handling complex scenarios and occlusions, making it suitable for applications requiring accurate pose estimation in challenging environments.BlazePose stands out for its efficiency and accuracy, offering real-time performance without compromising on the quality of pose estimations.

Another area of active research is 3D pose estimation, which outputs the three-dimensional positions of body joints from monocular images or videos. Research articles such as MonoPerfCap [11] utilize geometric constraints and depth information to infer accurate 3D poses, opening up new opportunities for applications in augmented reality and human-computer interaction.

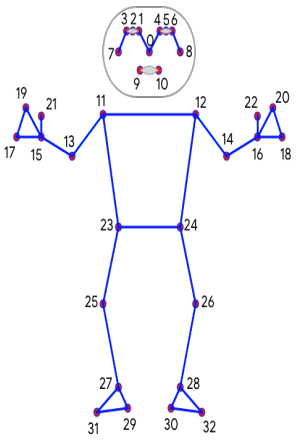

Fig. 1. Blazepose Keypoints

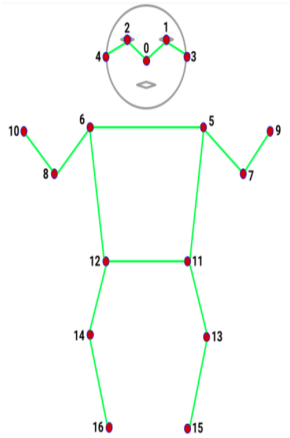

Fig. 2. Movenet & Posenet Keypoints

III. Data

Since the project involves using a pose detection model and then later the poses are scored based on other professional's pose detection model's keypoints, it is important to a novel dataset. Hence there are 2 parts in data used. One for the model, the other for scoring.

A. *Data for Fine Tuning the model*

Human pose estimation datasets play a crucial role in training and evaluating pose estimation algorithms. These datasets provide annotated images with key body joint locations, enabling us to fine-tune the pre trained models. Here are three prominent datasets widely used in the field of human pose estimations:

1) **Microsoft Common Objects in Context (COCO) Dataset**: The Microsoft COCO dataset [12], updated in 2017, is one of the most widely used and comprehensive datasets for various computer vision tasks, including object detection, segmentation, and pose estimation. For pose estimation, the dataset contains around 164,000 images with 250,000 annotated human poses, including the locations of 17 body keypoints.
2) **MPII Human Pose Dataset**: The MPII Human Pose dataset [13], released in 2014 introduced by the Max Planck Institute for Informatics, is a large-scale dataset specifically designed for the task of multi-person pose estimation. It consists of around 25,000 images of over 40,000 people with annotated 16 body joints, including a diverse range of human poses, body articulations, and activities.
3) **PoseTrack Dataset**: The PoseTrack dataset [14] is focused on pose estimation in video sequences, addressing the challenges of tracking and articulating human poses across multiple frames. It contains over 514 videos including 66,374 frames in total. The annotations include 15 body keypoints location.

Using the COCO Dataset for the project is a strategic choice, especially considering that the keypoints it deals with are

consistent with both MoveNet and PoseNet models. This alignment ensures that the dataset's annotations and the models' keypoint definitions are compatible, facilitating seamless integration and evaluation of the models' performance.

B. Data for scoring User's pose

Yoga involves various techniques including physical postures (asanas), breath control (pranayama), meditation, and ethical principles. This project discusses about the postures which can be observed using cameras. Yogas are effective when followed in certain set of sequences which can promote mindfulness and enhances the mind-body connection. In some yoga traditions, specific sequences carry symbolic or philosophical significance. One such sequence which is quite popular and adopted by the practitioners is "Surya Namaskar".Surya Namaskar (Sun Salutation) is practiced as a form of reverence to the sun and is associated with spiritual awakening and renewal. It consists of 12 poses.[Figure3].

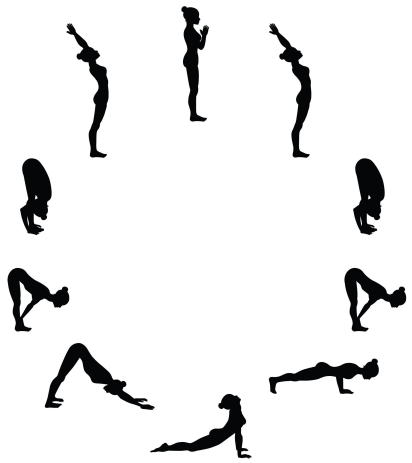

Fig. 3. Surya Namaskar poses

The availability and quality of data plays a important role. Focusing specifically on side profile videos scrapped from YouTube, where the quality is set at 1080p and the entire body is clearly visible, ensures a standardized dataset for analysis. 9 such videos [15]–[23] are selected and the videos are sampled at the required time such that only the poses are extracted.

