International Journal of Innovative
Computing, Information and Control
Volume **18**, Number **5**, October **2022**

ICIC International ©2022 ISSN 1349-4198
pp. **1657–1666**

HAND MOTION PREDICTION USING NEURAL NETWORK (ROCK-PAPER-SCISSORS)

Kippei Yanagi[1], Kotaro Hashikura[2], Md Abdus Samad Kamal[2] and Kou Yamada[2]

[1]Department of Mechanical Science and Technology
School of Science and Technology
[2]Division of Mechanical Science and Technology
Gunma University
1-5-1 Tenjincho, Kiryu 376-8515, Japan
{ t180b098; k-hashikura; maskamal; yamada }@gunma-u.ac.jp

Received February 2022; revised June 2022

Abstract. *In this paper, we propose a long short-term memory (LSTM)-based neural network to predict the hand motion one frame 33 [ms] before the hand is shown by observing a rock-paper-scissors game. The input to the network is the 2D pose estimation information of the hand and the displacement of each joint point obtained by Mediapipe Hands, and the 2D data of each joint point estimated one frame ahead 33 [ms] of a 30 [fps] (frames per second) camera is used for comparison. It is confirmed that the proposed method can predict the movements in rock-paper-scissors appropriately using a neural network.*

Keywords: Human-computer interaction, Machine learning, Latency, Hand motion prediction

1. **Introduction.** Much research has been done in the field of human-computer interaction (HCI), in which a computer reads and acts upon the intentions and emotions of human actions [1, 2, 3, 4, 5, 6, 7]. Among them, hand gestures have been adopted in many systems as a means of conveying intentions and emotions from humans to computers [3, 4, 5, 6, 7]. In previous research on gesture recognition, Sharma and Verma demonstrated fast static gesture recognition from a single RGB image using skin color-based image analysis methods and other methods [3]. Aashni et al. showed that robust gesture recognition is possible for both static and dynamic gestures using cascade classifier and other methods without using hand markers [4].

Methods for the recognition of hand gestures are divided into two types. One is a glove-based method [8]. In the glove-based method, various sensors attached to the glove worn by the user are used to measure acceleration, joint angles, etc., to recognize the user's hand gesture [8]. The other is a vision-based system [9]. In the vision-based method, hand gestures are recognized based on images obtained from a camera using image analysis without wearing a glove or any other device [9]. An interesting study has been published on the application of deep learning-based object detection to a hand bone age assessment system [10], the results of which are outside the scope of this study.

In hand gesture recognition, latency is one of the most important system factors. Latency is caused by various factors such as delay in sensor response and delay in data communication. In the interaction between humans and computers by hand gestures, the

DOI: 10.24507/ijicic.18.05.1657

effect of latency is significant. Depending on the degree of latency, the user's operability and usability are greatly impaired.

In order to reduce the latency in hand gesture recognition, Ito et al. have developed a vision-based active sensing system for hand gesture recognition with high response and low latency by combining high-speed sensing and high-speed vision cameras [11]. They demonstrated the system in a rock-paper-scissors game and showed that the system can respond in less time than humans can perceive [11]. However, to realize these systems, much expensive equipment such as high-speed sensing and high-speed vision cameras is required. Furthermore, their method cannot predict future hand motion.

There is a possibility to predict future hand motion using machine learning. The prediction of short-term motions using neural networks has attracted much attention. In particular, there have been many studies on predicting human motion [12, 13, 14, 15, 16]. Martinez et al. analyzed human pose estimation using recurrent neural networks and proposed a new method for short-term motion prediction of the human body [12]. Chao et al. integrated human pose estimation and sequence prediction to transform a single image into 3D space and predict its dynamics [13]. Erwin and Hideki proposed a new mixed reality type martial arts training system by predicting the dynamics of the human body and projecting them onto VR goggles in real time [14]. As for the prediction of short-term movements in hand gestures, Kanokoda et al. predicted several finger movements using time delay neural network (TDNN) based on information obtained from a wearable device using Pyrolytic Graphite Sheets (PGS) [16]. However, no paper examines predicting short-term hand gesture movements using neural networks in a vision-based method.

In this paper, we overcome the problem by the method by [11] and the problem of predicting human motion in [12, 13, 14, 15, 16] and propose a method for a highly responsive and low-latency hand gesture recognition system, as a different approach from the aforementioned research to reduce latency in hand gesture recognition. We adopt a neural network to predict hand gesture movements based on information obtained from a general camera at 30 [fps] as input to the system.

This paper is organized as follows. In Section 2, we propose a system for highly responsive and low-latency hand gesture recognition. In Section 3, we describe the dataset and the model training method, followed by the model training results, real-time demonstration results, and discussion. Section 4 gives some concluding remarks.

