acrt AI, Computer Science and Robotics Technology

IntechOpen Journals

RESEARCH PAPER

Enhanced Detection of Musical Performance Timings Using MediaPipe and Multilayer Perceptron Classifier

Kazuteru Tobita* and Kazuhiro Mima

Shizuoka Institute of Science and Technology, Shizuoka, Japan
*Corresponding author. E-mail: tobita.kazuteru@sist.ac.jp

Citation
Kazuteru Tobita and Kazuhiro Mima (2024), Enhanced Detection of Musical Performance Timings Using MediaPipe and Multilayer Perceptron Classifier. *AI, Computer Science and Robotics Technology* 3(1), 1–15.

DOI
https://doi.org/10.5772/acrt.20240002

Received: 25 January 2024
Accepted: 21 August 2024
Published: 30 August 2024

Abstract

This research aims to enhance the collaborative work system between humans and robots by exploring "ensemble music." In an ensemble, it is crucial to adhere to the score, synchronize it with the breathing of fellow musicians, and ensure harmonious performance. This represents one of the most intricate collaborative endeavors achievable by humans. In this study, by examining various image processing methods for detecting the movement of a performer, it was shown that skeleton detection using MediaPipe is appropriate in terms of a large amount of information and processing speed. Next, a deep neural network utilizing the history of MediaPipe's 3D skeletal coordinates as the input was developed to detect the performance start and end points. A comprehensive examination of learning and estimation conditions via grid search revealed that the start and end points could be estimated with approximately 70% and 100% accuracy, respectively, when using a history of 10 points, the ReLU activation function, and the L-BFGS optimizer. Additionally, the estimation time was 10 ms or less when the hidden layer had 100 or fewer units. Future detection accuracy will be enhanced by incorporating additional learning data and assigning greater weights to skeleton points with significant changes.

Keywords: ensemble, cooperative work, neural network, skeleton detection, pure chord

1. Introduction

In recent years, there has been progress in the research and development of robots that collaborate with humans, against a background of labor shortages in various industries. Currently, cooperative robots are mainly utilized for machine tending purposes, which reduces the need for safety fences [1], based on various safety standards [2–4]. In the coming years, the need for more sophisticated collaborative work systems is predicted to rise. This is aimed at boosting productivity in different sectors, given the dwindling working-age population.

The study aims to explore the concept of "ensemble" to enhance collaborative work systems. In this context, "ensemble" refers to not only adhering to a predetermined score but also attuning to each other's breath, which is a pinnacle of sophisticated collaboration among humans. Therefore, by developing a collaborative system that combines human and machine capabilities, we expect to see wider applications across multiple industries and the emergence of more sophisticated collaborative work systems.

Detecting the performer's movements in this way is essential for improvisation support for music beginners [5, 6] and for performance movement learning support [7]. Alternatively, in a device that controls volume and other functions with hand gestures while playing [8], it is possible that control can be performed only with the movements necessary for playing.

Although there have been examples of research on systems in which robots play together based on synchronous signals, these have not been designed for performance with people [9, 10]. There is research on real-time musical synchronization between a human musician [11] and an accompaniment system that detects cues from breath [12], and also research on a visionary collaboration that transcends time by reproducing the performances of past masters on an automatic piano [13, 14]. However, there have been no efforts made to achieve dual coordination, incorporating not only the temporal aspect but also the frequency dimension. This involves synchronizing the moments when a human and a performance device synchronize, seamlessly playing a harmonious chord without any fluctuations [15]. So far, we have envisioned a "just intonation concert system" that plays in concert with a human player, and prototyped and established the "player motion detection system" and "real-time volume pitch control system," which are the two main subsystems. Several movement experiments were conducted [16].

This paper describes the comparative results of the performance of various tracking methods used in the player motion detection system, and experiments on the detection of the start and end timing of a performance using MediaPipe [17] and

a neural network. The present paper includes material previously reported in oral presentations [18, 19], but new experimental results also have been added.

2. Tracking method considerations

2.1. Various tracking methods

2.1.1. Automatic recognition of human frontal faces using cascade classifiers

OpenCV's cascade classifier automatically and consistently identifies particular color patterns in video frames to recognize human frontal faces and eyes. Obtaining the positional coordinates of the identified frontal faces and eyes on the pixels opens up the possibility of using the change history for motion discrimination.

2.1.2. Tracking of arbitrary object positions by trackers

The OpenCV tracker is a software tool that enables users to define a region of interest (ROI) in a video frame and subsequently track areas with similar characteristics in subsequent videos. Its positional coordinates can be used for motion discrimination, similar to the cascade classifier. The study employed KCF and MIL as tracking algorithms.

2.1.3. Skeletal detection by MediaPipe

MediaPipe is a video analysis library for the detection of various objects. MediaPipe Pose, a human skeleton detection model, can predict the coordinates of a total of 33 points on a human body in a video frame, including the nose, both eyes, both pupils, both eye corners, both ears, both shoulders, both elbows, both wrists, both little fingers, both index fingers, both thumbs, both hips, both knees, both ankles, both heels, and both feet.

