Miss Yoga: A Yoga Assistant Mobile Application Based on Keypoint Detection

Renhao Huang, Jiqing Wang, Haowei Lou, Haodong Lu, Bofei Wang
School of Computer Science and Engineering, University of New South Wales, Australia
{renhao.huang, jiqing.wang, haowei.lou,haodong.lu,bofei.wang}@student.unsw.edu.au

***Abstract*—This paper demonstrates a Yoga assistant mobile application based on human-keypoints detection models, which imitates the scene that real Yoga tutors guide and supervise their students to do Yoga via the video chat. In order to provide humanize, safe and convenient service, the core function is designed as hands-free using voice service, and embedding fast and accurate models to detect keypoints and calculate the scores. In addition, we propose an improved algorithm to calculate scores that can be applied to all poses. Our application is evaluated on different Yoga poses under different scenes, and its robustness is guaranteed.**

I. Introduction

As the accelerating pace of modern life, many students, people with jobs and elders prefer exercising at home to a GYM to avoid the limitations from the location, time or privacy. To meet these requirements, many fitness applications, such as Keep (**https://www.gotokeep.com**), have become popular during these years and provide services including professional training guidance, supervision and communication platforms. Users can follow the training steps, via videos or literal statements, to complete their home-training plans.

Yoga, as a category of light-weight exercises, can benefit the physical and psychological health with little demand on physical strength and locations requirement. It is suitable for people of all ages. However, trainees with no scientific guidance from the Yoga specialist are likely to ignore some important pose details, and nonstandard Yoga poses may result in bad harm on the bones and joints. Therefore, a mobile application with "real supervision" is helpful for these trainees who decide to train at home. Unlike those applications that provide the instructions only, the "real supervision" requires the mobile application to "watch" trainee's poses and provide them with warning once they are posing incorrectly. In order to imitate a real Yoga teacher who is watching the trainee doing Yoga, the mobile application needs to illustrate steps, analyse the postures in real-time, evaluate and provide correct suggestions.

Many Yoga self-assessment systems are being proposed through the years. [1] is a segmentation-based approach and uses two Kinects to analyse users' poses from both front and side. [2] provides a Yoga self-assessment system based on OpenPose[3] and utilizes an angle score calculation without using hardware. [4] is an end-to-end Yoga Pose classification technique that using OpenPose and a Recurrent Neural Network structure. It is apparent that OpenPose is highly appreciated on Yoga Pose analysis.

In this paper, We demonstrate a practical and humanized Yoga Assistant mobile application inspired by [2] but utilize an improved score calculation algorithm to adapt different Yoga Poses and appended with more humanized designs such as voice service. We also evaluated these models with different Yoga poses under different scenes to ensure its robustness.

II. Implementation

A. Human Pose Estimation

Human pose estimation tasks are to detect human skeletons' points to represent human real poses. Selecting a suitable model is essential for our application as it needs to handle cases under both simple and complex environments in real-time. OpenPose is a classical and excellent model that uses the part affinity fields and convolutional neural networks(CNN) to localize the human joints. [5] introduces the boundary box to reduce the redundant information during the detection to improve accuracy.

Although OpenPose can handle its tasks in real-time accurately, the raw version requires GPU to accelerate convolutions which causes large memory usage. [5] is limited by the performance of its boundary detection algorithm so that it is not stable in some cases. [6] develops a lightweight OpenPose and can run on the CPU but the performance is still challenged. Currently, we select raw OpenPose as our human estimation model and we left the exploration of more advanced models in future works.

B. Score Calculation

Angle Calculation: The angles α on each keypoint of the whole detected body is calculated by the equations below in order to be used for score computation in the subsequent steps.

$$\alpha = \begin{cases} |a_1 - a_2|, & \text{if } a_1 \cdot a_2 \geq 0 \\ |a_1| + |a_2|, & \text{otherwise} \end{cases}$$

where we assume there are two vectors $\overrightarrow{x_1} = (x_1 - x_0, y_1 - y_0)$ and $\overrightarrow{x_2} = (x_2 - x_0, y_2 - y_0)$ to calculate a_1, a_2 as,

$$a_1 = \arctan(\frac{y_1 - y_0}{x_1 - x_0}), a_2 = \arctan(\frac{y_2 - y_0}{x_2 - x_0})$$

Score Computation: By given the angle from user and instructor at keypoint i, the score can be computed as:

$$\Delta_i\% = \frac{\alpha_{user,i} - \alpha_{instructor,i}}{\alpha_{instructor,i}} \cdot \frac{\alpha_{instructor,i}}{\pi} \cdot 100\%$$

where, $\alpha_{instructor,i}$ represents the angle of keypoint i for the instructor and $\alpha_{user,i}$ represents the angle of keypoint i for the user. These percentage differences are scaled by the angles of the corresponding keypoints to improve the robustness of the results, especially when the angle of a particular keypoint is relatively small. In addition, the percentage differences could be either positive or negative which is used to indicate the direction that the user's body parts should move towards. The score for a particular keypoint i could be computed by:

$$score_i = (1- \mid \Delta_i\% \mid) \cdot 100\%$$

C. Voice Service

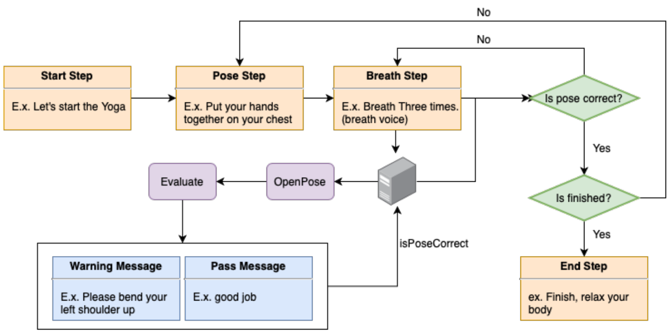

Fig. 1: The pipeline of our scenario. We have four main steps(Yellow) and each has its examples of vocal instructions and feedback.

