





Development of Kendo Motion Prediction System for VR Kendo Training System

Yuki Saigo¹, Sho Yokota¹^a, Akihiro Matsumoto¹^b, Daisuke Chugo²^c,
Satoshi Muramatsu³ and Hiroshi Hashimoto⁴^d

¹Dept. of Mechanical Engineering, Toyo University, Saitama, Japan

²School of Engineering, Kwansai Gakuin University, Sanda, Japan

³Dept. of Applied Computer Eng., Tokai University, Hiratsuka, Japan

⁴Adv. Institute of Industrial Tech., Shinagawa, Japan

Keywords: Sports Training, Machine Learning, Human Motion Prediction, Recurrent Neural Network.

Abstract: In this study, we developed and evaluated a system within the system to predict the user's Kendo (Japanese fencing) motions which is the function of the VR Kendo system that enables easy Kendo training at home or in similar settings. We utilized markerless motion capture and machine learning based on recurrent neural networks (RNN) to learn and predict kendo motions. As a result, the proposed system successfully predicted Kendo motions as it started with high accuracy.


1 INTRODUCTION


Training is crucial for improving skills in any sport. Sports training typically requires a designated location and training partners. Depending on the sport, some activities are easy to train, and others are not. An example of a sport that is easy to train is long-distance running. Long-distance running can be trained alone near one's home, such as on nearby roads or tracks.


On the other hand, kendo is one of those sports that are not easy to train without a training partner. Kendo is one of the Japanese martial arts in which players wear protective gear and use bamboo sword to fight each other, namely, it is "Japanese fencing". The purpose of kendo is to strengthen the body and mind and to build moral character through continuous training (All Japan Kendo Federation, 2023). In the case of kendo, there must be at least two persons wearing training uniforms and protective gear and holding bamboo swords. Furthermore, an indoor facility with sufficient space to swing the bamboo sword is required. Therefore, it is possible to overcome space constraints by utilizing a virtual


environment (VR). Additionally, if training partners can be provided as virtual agents within the virtual space, engaging in easy kendo training at home or similar locations would be possible. In particular, in actual Kendo training, the process of players mutually predicting each other's motions(techniques) is crucial. We think more practical training is possible if the agent confronting the user in the virtual environment can predict the technique's motion when the user performs it. Therefore, in this study, we develop the Kendo motion prediction system as a first step toward realizing this system. In particular, this paper proposes a method for predicting the type of techniques at the moment of its execution, i.e., the moment when a user performs a "Men", "Kote", "Dou" or "Stance" motion.

There have already been several studies on kendo training systems. A method for predicting kendo movements using GMM from body and bamboo sword motion capture (Y. Tanaka, K. Kosuge, 2014), a training system that provides feedback on kendo movements using IMU (M. Takata, Y. Nakamura et al., 2019), and a system that predicts kendo movements using machine learning and markerless motion capture from information obtained from two

^a <https://orcid.org/0000-0002-8507-5620>

^b <https://orcid.org/0000-0002-3004-7235>

^c <https://orcid.org/0000-0002-3884-3746>

^d <https://orcid.org/0000-0003-2416-8038>

high-speed cameras (Cao, Yongpeng, Yuji Yamakawa, 2022). On the other hand, this research aims to develop an easy and practical VR kendo practice system using a small bamboo sword-type controller and a motion acquisition system using only one camera in pursuit of convenience.

2 CONCEPT OF VR KENDO TRAINING SYSTEM

A concept of the proposed VR Kendo training system is shown in Figure 1. The user wears a Head-Mounted Display (HMD) and holds a small VR controller that resembles a bamboo sword. The user performs kendo motions in front of a camera that measures the user's motion. Within the user's field of view in the HMD, a virtual opponent is standing in front of them. The user trains kendo with the virtual opponent in the VR environment.