Side profile videos offer an optimal perspective for capturing these nuances. They provide a clear view of the body's lateral movements, aiding in precise pose estimation. This angle facilitates a comprehensive understanding of posture alignment, limb positioning, and spinal curvature, crucial for accurate analysis. While other views, such as front or rear angles, may provide additional information, they can also introduce complexities related to occlusions, perspective distortion, By prioritizing side profile views, the user's pose can be compared with the professional's pose. Additionally, ensuring the entire body is covered minimizes occlusions and ensures completeness in pose representation which is further passed on to Movenet Lightning model.

Data Preprocessing: Prior to pose detection, a series of preprocessing techniques were applied to optimize the quality of the input images. These included cropping and flipping to ensure uniform orientation, with all performers facing the left side for consistency. Gamma correction was employed to adjust brightness and contrast levels, enhancing feature visibility. Additional preprocessing steps involved noise reduction and contrast enhancement to improve image clarity. By carefully refining the input data through these preprocessing techniques, the overall quality and reliability of the pose detection process were significantly enhanced, leading to more accurate and confident results.

IV. Methods

There were multiple models were considered for Pose Detection model as discussed earlier. There were also various methods to calculate grades as well, which will be discussed in detail in the following section.

A. Pose Detection Model:

Based on the chosen dataset, i.e, COCO dataset, we are considering Movenet and Posenet. The models were fine tuned.

PoseNet: PoseNet, introduced in 2017, represents several previous iterations of pose estimation models. Trained on the COCO dataset, it also provides single and multi-pose prediction capabilities. The single-pose variant focuses on recognizing a single person in a photo or video, while the multi-pose variant can recognize multiple individuals. Each variant uses specific parameters and methods. Single-pose estimation is simple and fast and requires only one person in the frame; otherwise, points from multiple individuals may be mistakenly combined into a single theme.

PoseNet used MobileNetV1 as it's Backbone CNN for feature extraction.MobileNetV1 is characterized by its compact size and rapid processing, sacrifices some accuracy. The model outputs the coordinates of 17 key points, accompanied by confidence scores, providing detailed insights into the pose configuration within the image or video. Figure 4 represents implementation of Posenet.

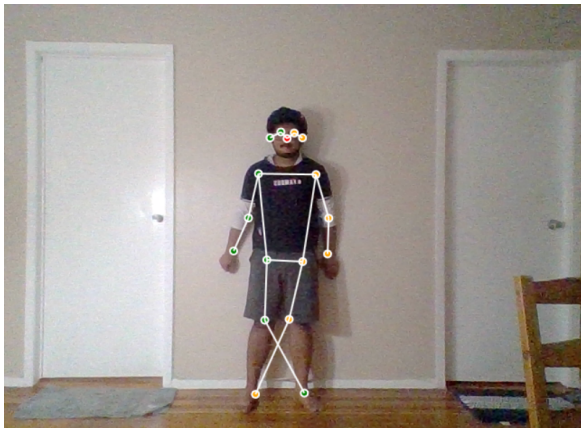

Fig. 4. Posenet Implementation

MoveNet: Developed by IncludeHealth, a digital health company, and sponsored by Google, MoveNet represents a milestone in remote health care, as revealed by 2021. Like PoseNet, MoveNet uses TensorFlow.js for the web functionality and TensorFlow Lite for mobile applications This example offers two distinct options: Lightning[Figure 5] optimized for performance, and Thunder[Figure 6] with an emphasis on accuracy. Lightning handles fixed-size (192×192) inputs with a depth factor of 1.0, while Thunder handles 256×256 inputs with a depth factor of 1.75, changing the channel count of the input, usually in RGB format Thunder has 1.75 times more layers than Lightning, resulting in 1.75 times a Computational burden there.

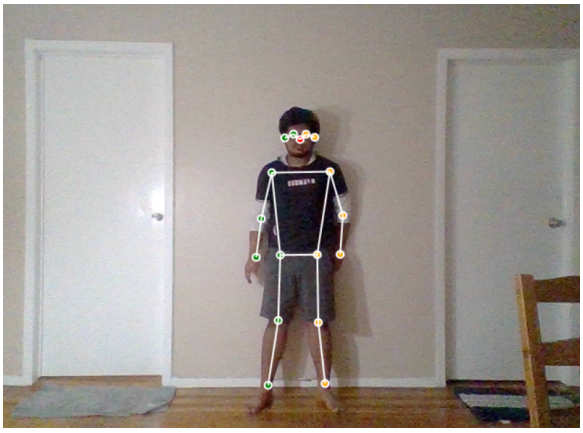

Fig. 5. Movenet Lightning Implementation

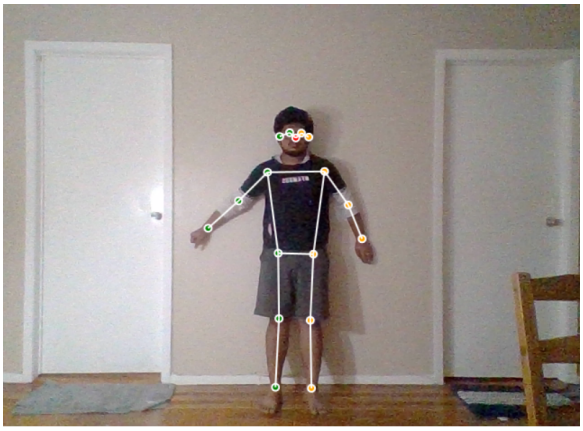

Fig. 6. Movenet Thunder Implementation

MoveNet takes a bottom-up approach, using the TensorFlow object recognition API and MobileNet V2 for feature extraction. Driven by CenterNet, MoveNet departs from traditional anchor-based detection models by focusing only on the focal point, facilitating object detection through regional offers without the need for non-maximum suppression (NMS).