2.2. Tracking performance comparison

The processing speed and tracking stability were compared for frontal face detection using a cascade classifier, tracking of arbitrary objects (frontal faces of persons) using a tracker (MIL, KCF), and skeletal detection using MediaPipe. A Razer Blade Stealth 13 (CPU: Intel Core i7-1165G7, GPU: GeForce GTX 1650 Ti) was used as the analysis computer.

The camera images were acquired in real time, with a resolution of HD (1280 × 720 px) and a frame rate of up to 30 fps. The programming language used was Python (version 3.8.12, development environment Anaconda3 for Windows 64-bit). The processing speed was evaluated as the frame rate for a series of processes from reading image frames from the camera to the tracking process. The evaluation of tracking stability was based on the presence or absence of tracking failures under the following six conditions:

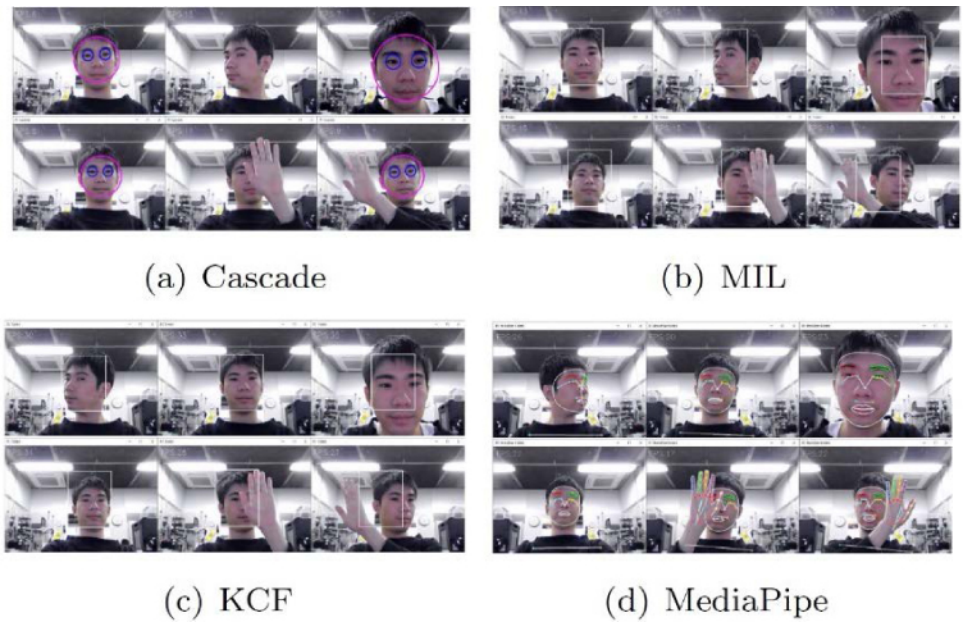


Figure 1. Tracking test example.

(a) Face forward at a distance of approximately 0.5 m from the camera
(b) Face rotated at an angle of approximately 45° at a distance of approximately 0.5 m from the camera
(c) Face on at a distance of approximately 0.3 m from the camera
(d) Face on at a distance of approximately 0.7 m from the camera
(e) Half cover the facing face with one hand
(f) Remove one hand from the face.

Figures 1 and 2 show the status of tracking by each method and a comparison of the frame rate (processing speed) time transition of each method, respectively.

Tracking by the cascade classifier accurately detected the frontal face and both eyes (three points in total) at a processing speed of around 10 fps, regardless of the perspective of the tracking target, but could not detect oblique faces or faces covered by one hand.

Tracking by MIL and KCF was possible at processing speeds of around 15 fps or 30 fps, respectively, but the ROI range did not change depending on the perspective of the tracking target, which raises questions about the reliability of the coordinates. In addition, it was found that there was a problem with the stability of the tracking as the ROI deviated from the face from (e) to (f).

The processing speed of KCF is sometimes as fast as 60 fps but sometimes drops to about 10 fps, so stable processing cannot be expected.