2. System for a Highly Responsive and Low-Latency Hand Gesture Recognition.

Consider a system for a highly responsive and low-latency hand gesture recognition in Figure 1. The system in Figure 1 consists of three parts: 2D pose estimation, preprocessing, and 2D pose forecasting. For 2D pose estimation, we adopt Mediapipe Hands [18], a library specialized for hand pose estimation from MediaPipe [17] provided by Google limited liability company (LLC), a pose estimation framework provided by Google LLC. The pose estimation coordinates output by Mediapipe Hands are 21 joint and fingertip coordinates normalized to the range of 0 to 1 depending on the height and width of the frame image. The next step is preprocessing. As a preprocessing method for short-term motion prediction using neural networks, Erwin and Hideki used the pose estimation information obtained from a single image and the motion estimation information obtained using the lattice optical flow method, which is a lightweight version of optical flow, as inputs to the network [14]. We have improved the prediction accuracy of the neural network model. Referring to the aforementioned method, as a more lightweight preprocessing, the pose estimation coordinates of two frames in two dimensions and the amount of displacement between frames are used as the motion estimation information as inputs to the network. Therefore, the input dimension of the neural network in the 2D forecasting described

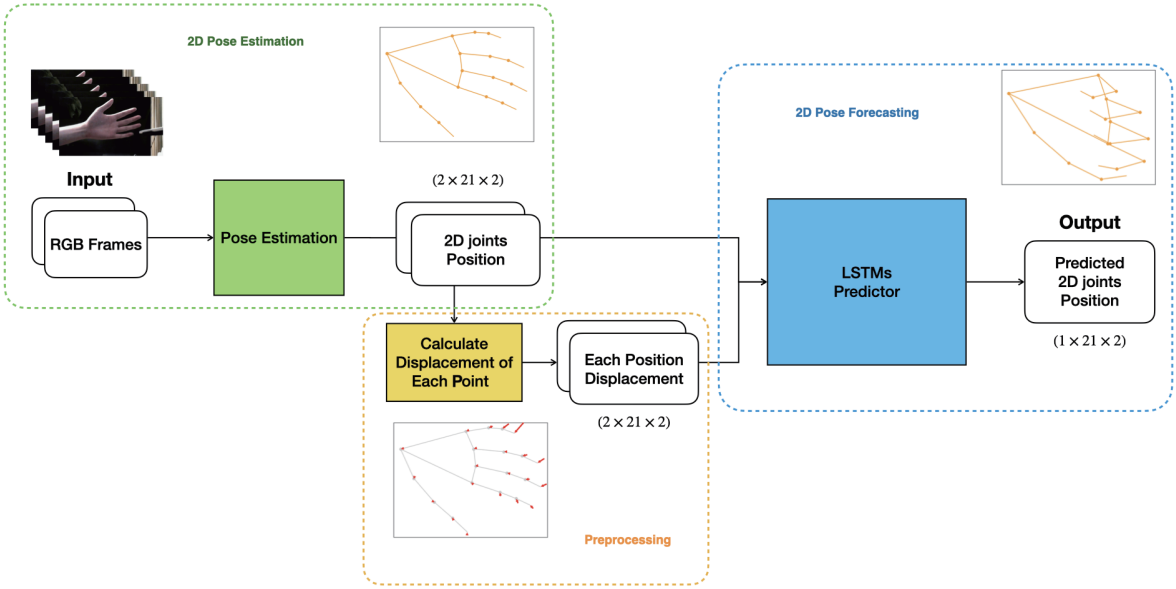


FIGURE 1. The overview of 2D hand pose forecasting system, consisting of three parts: 2D pose estimation, preprocessing, and 2D pose forecasting

below is (2×84), which is the sum of the 21 joint point information (2×42) and the amount of each displacement obtained from the pose estimation for 2 frames. For the 2D pose forecasting, we use the neural network to predict hand motion. The neural network model used in the system in Figure 1 is summarized in Figure 2. The network in Figure 2 consists of four layers including a long short-term memory (LSTM) [19, 20] layer to learn temporal features in hand movements and a fully connected layer. The input layer is an 84 units LSTM layer with a hyperbolic tangent as the activation function. The input is then branched for each frame and sent to the next LSTM layer of 252 units. Each of these LSTM layers has a rectified linear unit (ReLU) [21] as the activation function. The outputs of the branched LSTM layers are then added together and sent to the 42-unit fully connected output layer. The output layer has ReLU as the activation function. During training, in order to prevent overfitting, the drop out layer [22] is applied. After tuning, the inactivation rate p in the drop out layer, which is a hyperparameter, is set to 50% $(p = 0.5)$. Using Adam proposed in [23], the loss function L is trained as

