HIGH SCHOOL EDITION
Journal of Student Research

Volume 12 Issue 3 (2023)

AiShifu: AI Karate Pose Trainer Using Human Pose Estimation

Frederick Lu

Canyon Crest Academy

ABSTRACT

One application of the artificial intelligence (AI) technology is self-guided physical activities where a computing device acts as a trainer. One key challenge for these applications is how to measure the performance of such an AI trainer, especially when the AI trainer is run on a generic PC or a mobile device. In the spirit of the Turing test, an AI trainer should mimic the behavior of a human trainer. A good human trainer generally considers the training history and the level of the trainee when providing feedback, which requires more than body position analysis. In this work, we built a Martial Art trainer application called AIShifu that helps users practice martial art poses using Human Pose Estimation (HPE). We chose an open-source neural network called HRNET trained with MS-COCO dataset as the core of the HPE. The joint coordinates and angles were used to identify the pose being practiced by the trainee, whether the active side is left or right, and how close is the key joint angle to that from a “golden” image. We collected data from both a black belt martial artist and a novice trainee and on three Karate poses. Based on the data, it is clear that the blackbelt performed the poses more consistently. A much larger sample size was required to test how well an AI trainer can discern the difference between trainees with different levels of proficiency. This understanding forms the foundation to customize AI trainer software for different users.

Introduction

In most traditional martial arts, forms such as a sequence of different punches and kicks (poses) were practiced as a way to become better at certain actions. Generally, forms were practiced in a studio where students mimicked the master’s poses and the master watched over and made corrections to students’ forms. The master's attention must be spread to all students, and the practice time at the studio is very limited. Self practice would benefit students but it is difficult to obtain feedback and track progress. A computer based pose trainer would make self practice more effective and make the martial art more accessible to people in remote areas or with limited financial means.

While an ideal martial art pose or movement could be defined in theory, a human trainer also takes into consideration the training history of trainees when providing guidance. Due to variations in human physiology, the path to a “good” pose of movement might be very different for each individual. Even the goals might not be identical. The first step in understanding this diversity is to answer the question: how do joint angles extracted by HPE correlate to different training levels?

While a few AI sports software [1, 2] existed, how users of different training levels were perceived by the AI sports trainer had not been systematically studied. The studies are even more limited among traditional martial arts such as Karate. Answering this question was critical for future AI karate trainer software to adapt to different users. We developed a computer-vision based martial art pose trainer called AiShifu. AiShifu is an easy to use, lightweight system that can be run on a generic PC without additional wearable devices. AiShifu provides real time and actionable feedback to the trainee for improvement. We used AiShifu to examine a test dataset from two user groups with very different training levels and found that the angle of active arm from a black belt martial artist with more than five years of Karate training was more consistent (smaller variation) as compared with a person without any training.

Background and Related Works

One of the fastest-growing fields in computer technology was artificial intelligence, specifically neural networks. An important sub-field of artificial intelligence was neural network based Human Pose Estimation or HPE, which identifies different parts of human bodies of specific points such as shoulders and elbows. The process can be in a 2D or 3D space and sometimes through images or other data obtained devices such as sensors. HPE has received considerable attention during the past years and introduced new interest areas (applications) such as augmented reality, animation and gaming, medical rehabilitation, security and activity analysis, and sports. Due to its ability of accurately and real-time tracking of human movements, using HPE for sports was one of the fast growing applications and had many potential benefits, including injury prevention, move effective training and fitness promotion.

Despite the growing interest in this field, there is still a lack of publications regarding the specific topic of human pose estimation (HPE) applied to sports, especially martial arts. Among the available publications, literature has been focused on the benchmark metrics for performance of pose estimation accuracy and pose classification accuracy. In the area we are interested in: an intelligent coaching system for martial arts, it is recognized that the amount of available data is a specific challenge in this field.

HRNet [3, 4] (short for High-Resolution Network) follows a top-down approach for HPE and is an architecture specifically designed for obtaining accurate HPE. It was introduced to address the issue of spatial resolution loss during the network's computation. Unlike traditional CNNs that down-sampled the feature maps to reduce spatial resolution, HRNet employed a multi-resolution fusion strategy to preserve fine-grained details while still capturing high-level semantic information. The key idea was to build parallel branches, each processing the input at a different resolution. These branches were then interconnected, allowing information exchange between them at multiple scales. This design enabled HRNet to handle both global and local information effectively, resulting in superior performance. It also maintains high-resolution representations throughout the entire network. HRNet [3] achieved good performance in multiple benchmark datasets for HPE. Given its ability to maintain high-resolution representations and capture fine-grained details, HRNet can be adapted to various computer vision tasks that require precise spatial information and accurate localization. The key applications of HRNet included HPE, image segmentation, facial landmark identification, and objection detection.