B. Dataset and deployable App's pipleines

There are multiple data collection pipeline were considered by which how the user's pose is graded, the following are the pipelines that are considered.

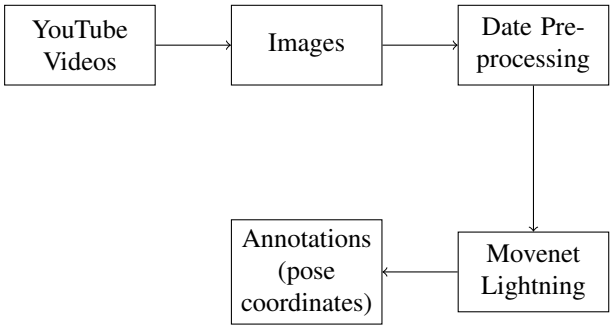

Fig. 7. Flowchart of Data collection pipeline

Pipeline 1: The first pipeline considered for calculating grade, is by directly comparing the keypoints. The keypoints are considered are:

- Nose
- Left eye
- Right eye
- Left ear
- Right ear
- Left shoulder
- Right shoulder
- Left elbow
- Right elbow
- Left wrist
- Right wrist
- Left hip
- Right hip
- Left knee
- Right knee
- Left ankle
- Right ankle

The entire data collection pipeline [Figure7] is explained below:

1) The videos are sampled and the required frames are collected.
2) The collected images needs to be cropped to a aspect ratio of equal width and height, since Movenet Lightning accepts an image of size (192, 192), The cropping needs to be manually done to ensure the human is entirely enclosed in the image.
3) The processed images are then passed on to Movenet Lightning model to get the coordinates of the keypoints.

Pipeline 2: In the experiments, it will be observed that there is a high standard deviation in the position of key points in the dataset. Hence a new pipeline needs to be considered which reduces this standard deviation. The second pipeline considered for calculating grade is by comparing the key points after passing through a human detection model. A pretrained yolo v7 mini model is considered. The key points are the same as considered before. The entire data collection pipeline [Figure8] is explained below:

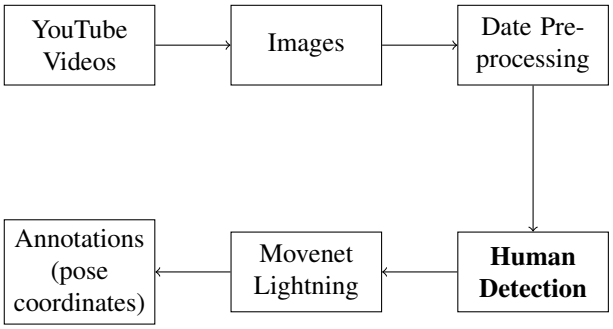

Fig. 8. Updated Flowchart of Data collection pipeline

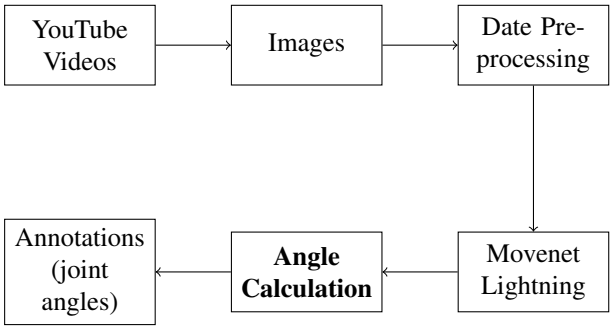

Fig. 9. Updated Flowchart of Data collection pipeline

1) The videos are sampled and the required frames are collected.
2) The collected images needs to be cropped to a aspect ratio of equal width and height, since Movenet Lightning accepts an image of size (192, 192), The cropping needs to be manually done to ensure the human is entirely enclosed in the image.Also flipping was done to ensure uniform orientation, with all performers facing the left side for consistency. Gamma correction was employed to adjust brightness and contrast levels, enhancing feature visibility.
3) The processed images are then passed on to Movenet Lightning model to get the coordinates of the keypoints.
4) After the keypoints are retrieved the angles of the chosen joints are calculated. First, The vectors are drawn based on a human sketch figure as shown in Figure ref and the 2 vectors at that joint are considered and the angle is computed by inverse cosine formula. Finally, it converts the result from radians to degrees.