$$L = \lambda_{\text{mse}} L_{\text{mse}} - \lambda_{\cos} L_{\cos}, \tag{1}$$

where L_{mse} is mean squared error (MSE) and written by

$$L_{\text{mse}} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{2}$$

λ_{mse} and $\lambda_{\cos}$ are constants, y is the ground-truth joint position, $\hat{y}$ is the predicted joint position, N is the output dimension and $L_{\cos}$ is cosine similarity written by

$$L_{\cos} = \frac{y \cdot \hat{y}}{\|y\|_2 \|\hat{y}\|_2}. \tag{3}$$

Note 2.1. *The $L_{\cos}$ cosine similarity measures the similarity of two vectors in the inner product space, where the two vectors are the prediction and the ground-truth coordinates. Cosine similarity ranges between 1 and -1, and the higher the similarity, the closer to 1.*

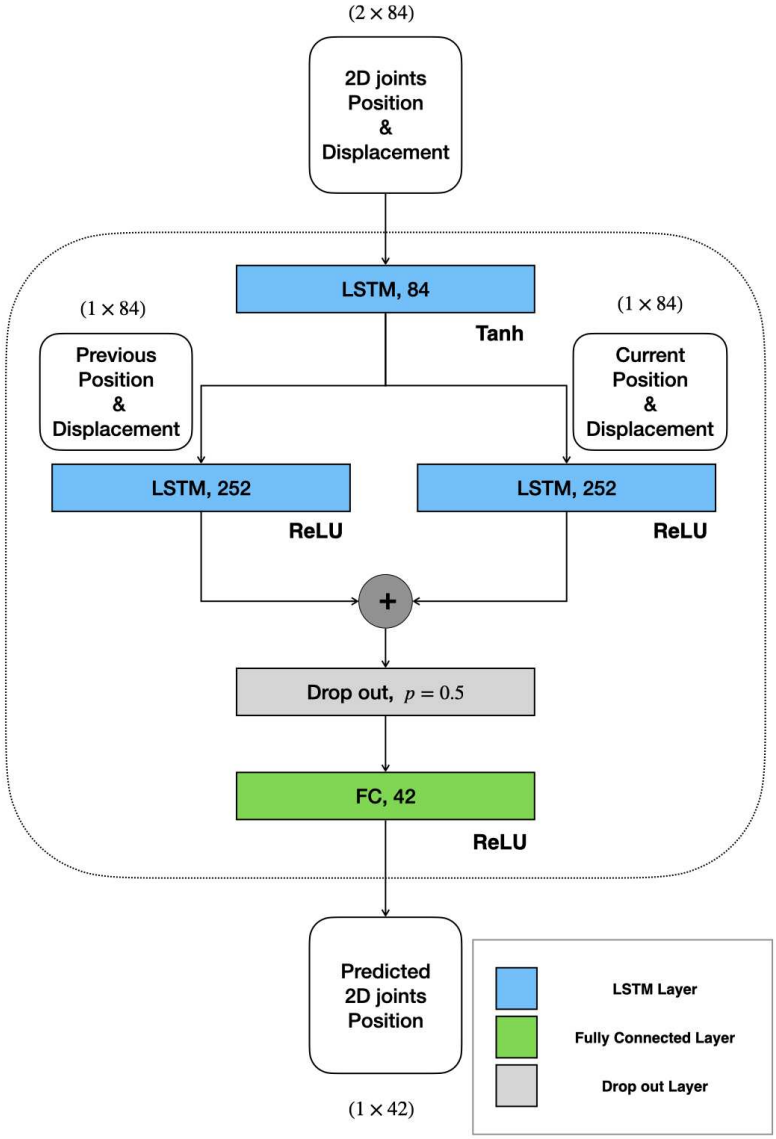


FIGURE 2. The structure of the 2D hand pose prediction network

Cosine similarity is used as a similarity metric between faces in face verification [24] *and between human poses in pose estimation* [25]*, etc.*

The neural network is trained from information including 2D joint and fingertip coordinates and their displacements (2×84) for the last two frames including the current frame, and 21 joint and fingertip x, y-coordinates (1×42) for each of the 21 joints and fingertips one frame (33 [ms]) later as ground-truth data.

3. **Results and Discussions.** In this section, we explain the training results, the results of the real-time demonstration experiment, and the discussion.