Several variations of the basic HRNet architecture had been described and implemented, including Lite-HRNet [5,6] and HRNetV2 [7]. Those variations were aimed to improve performance, efficiency, or both. HRNetV2 focuses on enhancing performance while maintaining efficiency, while Lite-HRNet prioritizes lightweight and efficient deployments.

Related works

Dittakavi et al [8] described a pose trainer application based on HPE called Pose Tutor. The authors proposed to use a coarse-to-fine framework along with an angle-likelihood mechanism to achieve pose recognition and correction. Pose Tutor could be used for different exercise forms, and was “explainable” as it identified which joint contributed most to pose prediction and to correct any wrongly identified joints. As an application, Pose Tutor took an input frame and classified poses into three types (Yoga, Pilates and Kungfu). The authors also developed a graphic user interface which captures user stances with colored dots and lines denoting the critical joints and skeleton prediction from pose estimator.

Abin Aju et al [9] explored the significance of human pose estimation in the field of computer vision and presented an approach for estimating and classifying karate poses. The proposed model leveraged PoseNet, a pre-trained model, to perform accurate pose estimation, and used angle calculations between specific joints to classify the poses. The model could classify four karate stances with an accuracy rate of 98.75% using a dataset of captured images from Karate trainers. Their work included a graphic user interface which captures the user stance and provides a score of stance accuracy. Moreover, the article emphasized the breadth of potential applications for HPE across various domains, including healthcare, gaming, augmented reality, virtual training, and sports.

Fatima-Ezzahra Ait-Bennacer et al [10] provided an overview of a multiview dataset that includes pose estimations of fundamental movements performed by a karate Shotokan expert. They employed OpenPose and FastPose for detecting keypoints on the human body. The significance of this dataset was also emphasized within the realms of computer vision and deep learning. Their research primarily concentrated on implementing action detection, recognition and classification techniques, specifically employing the LSTM and Spatial-Temporal Graph Convolutional Networks (ST-GCN) algorithms. Furthermore, they proposed representing these movements in 3D by utilizing Video inference for human body pose and shape estimation (VIBE). Their experimental results indicate 96% recognition accuracy using the LSTM algorithm, and 91.01% using the ST-GCN algorithm on a dataset of captured video clips of a karate coach.

Instead of full body pose estimation, Xiaoou Zhang [11] et al described a study of the arm movement analysis of martial arts competition images, where the temporal aspect of the motion was extracted from the “bone sequence” or the position information of bones at different times. The authors created the arm movement dataset from the Wushu competition. The authors used two video channels with different frame rates to capture motions. The deep learning model used 16 frames from each video as input samples. To determine the action category to which the human action in the video belongs, a softmax classifier was employed. The test results demonstrated accuracy and recall rates of 95.477% and 92.948%, respectively, on a dataset of Wushu competition videos.

Kamel et al [12] described a training system for Tai Chi. The authors used a Convolutional Neural Network (CNN) that was specifically trained for Tai Chi poses and movements. In the experiment the user's motion was evaluated by comparing to a template. The results were presented to users to help with pose correction.

Thành et al [13] built a small video dataset and used CNN for estimating keypoints and joints of actions in traditional martial arts videos. The authors considered both 2-D and 3-D cases, and used measurements such as joint length, joint deviation angle, and keypoint deviation. The estimator was trained using the MS-COCO dataset. The authors also described an occlusion challenge where some joints would be hidden from camera views.

Two examples of commercial applications were (1) Zenia [1] was an online APP that trains beginner Yoga poses, and (2) HIT coach [2] that focused on helping users determine the power of their martial art punches / kicks and also improving the users reaction time.

System Setup and Design Details

We developed AiShifu as a martial art pose trainer that provided the user with human-like real time feedback to guide the user while practicing. AiShifu was built on a pre-trained HRNet (training & evaluation dataset was Microsoft COCO). Based on the joint coordinates, we computed all limb angles for both the left and right side of the body. In the case of arm poses like punches and blocks, the calculated angles included the shoulder and elbow angles. Pose analysis and user feedback were focused on one angle of the primary limb. For beginners to intermediate level students, it was more helpful to practice one aspect of a pose. This was typically how a human coach would train students in a group setting.