We divide the training process into 4 main steps for all kind of poses(See fig.1): **(i) Start Step**, letting user standstill for preparation,**(ii) Pose Step**: describing the pose, **(iii) Breath Step**: asking the user to inhale and exhale, **(iv) End Step**: indicating that the training process finishes. All steps follow their pre-designed scenarios to give vocal instructions. During step **(iii)**, the mobile takes the pose photo and sends it to the host backend to complete the keypoints detection. The backend then calculates scores(see II-B), generates a detailed feedback message based on obtained evaluation scores as well as the position information of that key-point and sends them back to the mobile. All the messages are finally spoken out via the raw android text-to-speech module. To facilitate the addition of new scripts flexibly, we store these message words in a database.

D. Stability

Our application is aimed to offer high stability to improve the user experience. First, users do not need to wait for generating vocal feedback. Normally, our backend operation takes much less time than the breathing uses. In addition, our backend only sends one or two short sentences back to the mobile to avoid the internet congestion. Therefore, the most of beta testers commented that they did not notice the waiting time during the training. Our wide waiting allowance also prepares the future models to be embedded on device.

Furthermore, we also evaluated the accuracy of our application on 3 Yoga poses: Tree, Warrior I and Warrior II. Our data are collected from the internet including the dataset used in [4]. We also invited 4 people to take photos in complex scenes with different brightness(See Fig.3.). We self-drew the joint keypoints and compute the similarity with detected keypoints to assess the stability of OpenPose. Most evaluation results(See Fig.2[1] indicate that the average similarity of most cases are above 80%. Therefore, our system has the capability to give accurate feedback under different given scenes.

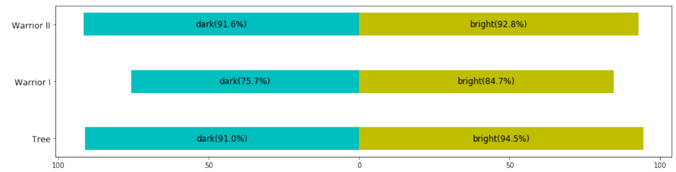

Fig. 2: Average similarity of between hand-drawn and detected keypoints for 4 people.

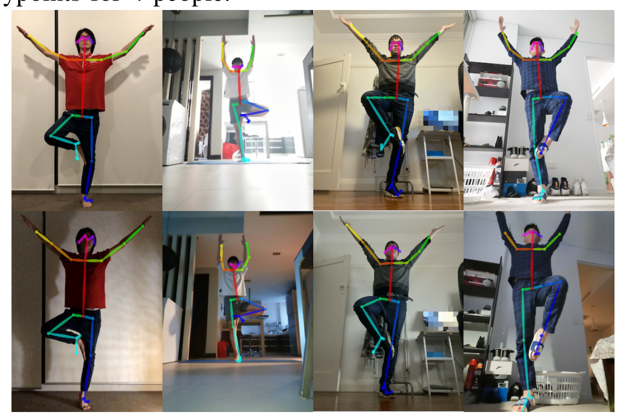

Fig. 3: Detected Keypoints of Examples in Real-Life Complex Scenes for Tree Pose(Bright(Top) and Dark(Bottom)).

III. Conclusion and Future Works

This paper proposes an OpenPose based Yoga assistant mobile application that provides users vocal instructions and feedback to correct their non-standard Yoga poses according to the scores calculated using pose skeleton. To explore more humanized functions and services, one of the improvements is to explore a more advanced model that can have three characteristics: First, it can accurately detect the keypoints in complex scenes; Second, it can be embedded on any device; Lastly, it can immediately complete calculation within 1 second. In this way, our application can run offline and provide more detailed vocal feedback. Besides, we can apply NLP models to recognize speech to realize the conversation between the application and users.

Acknowledgment

This research was supported by UNSW ChallENG Program-Vertically Integrated Projects. We thank Dr. Lina Yao and Xiaocong Chen for providing guidance and assistance.

[1]Full results: https://github.com/gym-tutor/Miss-Yoga-App

References

[1] Hua-Tsung Chen, Yu-Zhen He, and Chun-Chieh Hsu. Computer-assisted yoga training system. *Multimedia Tools and Applications*, 77:23969–23991, 2018.

[2] M. C. Thar, K. Z. N. Winn, and N. Funabiki. A proposal of yoga pose assessment method using pose detection for self-learning. In *2019 International Conference on Advanced Information Technologies (ICAIT)*, pages 137–142, 2019.

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[4] Santosh Kumar Yadav, Amitojdeep Singh, Abhishek Gupta, and Jagdish Lal Raheja. Real-time yoga recognition using deep learning. *Neural Computing and Applications*, 31:9349–9361, 2019.

[5] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017.

[6] Daniil Osokin. Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. In *ICPRAM*, 2019.