For the dataset, the distance between the hand and the camera is fixed at about 50 [cm], and the hand is oriented such that the palm always faces the camera. We apply the hand pose estimation by Mediapipe Hands to the captured images and used 7200 frames as a dataset. We train the predictor using 5040 frames (70% of the 7200 frames) as training data and 2160 frames (the remaining 30%) as validation data. Mini-batch learning is adopted for training, with a batch size of 32 and 150 epochs. The output

dimension N is $N = 42$. λ_{mse} in (1) and $\lambda_{\cos}$ in (1) are settled by

$$\lambda_{\text{mse}} = 0.80 \tag{4}$$

and

$$\lambda_{\cos} = 0.75, \tag{5}$$

respectively.

All model training and validation are performed on a laptop PC with Intel(R) Core i5-7360U 2.30GHz CPU, Intel Iris Plus Graphics 640 graphics, and 16GB memory. The Tensorflow ver 2.7.0 library [26] is used to build and train the network model. An RGB camera, Logitech C270 720p, 30 [fps], is used to create the dataset and to capture the real-time experiments. The root mean squared error (RMSE) is used to evaluate the prediction accuracy [27]. RMSE and its average value at each joint point for the prediction results of the training and validation data are summarized in Table 1. Each value is calculated from the pixel coordinates of each joint obtained by pose estimation for each frame and the measured values at each joint point. The average error between the training data and the validation data is about 2 [mm], and the prediction error tends to be larger the closer to the fingertip, but all the values are small.

Next, we show the relationship between the ground-truth coordinate data of each fingertip and the predicted coordinate data at one frame interval (33 [ms]) in the validation data. The relationship is shown in Figure 3. Figure 3 shows that the prediction model can

Table 1. Root mean squared error (RMSE) in training and validation data for x- and y-coordinates at each of 21 joint points [mm]

	Train		Test	
Joints/tips	x [mm]	y [mm]	x [mm]	y [mm]
Wrist	9.1	5.1	8.1	7.2
Thumb/Carpometacarpal joint	8.8	5.2	8.8	13.2
Thumb/Metatarsophalangeal joint	9.2	5.8	13.6	16.0
Thumb/Interphalangeal joint	9.9	6.9	17.6	12.4
Thumb/Tip	9.8	7.5	13.2	10.1
Index/Metatarsophalangeal joint	8.3	4.2	14.7	4.6
Index/Proximal interphalangeal joint	8.3	5.7	12.1	6.7
Index/Distal interphalageal joint	9.1	5.7	9.5	9.0
Index/Tip	11.6	6.0	12.2	9.8
Middle/Metatarsophalangeal joint	8.4	3.0	14.4	6.1
Middle/Proximal interpharangeal joint	8.2	5.1	10.9	6.2
Middle/Distal interphalageal joint	9.5	4.1	9.9	6.0
Middle/Tip	13.6	6.1	13.6	7.2
Ring/Metatarsophalangeal joint	8.1	4.4	11.6	7.0
Ring/Proximal interpharangeal joint	8.1	4.7	8.8	5.8
Ring/Distal interphalageal joint	10.9	3.8	11.0	4.9
Ring/Tip	13.2	4.7	15.0	4.9
Pinky/Metatarsophalangeal joint	9.1	6.0	9.6	8.2
Pinky/PIP	9.2	6.0	8.3	6.0
Pinky/Distal interphalageal joint	10.2	7.1	10.2	5.0
Pinky/Tip	13.0	7.8	13.1	7.2
Average	9.8	5.5	11.7	7.8

capture the movement tendency of the fingers and that there is no obvious delay between the predicted data and the ground-truth data.