AiShifu detected the pose as well as the practicing side by comparing these limb angles to a set of "golden" or ideal pose images. This greatly enhanced the usability of the software. The database of "golden" images would be selected by a studio or human coach in typical use scenarios, so the desired practice poses and the primary practice points could be customized.

Using the angles from these golden images, AiShifu calculated the angle differences, which was used to (1) determine if the pose is good or bad based on a threshold, and (2) give directional feedback such as raise or lower arm during a punch.

AiShifu was implemented in Python version 3.9.13 and tested on a laptop with AMD R5-3500U CPU, 8GB RAM, and a built-in camera. The operating system on the laptop was Windows-11 as this was the most commonly installed OS for the target users of AiShifu. Python library "cv2" was used for camera image capture. The HPE engine was installed from the Open-mmLab v0.29.0 (October 2022) on GitHub [14]. The particular configuration used in this work had HRNet-w32 as the backbone of the neural network and used associative embedding [15] as the method (loss function) for guiding HRNet during training. This neural network model was trained and tested using the Microsoft COCO data-set [16].

AiShifu's top-level flow diagram was shown in Figure 1. AiShifu had two run-time modes: master mode allowed the "master" such as a human coach or studio to process a set of pre-recorded golden images; and student mode allowed the "student" to use AiShifu for pose practice.

In the master mode, AiShifu first searched a designated directory for golden image files. The golden image files were required to follow certain naming conventions that included information such as the name of the pose and the active side in the image. Each golden image file was loaded and run through the HRNet to extract joint coordinates. Another text file defined by the master linked the name of a pose to which joints would define the key angle of the pose. Using this information, AiShifu calculated the key angles and stored the result in another text file: the golden pose file. In the master mode, the program would terminate once all golden images had been processed.

For the current work, the golden images were obtained by selecting one picture of a black belt practicing the pose. These pictures acted as place holders for the development of the code. In real use cases, the golden image files would be curated by the master.

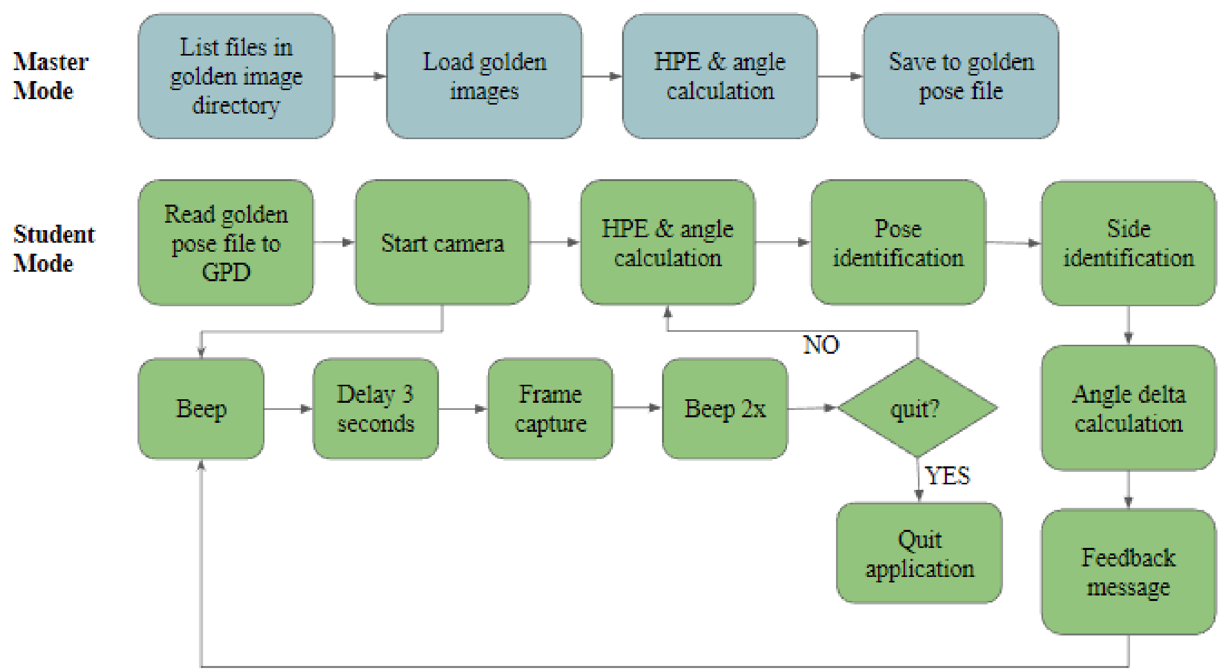

Figure 1. Top-level Flow Diagram of AiShifu. The top side showed the flow in the master mode. The master mode was meant to be executed once by an administrator. The bottom side showed the flow in the student mode. The student mode is executed during self-guided training.