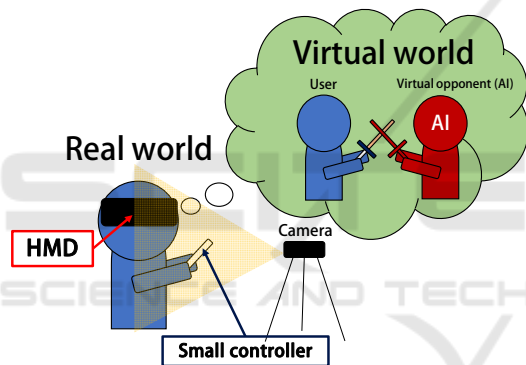


Figure 1: Concept of VR Kendo training system.

Generally, it is desirable for the training environment of a sport to resemble the actual competition environment as closely as possible. However, practicing VR Kendo with a regular bamboo sword requires enough space to swing the sword freely. Therefore, a VR-specific miniaturized bamboo sword is required, shorter in length than a regular bamboo sword but still provides a similar sense of swing. Furthermore, in Kendo training, it is essential to have one-on-one training that simulates a real match scenario. Both players explore each other's motion, predicting each other's technique, and determining how to react accordingly. This training is crucial for enhancing skills in Kendo. Therefore, to conduct more practical kendo training within the VR environment, the virtual opponent should be able to predict the user's motions. This paper proposes a motion prediction system.

Two essential components are required to realize a motion prediction system for kendo: a motion capturing system for player's motions and a motion prediction system to predict future motions based on the captured data.

2.1 Motion Acquisition System Using OpenPose

One of the most effective methods to obtain human motion is a motion capture system using markers to the human body. Considering the purpose of this study, this motion capture method is not suitable. Putting markers on the body takes time and effort and is far from easy. Therefore, this study used OpenPose (Zhe Cao et al., 2017) to obtain kendo motion.

OpenPose is a method for estimating posture by extracting the joints of humans using deep learning. Input the video images, and it can detect 25 joint points in 2D space. Figure 2 shows the extracted joint points using OpenPose. OpenPose is a markerless motion capture system that allows us to easily obtain kendo motion with only one camera.

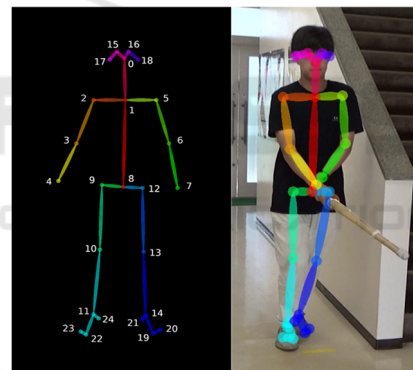


Figure 2: Extraction of joint points of a person holding a bamboo sword by Openpose.

2.2 Motion Prediction System Using Machine Learning

To predict Kendo motion from the user's motion obtained OpenPose, it is required to recognize Kendo motion and the ability to process time series information.

In this study, we used machine learning for these requirements, and we decided to use recurrent neural networks (RNNs), which are particularly good at processing time series data. In this study, we develop a user's kendo motion prediction system that combines OpenPose and RNN to realize an easy and practical Kendo training system.

3 MACHINE LEARNING WITH RNN

In this section, we explain the RNN used for machine learning for Kendo motion prediction and more details of the machine learning with data obtained from OpenPose as input.

3.1 Overview of RNN (and LSTM)

Figure 3 compares a typical feed-forward neural network with an RNN model. Generally, neural networks transmit information in only one direction, from input to output, however, RNNs have a loop structure in the middle layer that sends output results to itself. Figure 4 depicts an expanded loop structure. Where x is the input and h is the output, this makes it possible to use past information as new input to itself. Thus, each input can be treated as a series of input data rather than independently, making it possible to process time-series data.