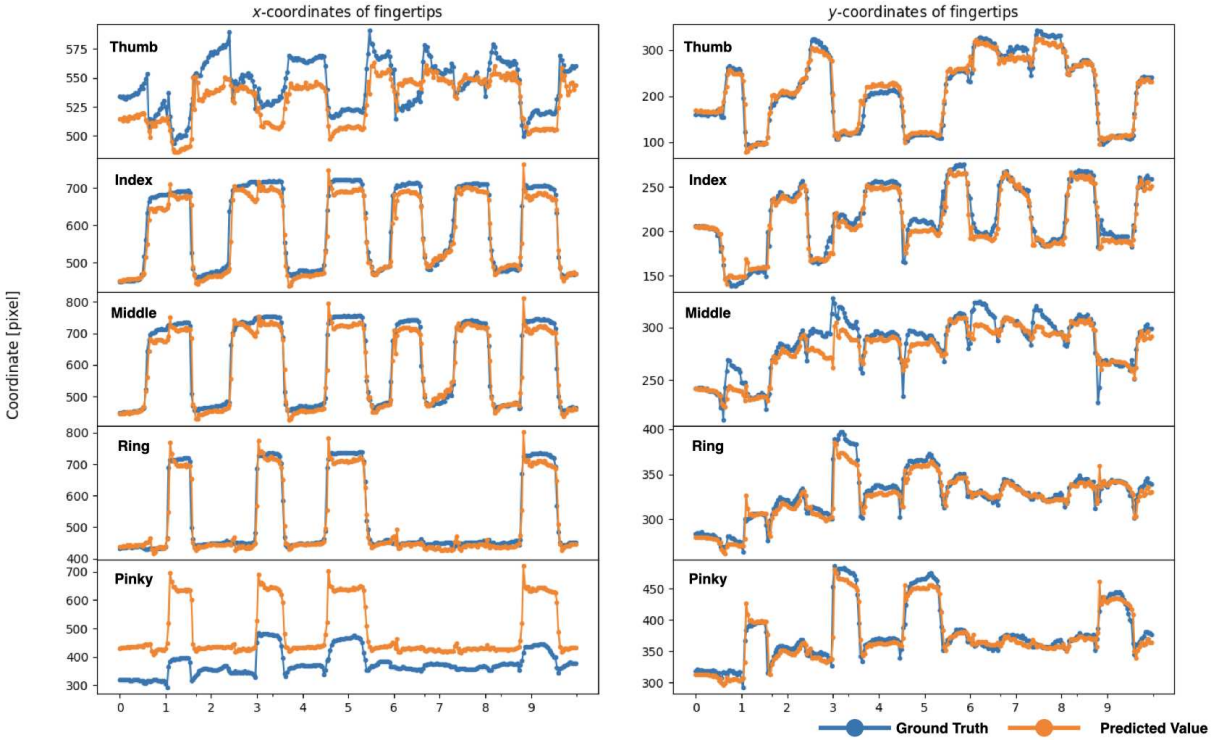


FIGURE 3. Relationship between ground-truth values and predicted values for the x- and y-coordinates of each fingertip

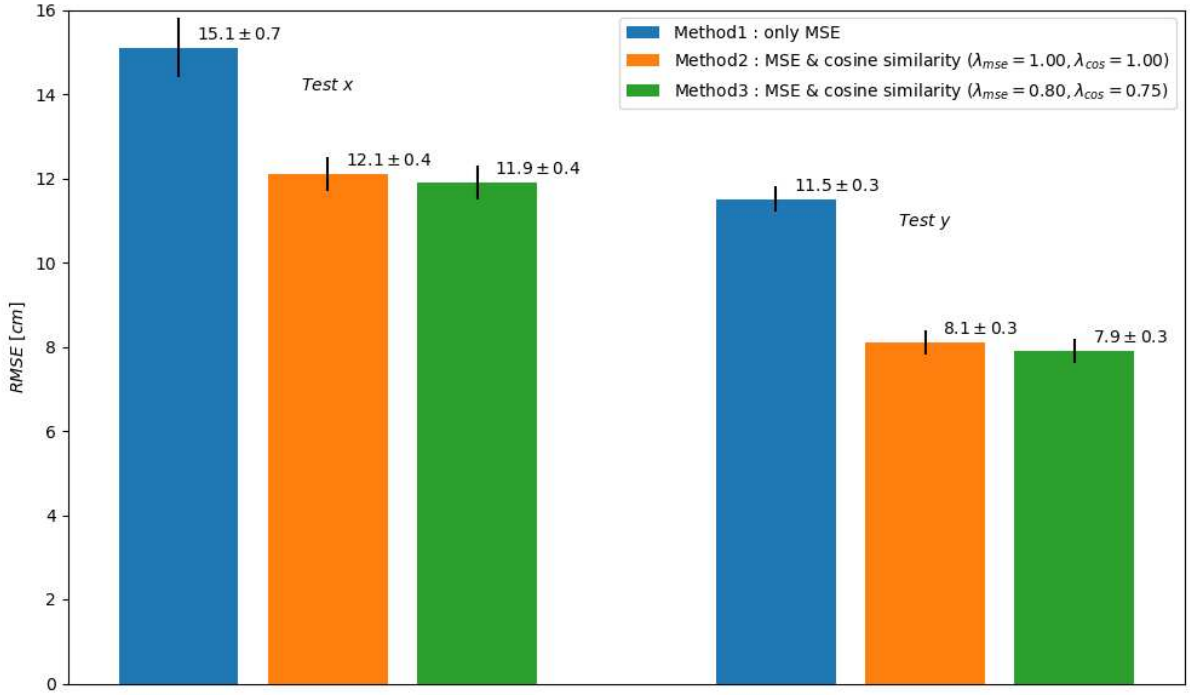


FIGURE 4. Prediction errors (RMSE) in the x- and y-axis directions for test data from methods 1 to 3 and their comparison