Pipeline 3: In the experiments, it will be observed that there introducing a human detection model affects the processing time, if the same pipeline is deployed to score the user's pose. So, the third pipeline considered for calculating grade is by comparing certain joint angles and extra image preprocessing steps were considered for data collection. The joint angles considered are

- Left shoulder angle (LSA)
- Right shoulder angle (RSA)
- Left elbow angle (LEA)
- Right elbow angle (REA)
- Left hip angle (LHA)
- Right hip angle (RHA)
- Left knee angle (LKA)
- Right knee angle (RKA)

The entire data collection pipeline [Figure9] is explained below:

1) The videos are sampled and the required frames are collected.
2) The collected images needs to be cropped to a aspect ratio of equal width and height, since Movenet Lightning accepts an image of size (192, 192), The cropping needs to be manually done to ensure the human is entirely enclosed in the image. Also a series of preprocessing techniques were applied to optimize the quality of the input images. Gamma correction was employed to adjust brightness and contrast levels, enhancing feature visibility.
3) The processed images are then passed on to Movenet Lightning model to get the coordinates of the keypoints.

C. *Scoring User's pose:*

The score d

V. Experimental Results

A. *Results for Pose Detection Models*

The following parameters were considered for choosing the appropriate model. The models have been trained over 20000 images, with batch size of 32, with a total of 2000 epochs.

Accuracy:

Accuracy in the context of pose estimation or keypoint detection refers to the measure of how closely the estimated positions of body joints match the ground truth or manually annotated positions. It is typically calculated as the percentage of correctly identified keypoints out of the total keypoints in the dataset.

Observation:

We can observe from Table I and Figure that Accuracy of Posenet is far better than Thunder and Lighning models. Posenet is better than Thunder by 22%. But the performance of Lightning and Thunder are almost the same. High accuracy indicates that the pose estimation model is effectively capturing the nuances of body poses and movements, making it suitable for applications such as yoga posture analysis. Hence in this aspect Posenet is better.

PDJ:

The Percentage of Detected Joints refers to the proportion of joints or key points on the human body that were successfully detected by a pose estimation or keypoint detection algorithm out of the total expected number of joints. In applications such as yoga posture analysis, where accurate detection of body joints is crucial, this metric indicates the reliability and effectiveness of the pose estimation model.

Observation:

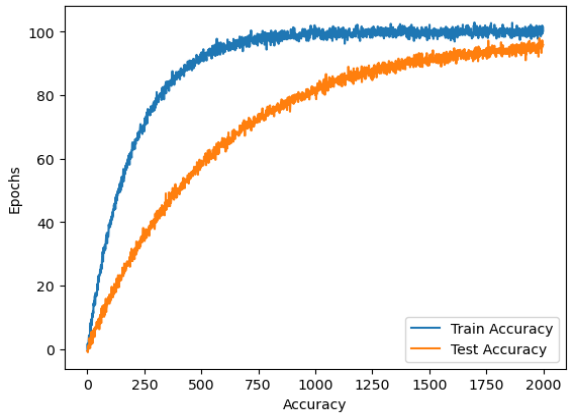

Fig. 10. Accuracy plot of Posenet

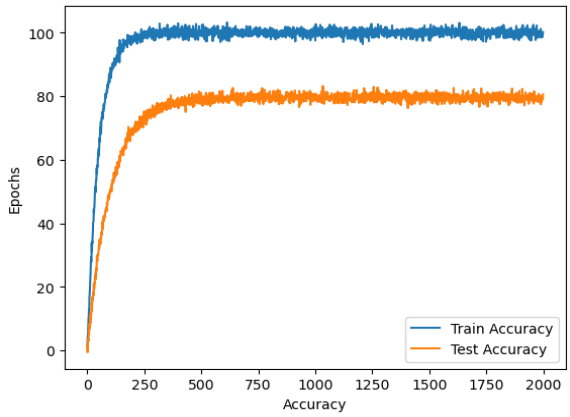

Fig. 11. Accuracy plot of Movenet Lightning

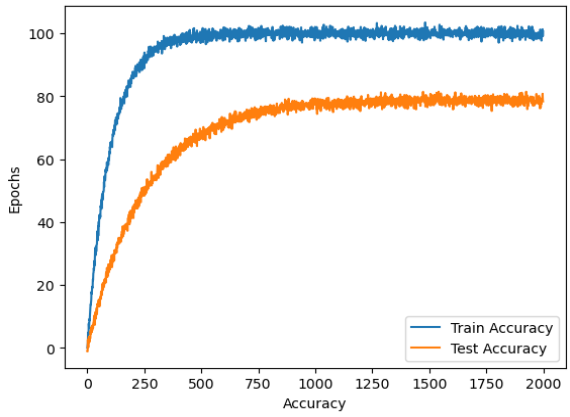

Fig. 12. Accuracy plot of Movenet Thunder

It is observed from Table Ithat Movenet Thunder is best performer. There is a significant difference in Posenet and Movenet models. Also Move Lighting has PDJ almost 3.5% less than that of the Thunder. A higher percentage suggests better performance in accurately identifying key body points, facilitating more precise analysis of body posture and movement. Hence in PDJ, Movenet Thunder is the best and Movenet Lightning is on par with Thunder.