However, traditional RNNs have limitations in retaining long-term temporal information, leading to the loss of sequential patterns when processing data. Therefore, in this study, we use Long Short-Term Memory (LSTM) networks developed to improve RNN (Hochreiter.S, Schmidhuber. J, 1995). Since LSTM can retain long-term data dependencies, it is assumed to be able to estimate the user's unique behaviour patterns. Therefore, the user and agent interaction could be more active. As mentioned above, the input to LSTM is not unit data but a series of input data holding time-series information, i.e., a collection of multiple data generally represented as a three-dimensional array, as shown in Figure 5.

The first dimension (horizontal) represents the features of the data, the second dimension (vertical) represents the time steps of the sequence, and the third dimension (depth) represents the number of data.

The larger the feature dimension, the more information the data will have. Similarly, the greater the number of time steps, the greater the processing capacity is required, and the greater the number of data the greater the amount of input data. However, since the third dimension is just the quantity of data, the first and second-dimension elements are computed in LSTM. Therefore, it is essential to tune properly the features and time steps for learning. Next, we will discuss the data obtained using OpenPose.

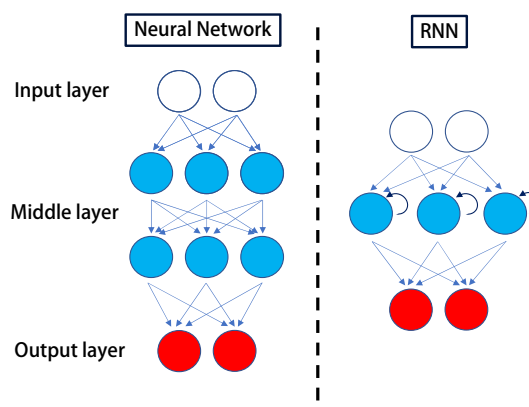


Figure 3: Comparison of Neural Networks and RNNs.

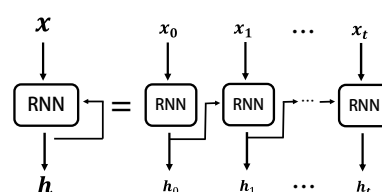


Figure 4: Deployment of RNN loop structure.

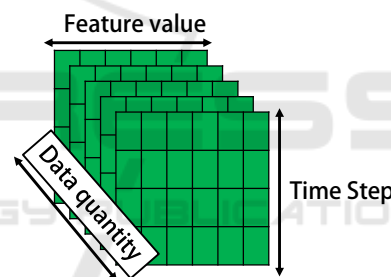


Figure 5: Image of input data shape.

3.2 Input Data from OpenPose

As described in Section 2.2, OpenPose can extract 25 joint points from a person in the input video and store each joint point's x and y coordinates for each frame. Hence, 50 points of coordinate information can be obtained from each frame, and the same number of coordinate information is stored for the video frames. Figure 6 shows an overview of the data available from OpenPose.

The number of features in this study is 50 because the coordinate information in each frame may contain critical information. All joint coordinates are expressed in relative coordinates to the midpoint of the person's waist as origin, and 50 data points expressed in relative coordinates were treated as features. Since the information handled by OpenPose is two-dimensional, a simple change in the positional relationship between the camera and the person may

interfere with the acquisition of coordinate information for the joints in motion.

In the time step, we used a few frames of a part of the video. The reason for not treating the entire video (meaning through start to end of motion) frames as a time step is that the purpose of the research is motion prediction, and if it can't be classified by a part of the motion, it is meaningless. In other words, if the entire video frames are used as a time step, the motion can be classified (predicted) only after the technique is completed.

The data quantity was calculated by dividing the total frames of the video by the number of time steps. Since the number of time steps is arbitrary, the amount of data depends on the size of the number of time steps. If there are many videos, the frame information of all videos is combined and divided by the number of time steps.

3.3 Machine Learning

There are four kendo motions to be learned: "Men", "Kote", "Dou", and "Pose". "Men" refers to the head strike, "Kote" refers to the right wrist strike, "Dou" refers to the right abdominal strike, and "Pose" refers to the stance motion. Figure 7 shows examples of each motion.