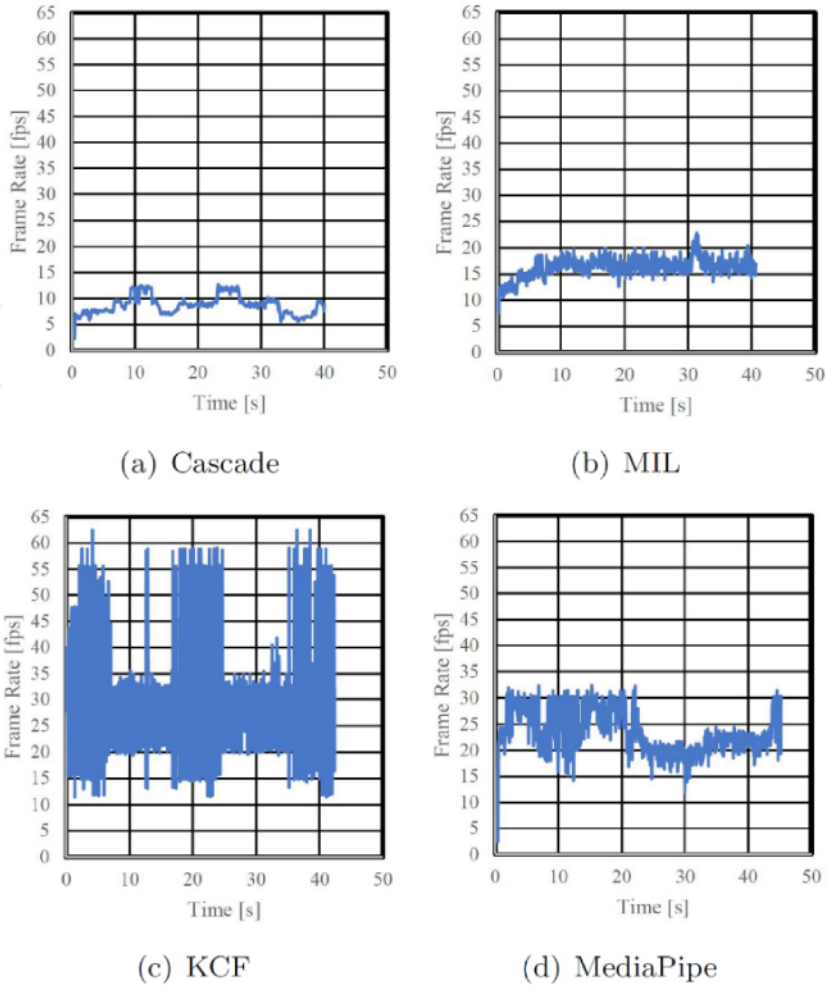


Figure 2. Tracking test frame rate.

Table 1. Stability, frame rate, and tracking points.

Software	Frame rate [fps]	Tracking stability	Tracking points
Cascade	10	Low	3
MIL	15	Middle	1
KCF	30	Middle	1
MediaPipe	20	High	33

The tracking by MediaPipe was found to be stable at around 20 fps with no tracking failure in any of the cases (a)–(f).

Table 1 shows a comparative evaluation of the above results in terms of processing speed, tracking stability, and number of tracking points (the amount of information that can be acquired). The best balance was achieved by the tracking by MediaPipe.

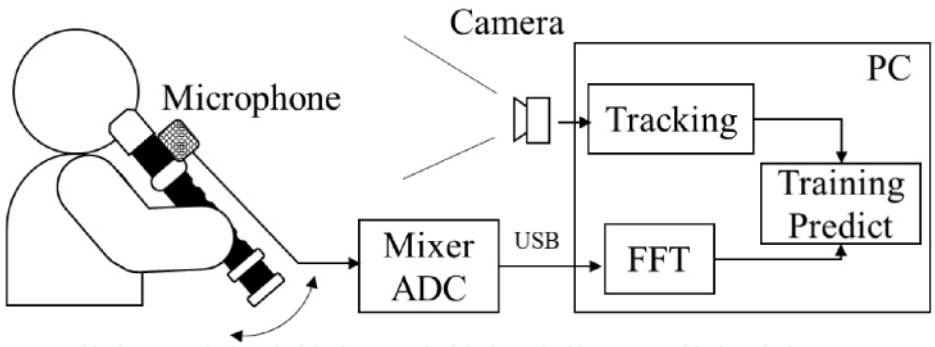


Figure 3. System block diagram.

MediaPipe was subsequently adopted as the method for detecting the player's movements from the camera images.

3. Detection of player movements utilizing MediaPipe and MLP classifier

3.1. System configuration

Figure 3 shows the block diagram of the system for detecting player motion. The audio is captured by the microphone attached to the recorder and then processed through a mixer and AD converter. The fundamental tone is extracted using FFT on a PC. At the same time, the camera captures the player's movements, including signals indicating the start or end of the performance. The movement tracking data is analyzed on the PC, and machine learning is used to learn and estimate the hit points. As described in the previous section, Python was used as the programming language.