In the student mode, AiShifu started by reading the golden file and storing the key angles from golden images into a Python Dictionary, the Golden Pose Dictionary or GPD (Figure 2). AiShifu then initialized the camera for video capture, and made a beep sound to inform the user that it was about to take a picture of his/her pose practice. The user was given three seconds to get ready before the frame was captured. Immediately after the frame capture, AiShifu gave two more beeps in quick succession to tell the user that an image had been captured. AiShifu proceeded with HPE (HRNet inference) and angle calculation. Limb angles were calculated from joint coordinates using basic trigonometry and placed into an array. The user could hit the "q" key to exit the program at any time.

Key	Value
Posename1	(active_side, right_angle, left_angle)
Posename2	(active_side, right_angle, left_angle)
…	…

Figure 2. Golden Pose Dictionary (GPD) structure diagram. The posename string was used as the “key”, and the value field contained three fields: the active side, the right and left side angle of the relevant limb.

Since the pose had not been identified, all limb angles were calculated. This array of angles were compared with the GPD to identify the pose being practiced. This was accomplished by finding which entry in the GPD minimizes the difference between the active limb angle of the golden pose and the corresponding angle in the captured frame. For example, if the GPD contained a "high punch" and a "side kick" poses, AiShifu determined which pose matched the captured image by comparing the shoulder angle difference to "high punch" golden pose, and the hip angle difference to "side kick" golden pose. If the shoulder angle difference to "high punch" golden pose was smaller, then the captured image was determined as being a “high punch”.

After the pose had been identified, a two-stage boolean operation was used to determine if the user was practicing the left side of the right side. AiShifu accomplished this by observing that the relation of the active side angle and the non-active side angle can be obtained from the golden pose. As shown in Figure 3a, we first compared the two angles such as left shoulder and right shoulder in the golden pose data, and linked that to the known side in GPD. We then determined the active side of the captured image based on which angle was larger or smaller. For example, if the golden image was a pose of the left-hand side and the left shoulder angle in the golden image was larger than the right shoulder, we could infer that the active shoulder angle in this pose was larger than the non-active side. So we marked the larger of the two shoulder angles from the captured image as the active side (Figure 3b). This method enabled AiShifu to accept golden images using either left or right side, and could support both left handed and right handed students.

Golden pose GPD	Left side active	Right side active
Left angle bigger	(1) Bigger angle is active	(2) smaller angle is active
Right angle bigger	(2) smaller angle is active	(1) Bigger angle is active

(a)

captured image	Left angle bigger	Right angle bigger
(1) From 1st boolean	Left is active	Right is active
(2) from 1st boolean	Right is active	Left is active

(b)

Figure 3. Two stage boolean for active side identification. 3a combines the declared active side in GPD with the angle comparison of the left and right side in the golden pose data to determine whether the bigger angle side or the smaller angle side is the active side, noted as condition (1) and (2) respectively. 3b uses the output of the first stage boolean, and combine it with the angle comparison of the left and right side in the captured pose data to determine whether the left or the right side is the active side.

After the pose and the active side had been determined, we calculated the difference of the active side key angle to the active side angle from the golden pose image, referred to as delta(Angle) or δAngle. AiShifu set two thresholds for "good" and "almost good" poses. For the initial test, we arbitrarily chose 5 degree difference as the "good" pose threshold, and 15 degree difference as the "almost good" pose threshold. The feedback decision tree was shown in Figure 4. First, the absolute value of δAngle was compared with the "good" pose threshold. If ABS(δAngle) was less than the good pose threshold, AiShifu printed the message "Perfect!", and looped back to start next pose practice. Otherwise, ABS(δAngle) was compared with the almost-good threshold. If ABS(δAngle) was less than the good pose threshold, AiShifu prints the message "You are close!", and uses the sign of δAngle to decide whether to print "Raise your arm." or "Lower your arm.". When poses involving other limbs such as legs were added in the future, the pose name would be used to modify the feedback message.