There are two reasons for the addition of "Pose" that is not technique. The first is that in kendo, the time spent in stance is longer than the time spent performing techniques. The second reason is that the information that a player is not performing a technique is necessary to predict the start of a technique.

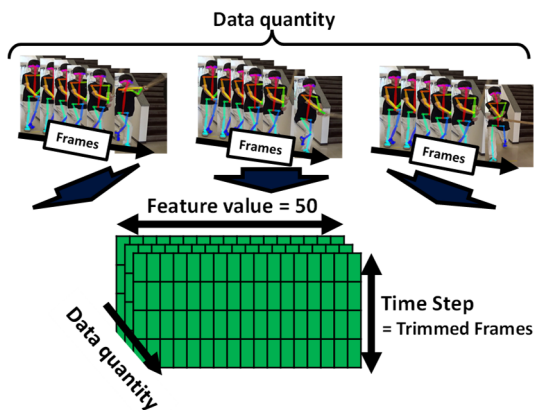


Figure 6: Overview of data obtained from OpenPose.



Figure 7: Examples of each Kendo motion.

3.3.1 Train Data

The video was captured during each motion and analyzed with OpenPose to obtain motion data. Table 1 shows the number of frames and the number of data in the captured video.

Because the average number of frames per video is short, the number of frames used as a time step was set to 5 frames for testing purposes. Therefore, the shape of each dataset used for training is shown in Table 2.

Table 1: Details of videos used for training data.

	Men	Kote	Dou	Pose
Number of videos	51	56	50	30
Average frames per video	86	71	112	140
Total frames	4406	3973	5613	4223
Data quantity (Time step = 5 frames)	881	794	1122	844

Table 2: Shape of each training data.

	Men	Kote	Dou	Pose
Shape of train data	(881, 5, 50)	(794, 5, 50)	(1122, 5, 50)	(844, 5, 50)

3.3.2 Neural Network Model for Learning Kendo Motion

Figure 8 shows a diagram of the neural network model used to train the kendo motions. To classify the input data into four motions, the shape of the output data is as follows (number of data, 4). The system also has two LSTM layers. At the same time, a one-layer LSTM is limited in the range of information because of limited time steps; layering increases expressive power and enables the capture of complex patterns within a limited time step.

This idea is supported in natural language processing (Sutskever, I et al., 2014). We also incorporated it in this study since the time step is limited to five frames. However, since LSTM can only process two-dimensional data of time steps and features at a time, the model is repeated as many times as the number of data, with input and output at each time step, as shown in Figure 9.

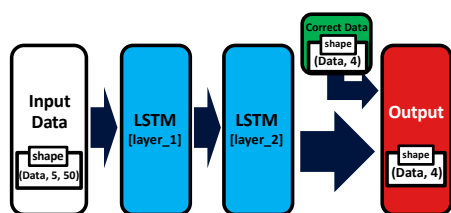


Figure 8: Entire Neural Network.

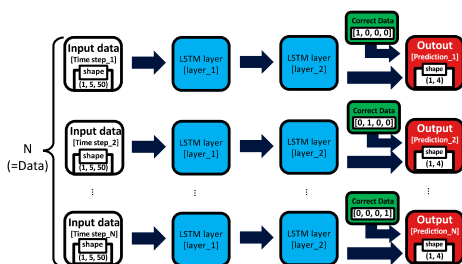


Figure 9: Details of the entire Neural Network.

The activation function used in the output layer is the Softmax function commonly used in multi-class classification problems; the values calculated from the Softmax function are real numbers between 0 and 1, and the sum of the values of the four elements of the output array is always 1. The predicted value of which motion is likely can be calculated as a probability value, and the class of elements with the highest value is the predicted result.

The teacher data has the same data shape as the output data and is input using one-hot encoding, where the relevant element is set to 1 and the others to 0, and the model trains using cross-entropy error.