3.2. Player motion detection method

We opted for MediaPipe as our tracking system due to its commendable frame rate, tracking stability, and an extensive array of track points, totaling 33. MediaPipe was installed using the pip command without any special procedures as described on the official site [17]. MediaPipe includes various functions, but the function used in this study is the Pose landmark detection function, which detects the skeletal structure of the face, body, hands, and feet. The two parameters for skeletal detection were set as follows: the minimum confidence score for pose detection to be considered successful was set to 0.5, and the minimum confidence score for pose tracking to be considered successful was set to 0.5. The player motion detection method is shown in Figure 4. We chose to detect time-series motion using the scikit-lean multilayer perceptron (MLP) classifier, which is a well-known machine learning library in

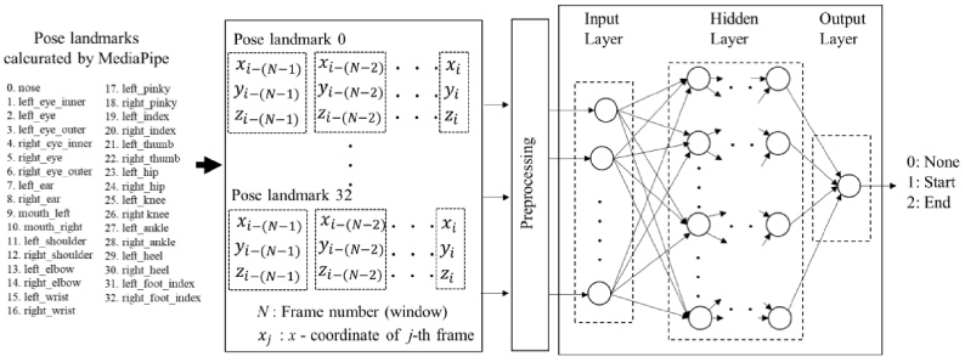


Figure 4. Flow of motion detection by MediaPipe and MLP classifier.

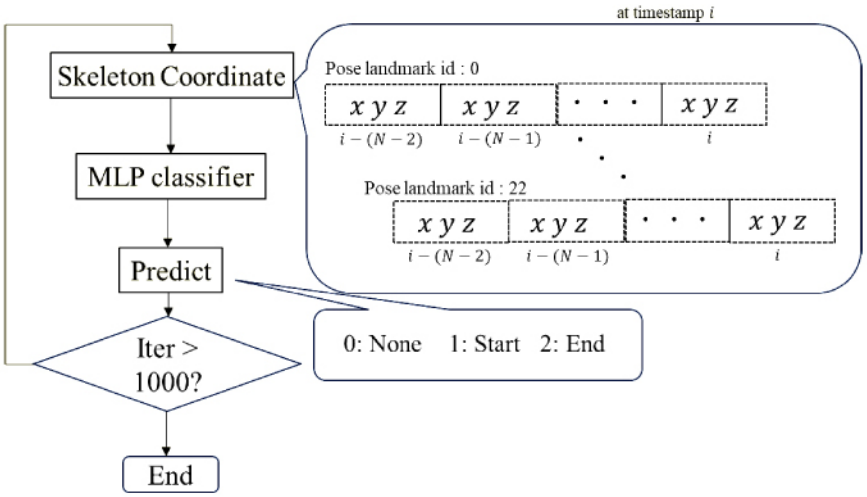


Figure 5. Flow diagram of the training process with MLP classifier.

Python. The aim is to identify the start of the performance (when sound transitions from silence to an acoustic state), labeled 1, and the end of the performance (when sound shifts from an acoustic state to silence), labeled 2, during training. All other instances are labeled 0. When features are input into the input layer, multiple neurons in the hidden layer process them, resulting in predictive discrimination outcomes in the output layer.

A flow diagram of the training process using the MLP classifiers is shown in Figure 5. The 3D coordinates of the historical skeleton within the range specified by the frame number *N* are input to the input layer from time to time, and the target label is used as the training data for learning.

3.3. Examination of player motion detection

In the past, only the prediction of the performance start point was reported [19], but this paper describes the results of the prediction of both the performance start and

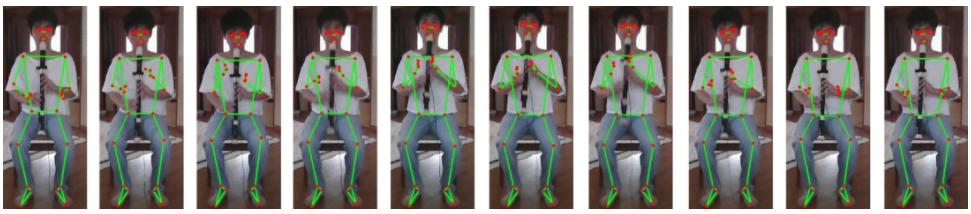


Figure 6. Example of time-series data at the start of a performance.