AiShifu was designed to focus on improving users pose accuracy through simple and actionable feedback. So the code did not include any features that would take into account the temporal tracking or movement of the user. Furthermore, we did not account for the possibility of more than one person in the frame as we considered that in the usage scenario of multiple students practicing together, it was best for each student to have his/her own computer running AiShifu rather than everyone sharing the same hardware. AiShifu was best used in conjunction with studio practice where pose transition and movement would be taught.

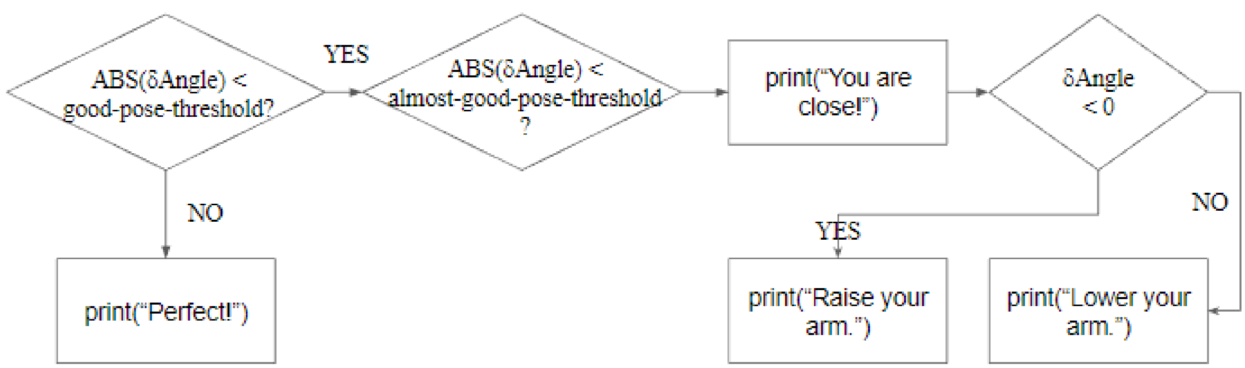

Figure 4. Feedback decision tree. First, the absolute value of the angle difference (δAngle) was compared with the good-pose-threshold and the almost-good-pose-threshold. Encouraging messages were printed based on the comparison result. Second, the sign of the dAngle was used to decide whether the user should be instructed to raise or lower the relevant limb (arm was used here as an example).

Test and Results

We picked three common poses in karate to test AiShifu. They were low block, middle punch and high punch. Note that we only chose poses related to the upper body as an initial test. Poses related to other parts of the body would be added in the future. Two participants, a black belt and a beginner, were invited to practice these three poses. We took 43-68 images for each pose and each participant. Participants varied where they stood in relation to the camera (both distance and angle) and used both their left and right side in the pose. This allowed for increased variation in the dataset. The number of images for each participant and pose were listed in the Table 1 below.

Table 1: Number of images for each participant and each pose in test data-set.

Participant	Low Block	Middle Punch	High Punch
Black Belt	43	53	50
Beginner	58	68	51

Three example images (one for each pose) and the same images with HPE joint overlay were shown in Figure 5. Note that in some images, some joints on the far side of the camera were blocked by the body and cannot be extracted in HPE. For example, the right shoulder in the Figure 5a (low block).

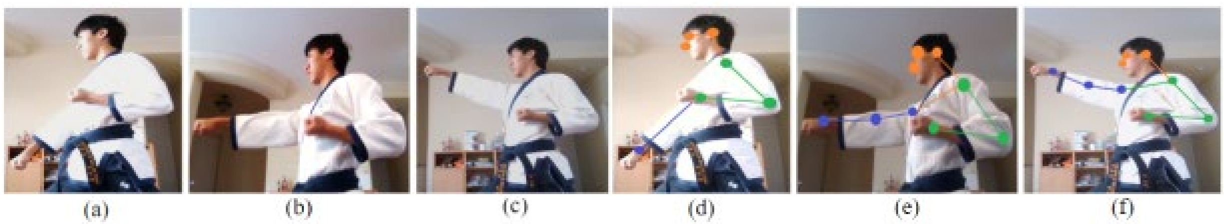

Figure 5. Example images from each pose (5a: low-block, 5b: middle-punch, 5c: high-punch) in the test data-set, and the same images overlaid with joint markers from HPE output (5d: low-block, 5e: middle-punch, 5f: high-punch).

This occlusion problem made the pose identification step less reliable. We also noted that the typical white uniform of some martial art traditions could make it difficult for HRNet to identify the joints due to the lack of any patterns in a large region of the image that is critical for joint detection. Another challenge we encountered was that some of the captured frames were too blurry for processing. In future works, we planned to add a step before HPE to determine if the captured image was clear enough to proceed. The added step could then inform the user to redo the pose if the captured image was not clear.