3.3.3 Learning Kendo Motion

We searched for the optimal hyperparameters to be used in learning. There are two methods for searching hyperparameters: grid search and random search. Grid search conducting a full search is more effective (Bergstra, J. et al., 2012). Table 3 shows the hyperparameters and optimal parameters searched by grid search. The number of nodes in the second LSTM layer is half that of the first layer to suppress overlearning.

Figure 10 shows the training results with the training data and hyperparameters presented in Tables 1, 2, and 3, where “Accuracy” represents the percentage of correct answers and “Loss” indicates poor performance. In learning, the model is evaluated in real-time using newly prepared validation data (data quantity is about 1/4 of the training data), which is separate from the training data. To control overlearning, we used “early stopping”, which

automatically terminates learning if no loss of progress is observed in a certain epoch period.

As Figure 10 shows, in this learning process, the learning is done accurately because similar trends are observed in the training data and the validation data as the learning progresses.

Table 3: Parameters explored and optimal results.

	Search target	Optimal combination
1st LSTM layers unit	(256, 512, 1024, 2048)	1024
2nd LSTM layers unit	Half of 1st layer	512
batch size	(32, 64, 128)	32
epoch	(100, 200, 400)	200
activation	only Softmax	Softmax
optimizer	(SGD, Adam)	Adam

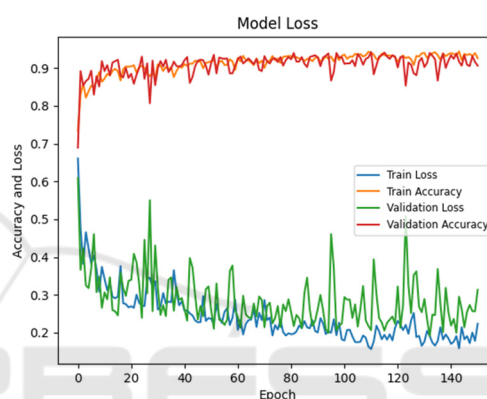


Figure 10: Learning Transition.

4 EXPERIMENTS AND DISCUSSION

We experimented to evaluate whether the LSTM model trained in the previous section could predict (classify) each of the four Kendo motions.

Apart from the training and validation data, the prediction performance was evaluated using data obtained from joint coordinates by OpenPose from newly captured Kendo motions for the prediction experiment. In training, the motion data of four techniques were inputted together. In this experiment, data obtained from a single video showing a single technique being recorded is used, assuming the data is used in real-time. Similarly, assuming real-time prediction, data is inserted into the model one frame at a time, as shown in Figure 11, and inference is performed using the latest five frames as input data.

We conducted two types of experiments. The first was an evaluation of each of the four kendo motions alone, and the second was an evaluation of motion in Pose in combination with the other motion.

4.1 One Motion Experiment

As in the training phase, the inferential actions were from the start to the end of a single technique. The motions of each technique were tested ten times and evaluated based on the percentage of correct answers compared to the prediction output by the model. Table 4 shows the accuracy of each motion. The accuracy is “the total number of correct responses / the number of input frames” per input frame.

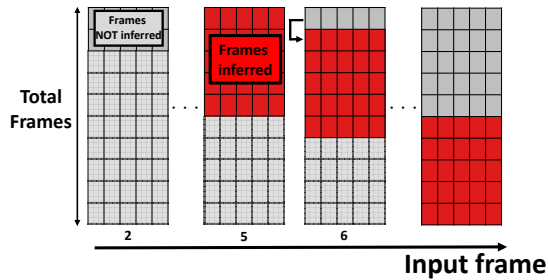


Figure 11: How to load experimental data.

Table 4: Accuracies of each motion prediction (one motion).

	Men	Kote	Dou	Pose
1	0.943	0.857	0.971	0.984
2	0.936	0.961	0.878	1.000
3	0.989	0.905	1.000	0.865
4	1.000	0.915	1.000	1.000
5	1.000	1.000	1.000	1.000
6	0.987	0.952	0.953	0.913
7	0.907	1.000	0.917	1.000
8	0.933	0.786	0.983	0.994
9	0.943	0.675	1.000	1.000
10	0.990	0.893	1.000	0.949
Ave	0.963	0.894	0.970	0.970

Dou, Men, and Pose all exceeded 95%, indicating that these motions are accurately predicted. Although Kote's accuracy was slightly lower than that of the other movements due to some mispredictions with Men, which have similar movements, Kote was able to correctly predict 89% of the movements, which can be considered a success in general.