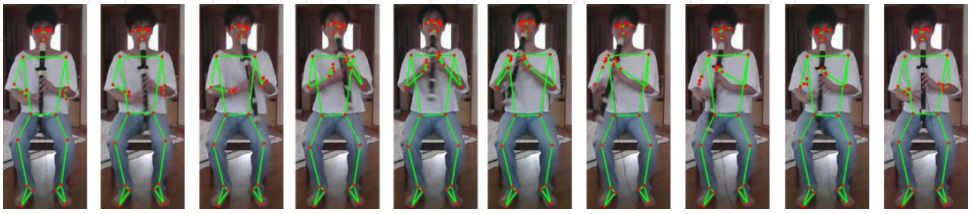


Figure 7. Example of time-series data at the end of a performance.

end points. The actions of the player at the start and end of the performance are shown in Figures 6 and 7, respectively. The cue at the start of the performance is almost exclusively a vertical movement, while the cue at the end of the performance includes an elliptical left–right movement. The results of skeletal detection during this operation are shown in a time series in Figure 8. The coordinate axes are positive in the x-direction to the right, positive in the y-direction downwards, and positive in the z-direction in the depth direction. The vertical axis of the data on the yellow-green spikes is the right vertical axis and indicates the target: when the target is 1, it is the start point of the performance and when the target is 2, it is the end point of the performance. The other lines are the coordinate transitions for each joint. The joint points below the waist, whose coordinates hardly change during the performer's movements, are excluded.

The parameters of the MLP classifier are considered through a grid search. Optimization methods such as Adam, L-BFGS, and SGD are considered along with activation functions like identity, logistic, ReLU, and tanh. In terms of features, the frame counts for tracking coordinates are configured to 10, 20, and 30 frames with and without preprocessing (relative coordinate conversion). The coordinates of the player's entire body are included as 3D coordinates for each joint, with the midpoint of the waist serving as the origin. While the previous report [19] dealt only with the performance start point, this paper deals with both the performance start point and the performance end point. Targets are labeled 1 for the moment when the player initiates music, 2 for the moment when the player concludes, and 0 for all other instances. The dataset consists of 20 sessions, with 10 allocated for training and the remaining 10 reserved for prediction. Timing detection is considered accurate when

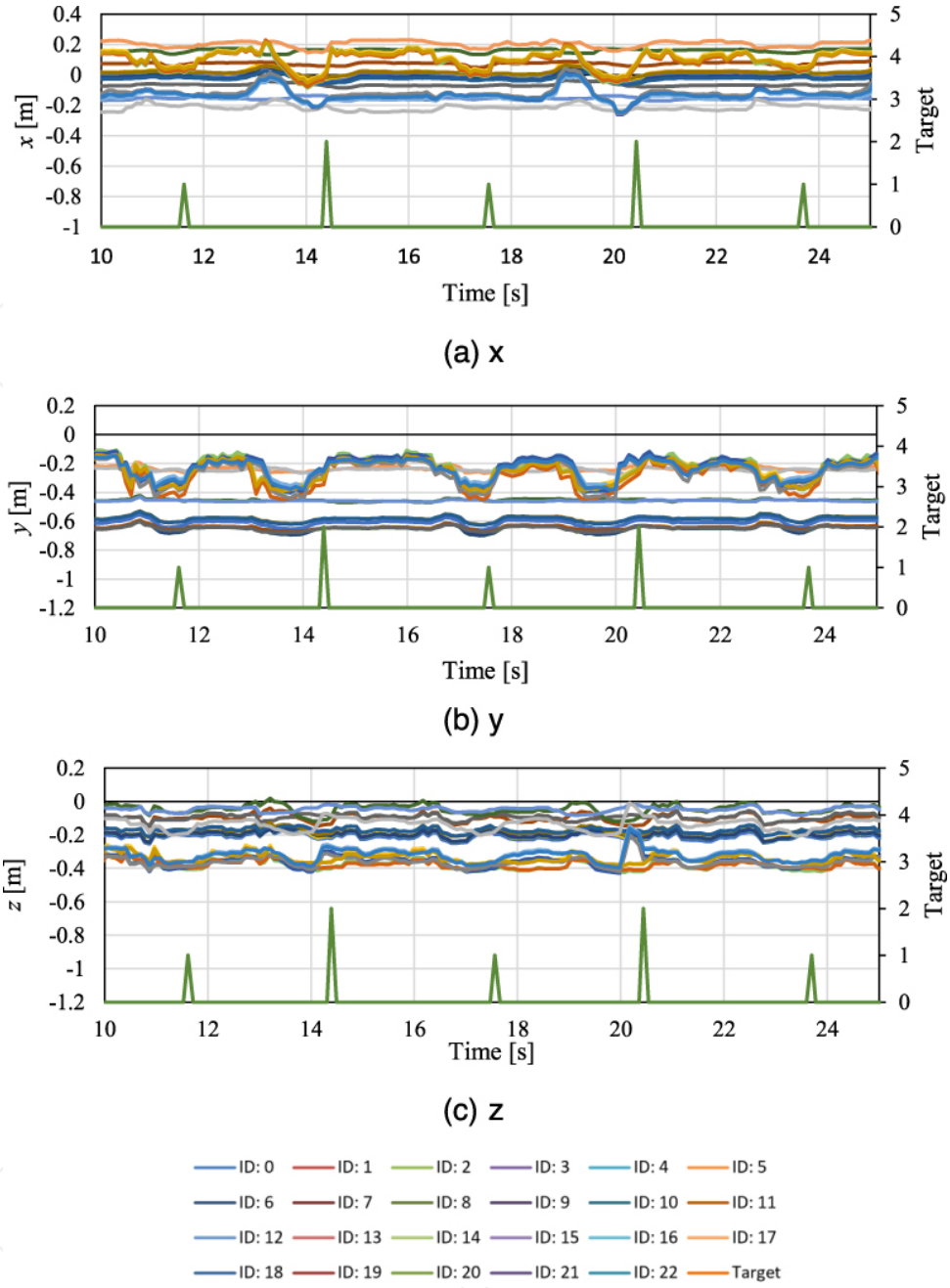


Figure 8. Time trends of skeleton-detected joints by MediaPipe at the start and end points of the performance.