Figure 6 showed the raw angle data from the high punch pose before the active side was identified. A bimodal distribution was clearly visible in Figure 6a. Figure 6b showed that the sum of the shoulder angles of the two sides was roughly constant. This confirmed that the two sides were swapped in some images, and that a simple comparison of angle sizes could work as the method to determine which side was in the frame. After the active side had been identified, the active arm angle and the non-active arm angle were plotted in figure 6c. The standard deviation of the right arm angle without identifying the practice side was 23 degrees, while after the correct active side had been detected, the standard deviation for the active side angle reduced to only 11 degrees.

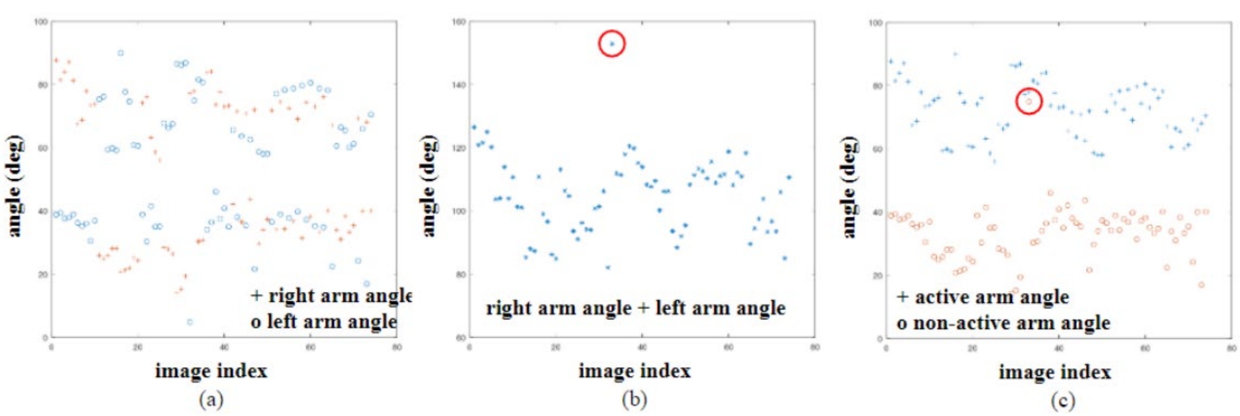

Figure 6. Raw arm angle for the pose "high punch". 6a were angles of the right arm and the left arm, and showed a clear bimodal distribution. To confirm the hypothesis that this was due to participants switched sides in some of the

images, the sum of the right arm angle and the left arm angle was plotted in 6b. The single modality of the sum confirmed the proposed active-side identification logic would work. 6c plotted the arm angle data after the active-side identification logic and matched the expected result. Note there was an outlier data point (marked with red circle), where both arm angles were higher than 50 degrees (78 and 75 degrees). This was a case where the HPE engine did not identify joint coordinates correctly.

The relevant arm angle for the three poses by the two participants were summarized in Figure 7. It is clear that the mean value of the black belt's active arm angle was very close to that of the golden image, even though the black belt arm angles had larger variation than that of the beginner. This observation needed to be confirmed with more participants and more poses, and would be used in future work to enhance the user feedback.

Table 2 below showed the percentage of images within the good-pose threshold (5 degrees) of each participant and each pose. It was evident that the black belt consistently performed better than the beginner in this metric as well. The black belt achieved 45-74% of good poses, while the beginner only achieved 0-32% of good poses.

Table 2: Percentage of good poses in test data-set by each participant and each pose.

Pose\Participants	Black Belt	Beginner
Low Block	45%	8%
Middle Punch	49%	0%
High Punch	74%	32%

The current code took about 8-10 seconds to execute each frame on the test computer. To estimate the execution speed requirement, we conducted use case simulations. On average, it took a user between 3-6 seconds to get back in front of the test computer after performing a pose. So the current code would require a user to wait 2-7 seconds of wait time before displaying the feedback. It was desirable to achieve faster execution to make AiShifu more responsive.