TABLE I
COMPARISON OF POSE ESTIMATION MODELS

Model	Posenet	Movenet Thunder	Movenet Lightning
Accuracy	97.6%	79.6%	78.7%
PDJ	87.34%	99.01%	95.42%
Average (FPS)	12	16	30

Inference Speed:
Inference speed is crucial for pose estimation models due to several reasons. Firstly, real-time applications needs to be performed swiftly. Secondly, faster inference speeds enable seamless integration of pose estimation models into live video streams and applications, enhancing user experience and responsiveness. Lastly, fast inference speeds reduce latency and processing overhead, enabling pose estimation models to keep pace with dynamic environments and capture subtle changes in human poses accurately.
Observation:
From Table IIt is observed that Movenet Lighning is twice as fast as that of the other 2 models. Faster inference speeds allow for more efficient processing of input data, enabling applications to run smoothly and respond quickly to changes in the environment. Hence Movenet Lighning is the best performing model in this aspect.
Overall Observation:
Here we can also observe that on all the metrics, Movenet Lighning model is the best performer since we can process the incoming video frames in real-time and PDJ is almost same as that of Movenet Thunder model. Based on all the above observations, Movenet Lighning is chosen for keypoint detection.

B. Results for User's score calculation

Pipeline 1: Sample images [Figures:13, 14, 15] after each block in the pipeline has been displayed. After extracting key points from images, a comparison is made among the collected key points from multiple images depicting the same pose.

TABLE II
X AND Y COORDINATES OF NOSE

Image id	**X coordinate**	**Y coordinate**
1_1	0.5243973	0.5121068
1_2	0.31955463	0.30726406
1_3	0.45065394	0.44655707
1_4	0.54897845	0.5407847
1_5	0.618625	0.618625
Standard deviation	0.11377	0.11701

Notably, the key points obtained from Movenet typically range from 0 to 1. However, an analysis reveals that the X and Y coordinates of the nose key points Table II and Figure16, exhibit a narrower range, falling between approximately 0.31 to 0.62 for X and 0.30 to 0.62 for Y. The percentage spread of these nose coordinates relative to the full range is calculated to be approximately 31% of the width and 32% of the height.

Fig. 13. Raw Frame scrapped from YouTube

Fig. 14. Frame after cropping

Fig. 15. Keypoints overlayed on the image

This indicates a relatively dispersed distribution of nose key points compared to the broader range of all key points.

The key points need to be tightly clustered for a low standard deviation among them. This low standard deviation implies that the key points are closely grouped spatially, facilitating accurate comparisons.

Pipeline 2: Sample images after each block [Figures: 17, 18, 19] in the pipeline has been displayed. We could observe for different poses, the output frame's aspect ratio keeps changing, which is not quite practical for a deployable App in which a real time inference is obtained. Also the human

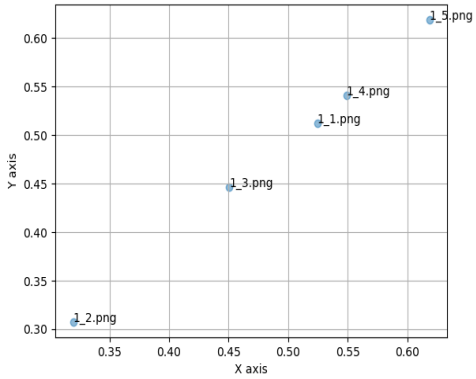

Fig. 16. Scatter plot of X&Y coordinates of nose

detection model slows down the entire inference speed to 15 FPS on an average, since two deep learning models (yolo v7 mini and Movenet Lighning) need to be run sequentially . Hence this isn't chosen.

Pipeline 3: Sample Image after calculating the angle and overlaying on the frame block (Figure 20). We can observe that the standard deviation by considering angles (last row) from Table III is far less than Pipeline 1. The average standard deviation is $16.82° \pm 0.046$, almost 20 times smaller than comparing the keypoints. Also, the angles are scale invariant. Hence Pipeline 3 is considered.

VI. System Overview

A. *Design 1*

a) Client-Side Pose Estimation: Users interact with the application, through a web browser, where TensorFlow.js is used to estimate poses directly on the edge (i.e., locally on the user's device). This involves capturing video input from the webcam and processing it in real time to estimate the pose coordinates. Once poses are estimated, they are formatted into JSON objects.

b) Data Processing on AWS: These JSON-formatted pose data are then sent to AWS for further processing and analysis. A Python file will hosted on AWS will receive the JSON-formatted pose data from the client through the API Gateway. The Python file will process the incoming JSON data, by comparing it with poses of the trained professionals which are stored in a AWS S3 bucket. The python will calculate a score and calculate the angles of each limbs. After processing the data, the Python script may generate a response, which could be another JSON object containing the results of the processing. The results will also be stored in another AWS S3 bucket.

c) Client-Side Display and Interaction: Finally, the client-side application receives the processed data from AWS and displays it to the user. This could involve showing scores, providing feedback on pose accuracy, or any other relevant information based on the processed results. HTML, CSS, and JavaScript will be used to build the front-end components of