within ±0.5 s. The prediction is true positive (TP) if the player's action was predicted at a time when it should have been detected. It is false positive (FP) if it was predicted when it should not have been detected. It is false negative (FN) if it was not predicted when it should have been detected. The following equations were used to evaluate accuracy and precision. In the present study, the true negative (TN) was

Table 2. Grid search results sorted by accuracy of start motion.

Frame number	Coordinate	Hidden layer	Activation function	Optimizer	Start [%]		End [%]		Predicted time [ms]
					Acc.	Prec.	Acc.	Prec.	
10	Absolute	2, 10	identity	L-BFGS	72.7	88.9	20.0	100.0	3.3
10	Relative	9, 10	identity	L-BFGS	71.4	71.4	0.0	—	4.0
10	Relative	100, 10	ReLU	L-BFGS	70.0	100.0	100.0	100.0	5.4
10	Relative	100, 90	identity	L-BFGS	66.7	80.0	0.0	—	5.2
10	Relative	5, 10	identity	L-BFGS	66.7	66.7	0.0	—	3.8
10	Relative	100, 60	identity	L-BFGS	64.3	69.2	0.0	—	4.8
10	Relative	20, 100	identity	L-BFGS	64.3	69.2	0.0	—	3.6
10	Relative	1000, 400	tanh	L-BFGS	64.3	69.2	0.0	—	41.7
10	Relative	1000, 400	identity	L-BFGS	61.5	72.7	0.0	—	29.2
10	Relative	70, 100	ReLU	L-BFGS	60.0	100.0	100.0	100.0	5.3

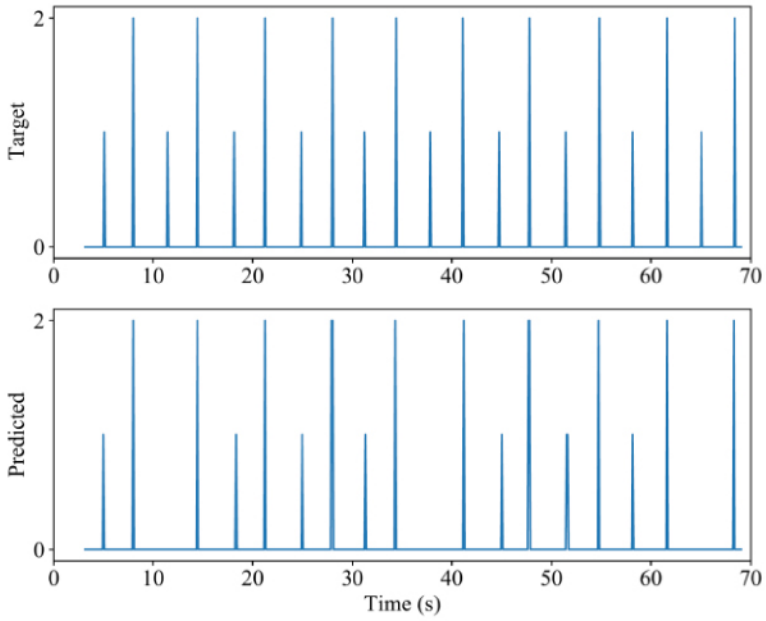
not used because it would have been irrational to include it in the calculations due to the long period of time in which the player was not moving or was in a state of continuous motion.

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{1}$$

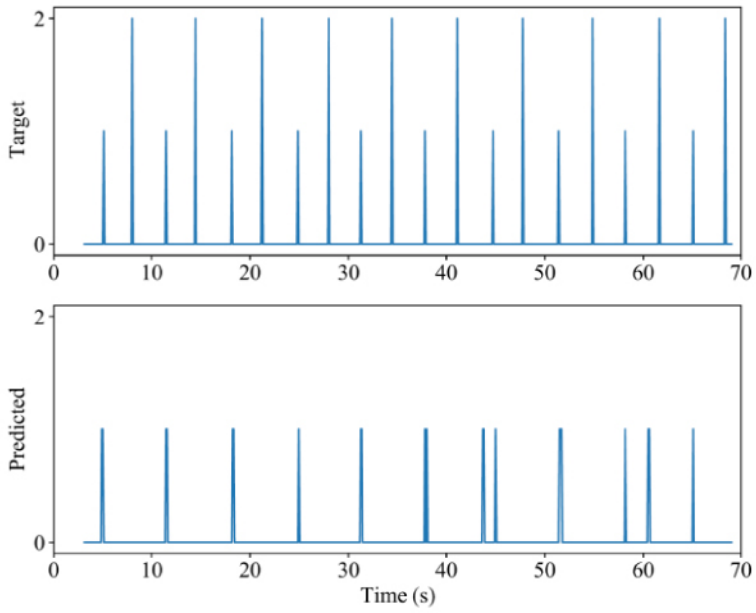
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2}$$