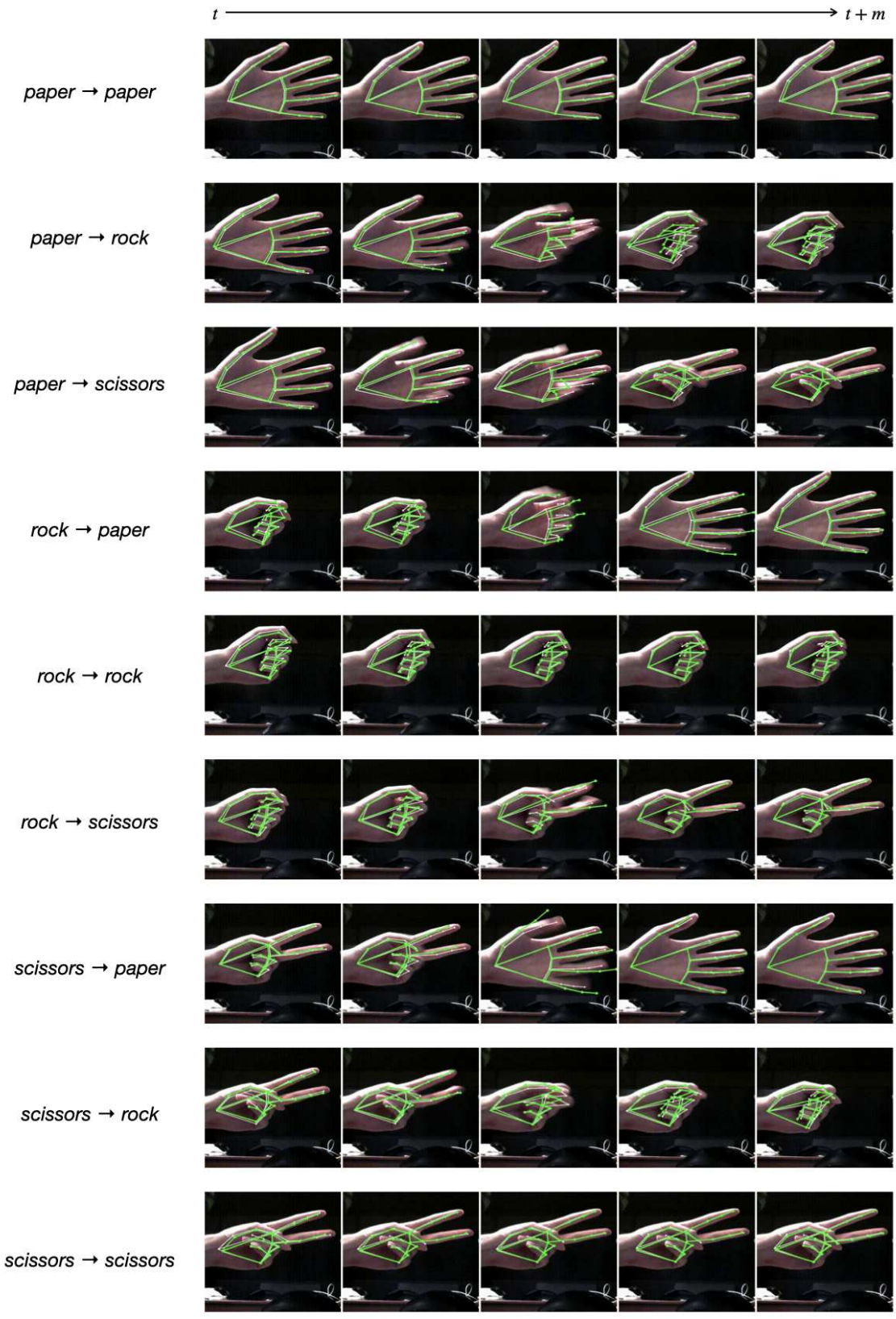


FIGURE 5. (color online) Real-time prediction results for all behavioral patterns of rock-paper-scissors. The actions of each pattern are plotted in chronological order, five frames at a time, from left to right.

In order to confirm the validity of the loss function and hyperparameters λ_{mse} and $\lambda_{\cos}$ newly introduced in this study, the model training was repeated 50 times for each of the following three methods for comparison. In method 1, only MSE was simply applied to the loss function. In method 2, the loss function was trained by combining MSE and cosine similarity with $\lambda_{\text{mse}} = 1.00$ and $\lambda_{\cos} = 1.00$. In method 3, the loss function combining MSE and cosine similarity was trained with $\lambda_{\text{mse}} = 0.80$ and $\lambda_{\cos} = 0.75$. Figure 4 plots the average prediction accuracy (RMSE) in the x- and y-axis directions for the 21 2D joint point positions predicted one frame (33 [ms]) later for the validation data trained using methods 1-3. Figure 4 shows that the prediction accuracy is improved by about 20% by using a loss function that takes the cosine similarity into account, rather than using MSE alone. It is also confirmed that the prediction accuracy is improved by appropriately setting the newly introduced hyperparameters λ_{mse} and $\lambda_{\cos}$.