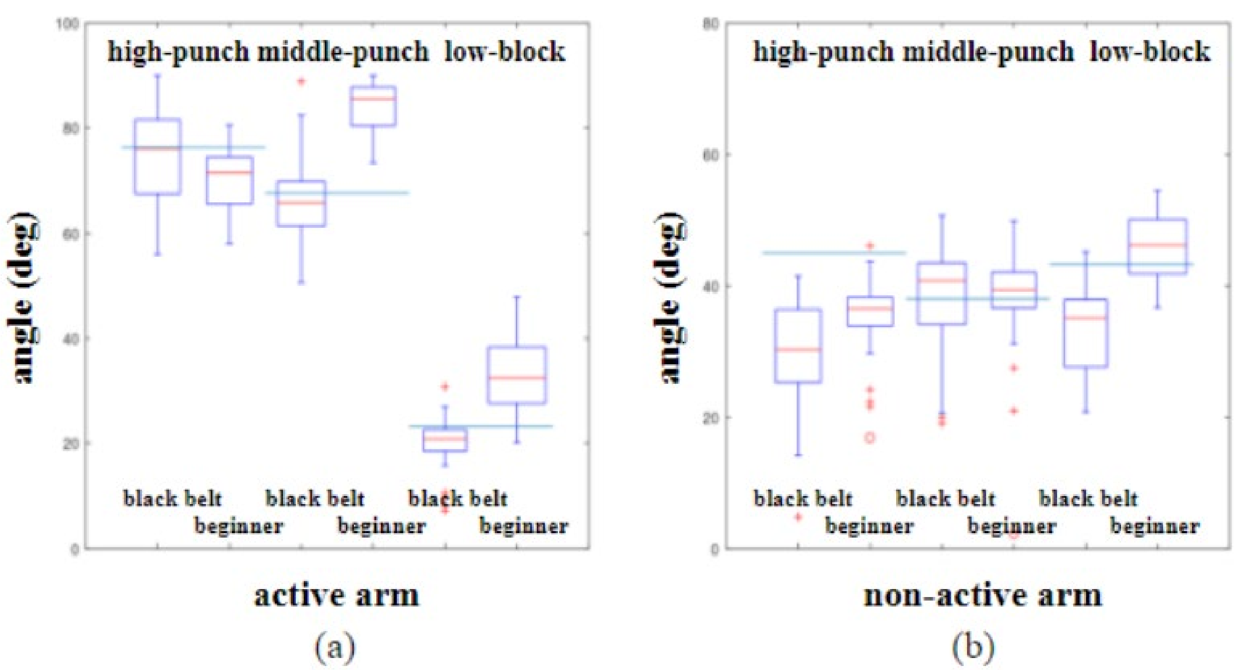

Figure 7. Box-and-whisker plot of the active 7a and nonactive 7b arm angles in test dataset. The arm angles were split by the three poses and the two participants with different skill level. The corresponding arm angles from the golden poses were indicated with the green lines.

Discussion and Future Works

AiShifu was designed as a computer application that used state-of-the-art HPE to improve at-home martial art practice by providing real time feedbacks and giving directional guidance to the user. Applications for static pose training were not limited to martial art forms. Many other sports such as shot put also require complex body posture training. Beyond sports, many performing arts such as dance could also be benefited from self-guided pose practice to supplement formal training. Also, there were many medical applications as well. For example, physical therapy involved certain body or limb positioning and needed to strike a balance between effectiveness and risk of over stressing. AiShifu could help physical therapists provide better guidance to patients at home.

Using a test data-set with three common Karate poses, our test results showed that AiShifu can detect a clear difference in accuracy between the more experienced black belt and the less experienced beginner. To further validate this, a much larger dataset that includes more trainees at different stages of training is required; and a more comprehensive statistical analysis of the joint angle data needs to be performed.

"White uniform" and arbitrary occlusion were well known challenges to HPE in general. While continued improvement of neural-network such as HRNet could be expected, we believed it is possible to find workaround such as more optimal camera and light source placement to improve the success rate of current and future HPE Algorithms. Incorporating some aspects of human anatomy assumptions into AiShifu could also improve the performance of pose detection. Another common challenge for HPE was related to a complex and cluttered background. For our application, the test images were taken with a typical home background with furniture and random household items. This was to mimic the background of the expected usage setting. We did not notice any problems related to image background. This aspect needed to be further tested in the future.

The current execution speed utilizing only CPU was marginally adequate. Considering a typical user might be running other workloads on their PC while using AiShifu, which will cause AiShifu to execute slower, future work was required to make AiShifu execute faster. For this purpose, we considered the following options:

1. Switching to faster neural-networks such as Lite-HRNet. The performance of these models and their execution speed on target hardware setups needed to be carefully checked. It was likely that an optimal balance between execution speed, hardware requirements, and HPE accuracy could be found.