4.2 Two Motions Experiment

The target motion was defined as the period from the state of Pose to Men, Kote, or Dou and the end of those motions. In this experiment, not only the accuracy of the motions but also the accuracy of the prediction time sequence is important because the

target motion includes the moment when the subject performs the technique from the state of Pose.

As experimental data, we prepared 15 videos for each motion and evaluated them the same way as in the previous experiment. Figures 12 to 14 show an example of the predicted transition for each motion. Table 5 shows the accuracies for each motion. Although the accuracies decreased compared to the previous experiment, the timing of the switch between Pose and Strike motions was predicted appropriately in many cases in all motions and can be considered a success. One of the reasons for the decrease in the accuracies was the misprediction during the striking motion, which was observed mainly in Men. The causes are discussed in the next section.

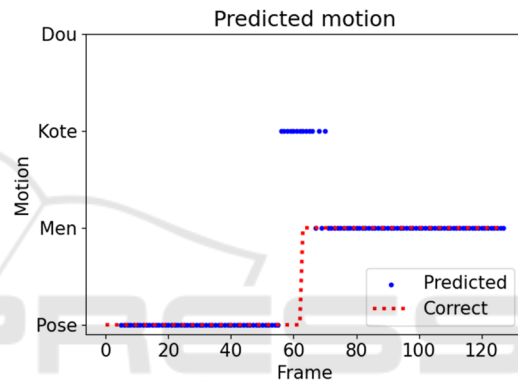


Figure 12: Transition of prediction (Pose to Men).

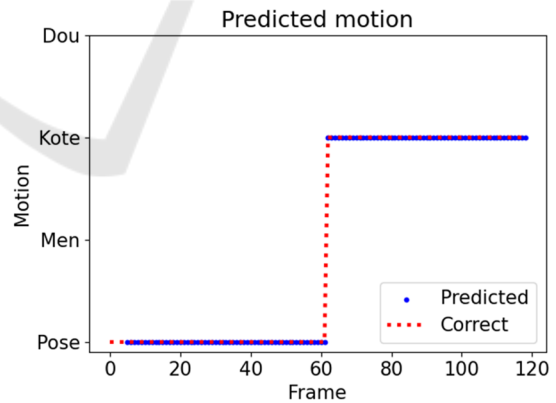


Figure 13: Transition of prediction (Pose to Kote).

4.3 Discussion

The causes of the mispredictions listed in the previous sub-section are discussed. Figure 15 shows how mispredictions occur in the Men and Kote motions.

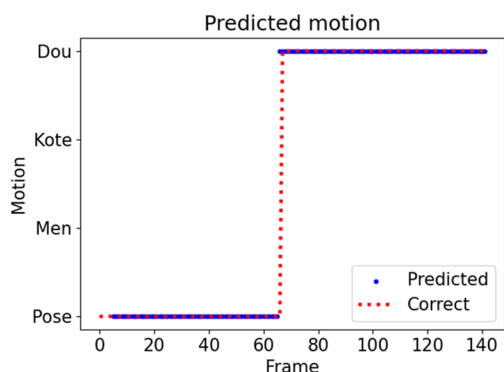


Figure 14: Transition of prediction (Pose to Dou).

Table 5: Accuracies of each motion prediction (two motions).