In order to verify the prediction capability of the model constructed in this paper, we demonstrate the hand motion prediction of the rock-paper-scissors motion by this model in real time. The prediction results for each pattern of rock-paper-scissors for real-time pose estimation and motion prediction using the neural network are shown in Figure 5. Here, the white line shows the pose estimation result of the frame, and the green line shows the motion after one frame (33 [ms]) predicted by the neural network. From Figure 5, it is confirmed that the neural network model predicts the behavior one frame (33 [ms]) ahead of the sign of the behavior in rock-paper-scissors even in the real-time prediction.

These results show that the LSTM-based model constructed in this study is useful for predicting even high-speed movements such as rock-paper-scissors without any lag.

4. **Conclusions.** In this paper, we have proposed a method for a highly responsive and low-latency hand gesture recognition system. The proposed method is to develop a neural network-based system for predicting the movements of a rock-paper-scissors game on the screen plane 33 [ms] in advance. As a result of training the system on our dataset, we obtained RMSE of 10 [mm] in the x-axis direction 6 [mm] in the y-axis direction for the training data, 12 [mm] in the x-axis direction, and 8 [mm] in the y-axis direction for the validation data. In addition, we filmed a rock-paper-scissors game and confirmed that the prediction was possible in real time. In this way, it is clarified that the proposed method is an effective method of a highly responsive and low-latency hand gesture recognition system by experimental results. The proposed method can provide higher responsiveness by giving the predicted input to the system, and it is expected that the proposed method will be applied to human motion support and cooperative motion with smaller latency in the future.

REFERENCES

[1] G. Vennila and M. S. K. Manikadan, Detection of human and computer voice spammers using hidden Markov model in voice over Internet protocol network, *Procedia Computer Science*, vol.115, pp.588-595, 2017.

[2] E. Caceres, M. Carrasco and S. Rios, Evaluation of an eye-pointer interaction device for human-computer interaction, *Heliyon*, vol.4, no.4, 2018.

[3] R. Sharma and G. Verma, Human computer interaction using hand gesture, *Procedia Computer Science*, vol.54, pp.721-727, 2015.

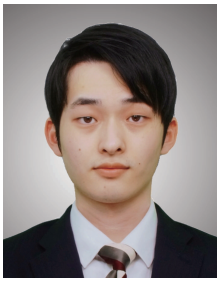
[4] H. Aashni, S. Archanasri, A. Nivedhitha, P. Shristi and N. Jyothi, Hand gesture recognition for human computer interaction, *Procedia Computer Science*, vol.115, pp.367-374, 2017.

[5] X. Li, Human-robot interaction based on gesture and movement recognition, *Signal Processing: Image Communication*, vol.81, 2020.

[6] M. Deng, Robust human gesture recognition by leveraging multi-scale feature fusion, *Signal Processing: Image Communication*, vol.83, 2020.