TABLE III
ANGLES AT EACH JOINT FOR 9 DIFFERENT PERFORMERS IN DEGREES

Image id	LSA	RSA	LEA	REA	LHA	RHA	LKA	RKA
User 1	175.0749	167.0273	13.50285	17.94246	48.10776	46.21558	0.363156	4.984877
User 2	152.0785	154.7672	31.26617	27.57014	50.98107	48.59726	6.932171	13.96445
User 3	157.9365	159.6632	16.41677	54.92526	2.109885	5.660362	7.216592	5.641078
User 4	135.559	143.6446	54.92095	56.03935	58.04096	56.80019	0.29342	5.018016
User 5	175.7142	173.264	25.71696	12.24751	40.28483	35.14866	7.830853	10.30246
User 6	167.5516	177.6029	13.61358	5.051599	14.77929	27.03008	4.547216	7.36274
User 7	177.901	176.1963	29.15646	37.9231	33.8497	35.01073	25.92639	32.2598
User 8	153.4779	147.3625	53.42158	44.36261	47.48492	47.90143	10.1638	11.04892
User 9	116.3027	96.48048	20.38394	0.176939	19.79887	20.68068	19.57313	15.75843
Standard Deviation in degrees	20.54988	25.17911	15.78359	20.99289	18.92389	16.10349	8.502349	8.601051
Standard Deviation	0.057083	0.069942	0.043843	0.058314	0.052566	0.044732	0.023618	0.023892

Fig. 17. Frame after preprocessing

Fig. 18. Frame after passing through Human Detection

Fig. 19. Keypoints overlayed on the image

the UI. A front-end framework like React.js, or Angular is chosen to develop dynamic and interactive UI elements.

However due to lack of compute this design hasn't been proceeded with and this app could only be hosted if the entire components could be hosted and deployed online. Hence another architecture has been developed.

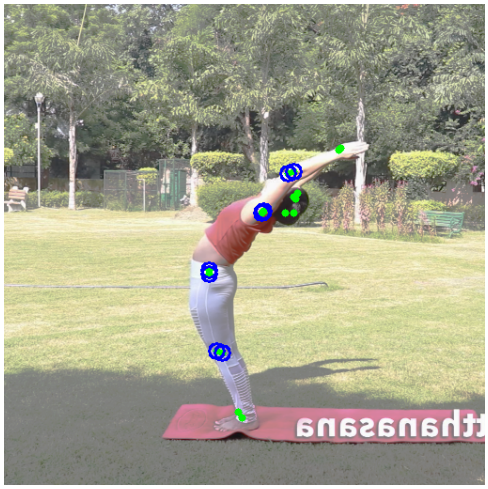

Fig. 20. Angles and Keypoints overlayed on the image

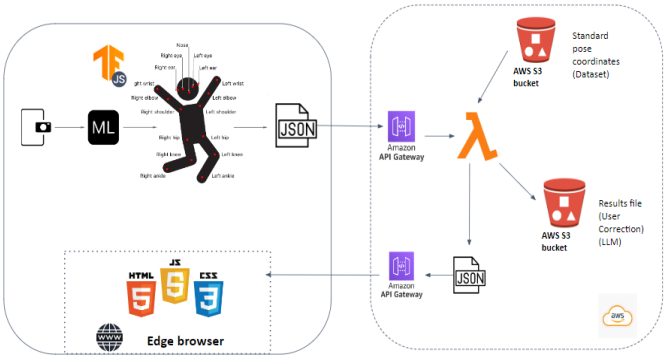

Fig. 21. Proposed System Architecture

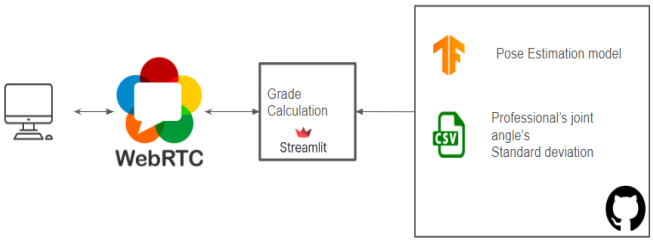

Fig. 22. Final System Architecture

B. Design 2

The system (Figure 22) comprises several key components working together to deliver yoga pose feedback:

- **WebRTC Capture:** WebRTC (Web Real-Time Communication) is a web browser technology that enables real-time audio and video communication. In this system, WebRTC acts as the bridge between the user's webcam and the pose estimation model. The user accesses the system through a web browser, and WebRTC facilitates the capture of their yoga pose video stream.
- **Streamlit Deployment:** Streamlit is a Python framework specifically designed to simplify web application development. It allows for the creation of user interfaces built from Python code. This system leverages Streamlit to deploy the pose estimation model and present the user's pose assessment results in a user-friendly web application format.
- **Video Preprocessing:** The raw video frames captured from the webcam might not be suitable for direct input into the pose estimation model due to resolution and format variations. The system incorporates a preprocessing step that resizes the video frames to a consistent size (e.g., 192x192 pixels) and potentially converts the format to a format compatible with the model (e.g., RGB). This preprocessing enhances the efficiency and accuracy of the pose estimation model.
- **Pose Estimation with MoveNet Lightning:** MoveNet Lightning is a pre-trained TensorFlow Lite model specifically designed for pose estimation on devices with limited resources. This model takes the preprocessed video frames as input and analyzes them to identify keypoints corresponding to various joints in the human body (e.g., shoulders, elbows, wrists, knees, ankles). The output of MoveNet Lightning is a set of keypoint coordinates for each frame, representing the user's pose in the video.
- **Grading:** To assess the user's pose accuracy, the system utilizes a reference dataset containing standard deviations for joint angles in a well-performed yoga pose. This data might be stored in a CSV file. The system calculates the user's joint angles based on the detected keypoints from MoveNet Lightning. By comparing these calculated angles with the reference values and their standard deviations, the system assigns a grade (e.g., excellent, good, needs improvement) for each joint.
- **Overall Assessment and Display:** The system doesn't simply provide individual joint grades. It also generates an overall comment on the user's yoga pose form based on the individual joint assessments. This comment offers a holistic view of the user's pose accuracy and areas for improvement. Finally, the system utilizes Streamlit to display the overall comment and individual joint grades on the web application. This allows users to receive real-time feedback on their yoga practice.

SYSTEM WORKFLOW

Here's a step-by-step breakdown of the system's workflow: label=0.

1) **User Access:** The user accesses the yoga pose estimation system through a web browser. This web browser needs to support WebRTC functionality.
2) **WebRTC Capture:** Upon accessing the system, the web browser initiates WebRTC to capture video from the user's webcam. The captured video stream is then transmitted to the server where the pose estimation model resides.

3) **Streamlit Web Application:** The server-side component, likely implemented in Python, utilizes Streamlit to create the web application interface. This interface might display instructions for the user, such as centering themselves in the webcam view and performing a specific yoga pose.
4) **Video Preprocessing:** As video frames are received from the user, the system performs preprocessing on each frame. This might involve resizing the frame to a consistent size and potentially converting the format for compatibility with MoveNet Lightning.
5) **Pose Estimation with MoveNet Lightning:** The preprocessed video frames are fed into the MoveNet Lightning model. The model analyzes each frame and outputs a set of keypoint coordinates representing the user's pose in the video.
6) **Grading:** Based on the detected keypoints, the system calculates the user's joint angles for each frame. These angles are then compared with the reference values and standard deviations stored in the CSV file. This comparison leads to the assignment of grades for each joint.
7) **Overall Assessment and Display:** The system analyzes the individual joint grades and generates an overall comment on the user's yoga pose form. This comment, along with the individual joint grades, is displayed on the Streamlit web application interface. The user can then observe their pose accuracy and receive feedback for improvement.

Screenshots[Figures: 23, 24, 25] of the working app are displayed below, Please follow this URL to launch the web application. https://namastai.streamlit.app

Here we can observe visually that each joint is color-graded based on how good or bad it is, making sure the user need to read the grades but rather just look at the dashboard. Also which joints needs a correction and how the correction needs to be done is also displayed at the bottom right.

VII. Conclusion

In conclusion, our project presents a comprehensive system for real-time feedback on yoga poses, integrating WebRTC capture, Streamlit deployment, video preprocessing, and the MoveNet Lightning model. By leveraging these components, users can receive immediate and accurate assessments of their pose accuracy, aiding in their yoga practice and enhancing mindfulness. The system's intuitive interface and holistic grading approach offer valuable insights into pose alignment and areas for improvement. Moving forward, further enhancements could include refining the grading algorithm and expanding the dataset for more diverse pose evaluations. Overall, our project contributes to the intersection of technology and wellness, fostering a deeper understanding of body awareness in yoga practice.

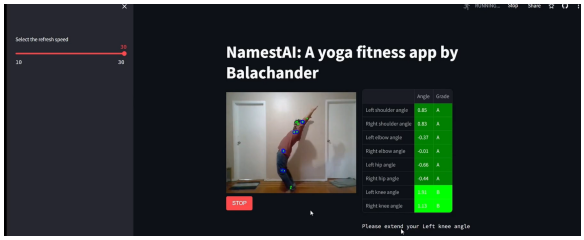

Fig. 23. Doing the correct yoga pose

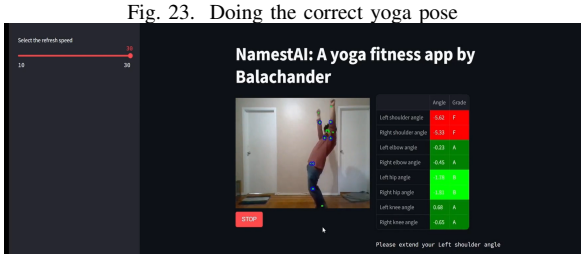

Fig. 24. Doing the yoga pose with a small deviation from the pose

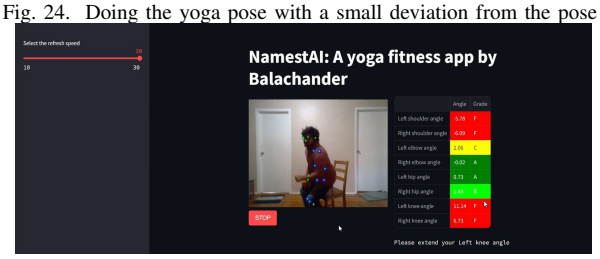

Fig. 25. Doing an irrelevant pose

Further more we can consider various pose and deploy in AWS with proper backend and frontend services, making it a full functional application.