	Pose to Men	Pose to Kote	Pose to Dou
1	0.894	0.826	1.000
2	0.838	0.829	0.973
3	0.748	0.812	0.963
4	0.796	0.897	0.965
5	0.791	0.876	0.993
6	0.791	0.983	0.971
7	0.848	1.000	0.986
8	0.755	0.828	0.944
9	0.826	0.658	0.957
10	0.714	0.780	0.844
11	0.622	0.913	0.930
12	0.765	0.811	0.937
13	0.706	0.805	0.933
14	0.762	0.836	0.963
15	0.826	0.845	0.978
Ave	0.779	0.847	0.956

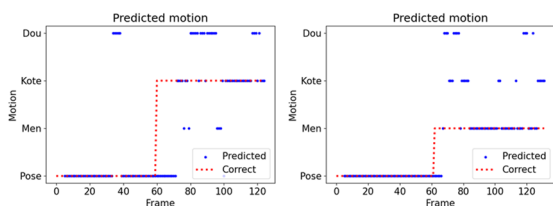


Figure 15: Transition of prediction (Pose to Kote or Men).

In the Pose phase, all of the predictions are made without any problem, but it can be seen that false predictions occur after transitioning to the striking motion. In this example, the prediction of the timing of the motion changeover is also far from the correct answer. One of these factors may be the similarity between Men and Kote motion. Figure 7 shows that Men and Kote swing bamboo swords vertically, while Dou swings bamboo swords slightly horizontally, so Men and Kote might be similar regarding player body

motion. Dou, who uses a different shinai swinging motion, also achieved a very high accuracy (96%) in the two-motions test, suggesting that the difference in sword swinging motion significantly affects the results.

However, the evaluation experiments of the single kendo motion shown in Table 4 all produced a high accuracy. Thus Figure 14 should also be stable during the striking motions. In this case, the possibility of overlearning due to insufficient training data cannot be denied. Figure 10 shows that the difference between Train and Validation losses widens slightly after epoch 100. Although not to the extent of fatal overlearning, overlearning should be prevented for versatility.

Therefore, the improvements are to reflect the differences between Men and Kote motion as numerical values in the learning model by adding joint position information that represents other joint positions using the shoulder angle and the height of the right foot as the origin as feature values, and to increase the number of learning data and take measures against overlearning in the learning phase.

5 CONCLUSION

In this paper, we developed a motion prediction system for kendo motions with the aim to design a VR Kendo system that allows users to easily training Kendo at home and other places.

The proposed system is consisted with OpenPose to obtain joint position information and machine learning using RNN (LSTM) to learn and predict kendo motions.

As a result, four kendo motions were predicted with an accuracy rate of over 95%, while some incorrect predictions were observed for the Men and Kote motions. We consider that this is due to the similarity between Kote and Men motions and slight overlearning. Therefore, additional features and learning data are needed to solve this problem.

As future works, we improve the machine learning and develop a VR Kendo system using a small controller for VR and the machine learning model.

REFERENCES

All Japan Kendo Federation. (n.d.). *What Is Kendo?* In All Japan Kendo Federation Web Site. <https://www.kendo.or.jp/en/knowledge/>, Last accessed on July 17, 2023.

- Y. Tanaka and K. Kosuge, (2014) Dynamic attack motion prediction for kendo agent, *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2187-2193.
- M. Takata, Y. Nakamura, Y. Torigoe, M. Fujimoto, Y. Arakawa, and K. Yasumoto. (2019). Strikes–thrusts activity recognition using wrist sensor towards pervasive kendo support system. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 243–248.
- Cao, Yongpeng, and Yuji Yamakawa. (2022). Marker–less Kendo Motion Prediction Using High–speed Dual–camera System and LSTM Method. *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 159–164.
- Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *The Journal of Machine Learning Research*, 13(1), 281–305.
- Cao, Z., Simon, T., Wei, S., & Sheikh, Y. (2017). Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7291–7299.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/https://doi.org/10.1162/neco.1997.9.8.1735>
- Turvey, M. T., Park, H., Dumais, S. M., & Carello, C. (1998). Nonvisible perception of segments of a hand-held object and the attitude spinor. *Journal of Motor Behavior*, 30(1), 3–19.
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2, 3104–3112.