[7] R. Khan and N. Ibraheem, Hand gesture recognition: A literature review, *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol.3, pp.161-174, 2012.
[8] Y. N. Khan and S. A. Mehdi, Sign language recognition using sensor gloves, *International Conference on Neural Information Proceedings*, vol.5, pp.2204-2206, 2002.
[9] P. Narayana, J. R. Beveridge and B. A. Draper, Gesture recognition: Focus on the hands, *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.5235-5244, 2018.
[10] C. Lee and B.-D. Lee, Enhancement for automatic extraction of RoIs for bone age assessment based on deep neural networks, *ICIC Express Letters*, vol.14, no.2, pp.163-170, 2020.
[11] K. Ito, T. Sueishi, Y. Yamakawa and M. Ishikawa, Tracking and recognition of a human hand in dynamic motion for Janken (rock-paper-scissors) robot, *IEEE International Conference on Automation Science and Engineering (CASE)*, pp.891-896, 2016.
[12] J. Martinez, M. Black and J. Romero, On human motion prediction using recurrent neural networks, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4674-4683, 2017.
[13] Y. Chao, J. Yang, B. Price, S. Cohen and J. Deng, Forecasting human dynamics from static images, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.3643-3651, 2017.
[14] W. Erwin and K. Hideki, FuturePose – Mixed reality martial arts training using real-time 3D human pose forecasting with a RGB camera, *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp.1384-1392, 2019.
[15] S. Toyer, A. Cherian, T. Han and S. Gould, Human pose forecasting via deep Markov models, *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp.1-8, 2017.
[16] T. Kanokoda, Y. Kushitani, M. Shimada and J. Shirakashi, Motion prediction with artificial neural networks using wearable strain sensors based on flexible thin graphite films, *Key Engineering Materials*, vol.826, pp.111-116, 2019.
[17] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. Chang, M. Yong, J. Lee, W. Chang, W. Hua, M. Georg and M. Grundmann, MediaPipe: A framework for perceiving and processing reality, *The 3rd Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019.
[18] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. Chang and M. Grundmann, MediaPipe hands: On-device real-time hand tracking, *arXiv.org*, arXiv: 2006.10214, 2020.
[19] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation*, vol.9, no.8, pp.1735-1780, 1997.
[20] F. Gers and J. Schmidhuber, Learning to forget: Continual prediction with LSTM, *Neural Computation*, vol.12, no.10, pp.2451-2471, 2000.
[21] A. Agarap, Deep learning using rectified linear units (ReLU), *arXiv.org*, arXiv: 1803.08375, 2019.
[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, vol.15, no.1, pp.1929-1958, 2014.
[23] D. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv.org*, arXiv: 1412.6980, 2017.
[24] V. H. Nguyen and L. Bai, Cosine similarity metric learning for face verification, *The 10th Asian Conference on Computer Vision (ACCV)*, vol.6493, pp.709-720, 2010.
[25] P. Borkar, M. Pulinthitha and A. Pansare, Match pose – A system for comparing poses, *International Journal of Engineering Research & Technology (IJERT)*, vol.8, no.10, 2019.
[26] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng, TensorFlow: A system for large-scale machine learning, *Operating Systems Design and Implementation (OSDI)*, pp.265-283, 2016.
[27] T. Chai and R. Draxler, Root mean square error (RMSE) or mean absolute error (MAE), *Geoscientific Model Development*, vol.7, no.3, pp.1247-1250, 2014.

Author Biography



Kippei Yanagi received the B.S. degree of Mechanical Science and Technology from Gunma University in 2022. His research interests include artificial intelligence, control theory and their application.

Kotaro Hashikura received the B.S. degree of Mechanical Engineering, the M.S. degree of Informatics, and the Dr. degree of Engineering from Kyushu Institute of Technology, Fukuoka, Japan in 2006, from Kyoto University, Kyoto, Japan in 2010, and from Tokyo Metropolitan University, Tokyo, Japan in 2014, respectively. From 2014 until 2018, he had been a Project Research Associate at the Faculty of System Design, Tokyo Metropolitan University. He is currently an Assistant Professor at the Department of Mechanical Science and Technology, Gunma University, Japan. His research interests include time-delay-related control techniques, such as deadbeat, preview-prediction and repetitive controls. He is a member of IEEE, ISCIE and SICE.

Md Abdus Samad Kamal received the B.Sc. degree in Electrical and Electronic Engineering from Khulna University of Engineering and Technology (KUET), Khulna, Bangladesh in 1997, Master and Doctor degrees from Kyushu University from Graduate School of Information Science and Electrical Engineering, Japan in 2003 and 2006, respectively. He was a post-doctoral fellow in Kyushu University till November 2006. He is currently an associate professor at the Department of Mechanical Science and Technology, Gunma University, Japan. His current research interests are reinforcement learning, intelligent transportation systems, and multiagent systems. He is a member of IEEE and SICE.

Kou Yamada received B.S. and M.S. degrees from Yamagata University, Yamagata, Japan in 1987 and 1989, respectively, and a Dr. Eng. degree from Osaka University, Osaka, Japan in 1997. From 1991 to 2000, he was with the Department of Electrical and Information Engineering, Yamagata University, Yamagata, Japan as a research associate. From 2000 to 2008, he was an associate professor in the Department of Mechanical System Engineering, Gunma University, Gunma, Japan. Since 2008, he has been a professor in the Department of Mechanical System Engineering, Gunma University, Gunma, Japan. His research interests include robust control, repetitive control, process control, and control theory for inverse systems and infinite-dimensional systems. Prof. Yamada received the 2005 Yokoyama Award in Science and Technology, the 2005 Electrical Engineering/Electronics, Computer, Telecommunication, and Information Technology International Conference (ECTI-CON2005) Best Paper Award, the Japanese Ergonomics Society Encouragement Award for an Academic Paper in 2007, the 2008 Electrical Engineering/Electronics, Computer, Telecommunication, Information Technology International Conference (ECTI-CON2008) Best Paper Award, and the 4th International Conference on Innovative Computing, Information and Control Best Paper Award in 2009, the 14th International Conference on Innovative Computing, Information and Control Best Paper Award in 2019, and Outstanding Achievement Award from Kanto Branch of Japanese Society for Engineering Education in 2022. He is a member of IEEE and SICE.