Tables 2 and 3 display the results of the grid search, presenting 10 accuracy estimates for the start and end of the performance in ascending order, respectively. The accuracy of the start of the performance was approximately 70%, while the accuracy of the end of the performance was 100% in many cases. This difference in classification accuracy is thought to be due to the fact that the start of a performance is mostly characterized by an up-and-down movement, whereas the end of a performance is characterized by a circular movement. Therefore, the circular movement at the end of a performance makes it an easy feature to classify. It was also found that as few hidden layers as possible are desirable as the prediction time exceeds 10 ms if either the length or width of the hidden layer is greater than 1000. Examples of successful and unsuccessful predictions are shown in Figure 9. When the frame number was 10, the coordinate was relative, the hidden layer was (100, 10), the activation function was ReLU, and the optimizer was L-BFGS, the accuracy and precision were high, and the estimation speed was fast. The results were good with high accuracy, precision, and fast estimation speed.

However, a traditional method for detecting changes is to differentiate the change in coordinates and detect them with a certain threshold value. Figure 10 shows a time-differentiated graph of the skeletal coordinate output from MediaPipe for



(a) 10 frames, relative coordinates, hidden layer (100, 10), relu activation, lbfgs optimizer



(b) 10 frames, relative coordinates, hidden layer (9, 10), identity activation, lbfgs optimizer

Figure 9. Example comparison between targeted and predicted values with different parameters used in the MLP classifier.

Table 3. Grid search results sorted by accuracy of end motion.

Frame number	Coordinate	Hidden layer	Activation function	Optimizer	Start [%]		End [%]		Predicted time [ms]
					Acc.	Prec.	Acc.	Prec.	
10	Relative	100, 10	ReLU	L-BFGS	70.0	100.0	100.0	100.0	5.4
10	Relative	70, 100	ReLU	L-BFGS	60.0	100.0	100.0	100.0	5.3
10	Relative	100, 100	ReLU	L-BFGS	50.0	100.0	100.0	100.0	5.5
10	Absolute	200, 1000	ReLU	L-BFGS	36.4	80.0	100.0	100.0	19.8
10	Absolute	10, 3	tanh	L-BFGS	30.0	100.0	100.0	100.0	3.6
10	Absolute	100, 70	ReLU	L-BFGS	27.3	75.0	100.0	100.0	5.5
20	Relative	10, 8	ReLU	L-BFGS	21.4	42.9	100.0	100.0	5.8
20	Relative	100, 70	ReLU	L-BFGS	21.4	42.9	100.0	100.0	9.7
10	Absolute	60, 100	ReLU	L-BFGS	20.0	100.0	100.0	100.0	5
10	Relative	200, 1000	ReLU	L-BFGS	20.0	100.0	100.0	100.0	18.1

prediction. The horizontal axis represents time, the left vertical axis represents the time derivative of the 3D coordinate transition of each skeleton from ID0 to ID22, and the scale of right vertical axis is target (0, 1, or 2), which indicates the start and end points of the performance. Although there are some areas where the derivative peaks near the target, we can see that it actually captures changes during the swinging motion prior to the hitting point and that it is difficult to set a certain threshold value. These results also demonstrate the effectiveness of the proposed method for learning the temporal transitions of skeletal coordinates.

4. Future research for improving start time detection accuracy

The following measures can be considered to improve the detection accuracy of the start time of a performance in the future. The first is the addition of training data.

Second, each joint coordinate is weighted to increase the sensitivity to the vertical motion of the hands and face. Some joints do not fluctuate significantly in response to movement, whereas others fluctuate significantly. In this experiment, all joints were treated with the same weights; however, increasing the weights of the parts with large fluctuations is expected to improve the accuracy of motion detection.

The third step is the addition of breath sounds and acceleration sensors. In the case of wind instruments, a breath sound is generated before performance. If the resolution of the image is insufficient, the accuracy can be improved by adding not only the image but also the output of the acceleration sensor to the learning process to make it multimodal.