References

[1] "Real-Time Human Pose Estimation in the Browser with Tensorflow.Js." The TensorFlow Blog, blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html.
[2] "Next-Generation Pose Detection with Movenet and Tensorflow.Js." The TensorFlow Blog, blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html.
[3] Osokin, Daniil. "Real-time 2d multi-person pose estimation on cpu: Lightweight openpose." arXiv preprint arXiv:1811.12004 (2018).
[4] Bazarevsky, Valentin, et al. "Blazepose: On-device real-time body pose tracking." arXiv preprint arXiv:2006.10204 (2020).
[5] Grishchenko, Ivan, et al. "Blazepose ghum holistic: Real-time 3d human landmarks and pose estimation." arXiv preprint arXiv:2206.11678 (2022).
[6] Dang, Qi, et al. "Deep learning based 2d human pose estimation: A survey." Tsinghua Science and Technology 24.6 (2019): 663-676.
[7] Bolaños, Cristina, et al. "A comparative analysis of pose estimation models as enablers for a smart-mirror physical rehabilitation system." Procedia Computer Science 207 (2022): 2536-2545
[8] Zheng, Ce, et al. "Deep learning-based human pose estimation: A survey." ACM Computing Surveys 56.1 (2023): 1-37.
[9] Wang, Jinbao, et al. "Deep 3D human pose estimation: A review." Computer Vision and Image Understanding 210 (2021): 103225.
[10] Munea, Tewodros Legesse, et al. "The progress of human pose estimation: A survey and taxonomy of models applied in 2D human pose estimation." IEEE Access 8 (2020): 133330-133348.
[11] Xu, W., Chatterjee, A., Zollhöfer, M., Rhodin, H., Mehta, D., Seidel, H.-P., & Theobalt, C. (2018). MonoPerfCap: Human Performance Capture From Monocular Video. ACM Trans. Graph., 37(2), 27:1-27:15. doi:10.1145/3181973

[12] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September

[13] Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014, June). 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[14] Andriluka, M., Iqbal, U., Insafutdinov, E., Pishchulin, L., Milan, A., Gall, J., & Schiele, B. (2018). Posetrack: A benchmark for human pose estimation and tracking. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5167-5176).

[15] Akshaya Agnes. (2021, August 10). SURYA NAMASKAR — 12 Rounds Of Sun Salutation — Step By Step Yoga Guide For Beginners [Video]. YouTube. https://www.youtube.com/watch?v=YAq_oCjnkWY

[16] Bharti Yoga. (2020, October 2). Surya Namaskar Step by Step — Sun Salutation with correct Breathing and Alignment — Bharti Yoga [Video]. YouTube. https://www.youtube.com/watch?v=MvdgkoqfSUU

[17] HimYogi Arjun. (2021, June 21). 108 Surya Namaskar — Sun Salutations — Daily Routine For 7 Day 100% Loss 2-3 inches [Video]. YouTube. https://www.youtube.com/watch?v=4ja4OXe7d_0

[18] Fit Tak. (2020, August 20). Step by step Guide to Surya Namaskar for Beginners — Sun Salutation — Fit Tak [Video]. YouTube. https://www.youtube.com/watch?v=PjwLlGXxeP4

[19] Yogalates With Rashmi. (2022, July 10). Surya Namaskar with Mantras — 12 Sun Salutations — Yogalates with Rashmi [Video]. YouTube. https://www.youtube.com/watch?v=Wls_EQLxOMo

[20] YogBela. (2020, March 29). Yoga at Home - Day 1- Surya Namaskar 27 rounds — 10 days of transformation — Yogbela [Video]. YouTube. https://www.youtube.com/watch?v=EIqsDkpzOgo

[21] Yogalates With Rashmi. (2018, September 30). Cardio Yoga Workout — 12 Rounds of Sun Salutations — Surya Namaskar — Yogalates with Rashmi [Video]. YouTube. https://www.youtube.com/watch?v=YrIqX5FkctA

[22] Shilpa Shetty Kundra. (2021, March 17). Guide to Suryanamaskara — The Art of Balance — Shilpa Shetty Kundra [Video]. YouTube. https://www.youtube.com/watch?v=aJb1AWMc-64

[23] VENTUNO YOGA. (2023, July 1). How to correct sun salutation technique — Improve your Surya Namaskar practise — [Video]. YouTube. https://www.youtube.com/watch?v=yunARv9Kn54