2. Compiling the source code for common processors and operating systems. This would optimize the executable code and reduce the workload at run time.

3. Adding a module to precondition the frame. For example, reducing the image resolution (current resolution is 1280x720) might improve execution speed without significant loss of limb angle accuracy. Also, the user experience could be enhanced if blurry images could be quickly identified so AiShifu could request the user to try again.

4. Batch processing. Processing multiple images through the HPE engine could dramatically increase the throughput of code, even for a serial processor like CPU. Even though the current usage model did not lend itself well to batch processing, collecting a few images in sequence and then running HPE computation together could be an effective method to reduce the blurry image problem or increase the performance of joint identification through a majority voting or similar logic.

Conclusion

We designed an AI enabled Karate pose trainer called AiShifu. AiShifu had two run-time modes, and could successfully run on a generic computer. Based on the initial test, AiShifu could perform the designed functions including pose and active side identification, and give correct feedback to a user based on a specific pose the user was practicing. AiShifu could also detect a difference in user experience level. Further analysis on larger dataset could provide more detail on the development paths of martial art practitioners, and open the door to training history based personalization of future applications. Compared with existing applications such as Zenia [1], AiShifu provided simple directional feedback so users would know which way to correct their last attempt. While AiShifu was impacted by some challenges of the state-of-the-art HPE models, sufficient performance was obtained with some workaround techniques.

Acknowledgments

I would like to express my sincere gratitude to my teacher Mr. Anthony Mauro and my mentor Mr. Ross Greer for their invaluable guidance and mentorship throughout the course of this research. Their expertise and support have been instrumental in shaping the direction of this study.

References

1. https://zenia.app. Yoga trainer, Retrieved on March 8, 2023
2. https://www.hit.coach, Retrieved on March 8, 2023.
3. Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.5693-5703.
4. https://github.com/HRNet. Retrieved on March 8, 2023.
5. Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L., Sang, N., & Wang, J. (2021). Lite-hrnet: A lightweight high-resolution network. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10440-10450).. https://doi.org/10.1109/cvpr46437.2021.01030
6. https://github.com/HRNet/Lite-HRNet Retrieved on May 30, 2023
7. Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., & Wang, J. (2019). High-Resolution Representations for Labeling Pixels and Regions. arXiv:1904.04514. https://arxiv.org/pdf/1904.04514.pdf
8. Dittakavi, B., Bavikadi, D., Desai, S. V., Chakraborty, S., Reddy, N., Balasubramanian, V. N., Callepalli, B. & Sharma, A. (2022), Pose tutor: An explainable system for pose correction in the wild, in ‘Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops’, pp. 3540– 3549.
9. Aju, A., Mathew, C., & Prakasi, O. S. G. (2022). PoseNet based Model for Estimation of Karate Poses. Journal of Innovative Image Processing, 4(1), 16–25. https://doi.org/10.36548/jiip.2022.1.002
10. Ait-Bennacer, F.-E., Aaroud, A., Akodadi, K., & Cherradi, B. (2022). Applying Deep Learning and Computer Vision Techniques for an e-Sport and Smart Coaching System Using a Multiview Dataset: Case of Shotokan Karate. International Journal of Online and Biomedical Engineering (iJOE), 18(12), 1-14. https://doi.org/10.3991/ijoe.v18i12.30893
11. Zhang, X., Wu, X., & Song, L. (2022). Arm Movement Analysis Technology of Wushu Competition Image Based on Deep Learning. Computational Intelligence and Neuroscience, 2022, 9866754. https://doi.org/10.1155/2022/9866754
12. Kamel, A., & Liu, B. L. P. a. S. B. (2019). An investigation of 3D human pose estimation for learning Tai Chi: A human factor perspective. International Journal of Human-Computer Interaction, 35(4-5), 427-439.
13. Thanh, N. T., Hung L. V., & Cong, P. T. (2019). An evaluation of pose estimation in video of traditional martial arts presentation. Research and Development on Information and Communication Technology, 2019(2), 1-8. https://doi.org/10.32913/mic-ict-research.v2019.n2.864

14. Contributors, M. (2020). Openmmlab pose estimation toolbox and benchmark.https://github.com/open-mmlab/mmpose
15. Newell, A., Huang, Z., & Deng, J. (2017). Associative embedding: End-to-end learning for joint detection and grouping. Advances in neural information processing systems, 30.
16. 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 6-12, 2014, Proceedings, Part V 13 (pp. 740-755). Springer International Publishing.