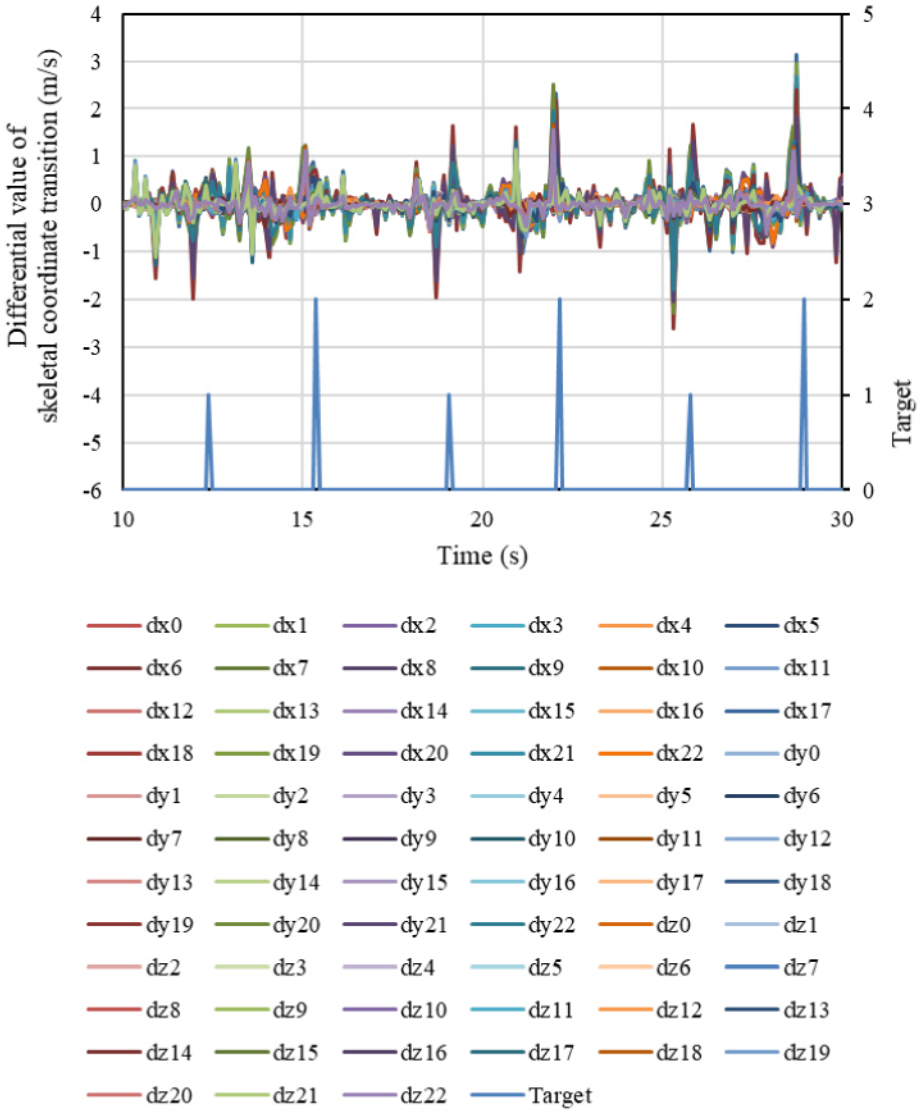


Figure 10. Differential value of skeletal coordinate transition.

5. Conclusion

To achieve highly cooperative tasks between humans and robots in the future, this paper proposed a method for detecting actions related to musical performances. The following conclusions were drawn.

(1) To track the performer's movements, we evaluated various methods including 2D image processing and tracking methods such as cascade, MIL, and KCF as well as skeletal detection by MediaPipe. Our evaluation showed that skeletal detection by MediaPipe was the most appropriate method due to its stability and speed performance.

(2) An algorithm was implemented to estimate the start and end points of the performer's performance by learning and estimating the time transition of the skeletal coordinates detected by MediaPipe using a deep neural network.

(3) A 10-set training data and 10-set test data estimation experiment was conducted. The start point of the performance had an accuracy of 70%, while the end point had an accuracy of 100%.

In the future, the accuracy of estimation will be improved and integrated with the performance system.

Conflict of interest

The authors declare no conflict of interest.

Acknowledgments

This work was supported by JSPS Grants-in-Aid for Scientific Research, Grant Number JP19K04300.

The authors would like to thank alumnus Ryusuke Ishikawa for building the experimental apparatus and conducting the experiments.

References

1 UNIVERSAL ROBOTS *[Internet]*, [cited 2024 Aug 20]. Available from: https://www.universal-robots.com/applications/machine-tending/.

2 ISO 10218-1: 2011. Robots and robotic devices Safety requirements for industrial robots Part 1: Robots, 2011.

3 ISO 10218-2: 2011. Robots and robotic devices Safety requirements for industrial robots Part 2: Robot systems and integration, 2011.

4 ISO/TS 15066:2016. Robots and robotic devices Collaborative robots, 2016.

5 Ichinose S, Mizuno S, Shiramatsu S, Kitahara T. Two approaches to supporting improvisational ensemble for music beginners based on body motion tracking. *Int J Smart Comput Artif Intel*. 2019;**3**(1):55–70.

6 Ichinose S, Mizuno S, Shiramatsu S, Kitahara T. Improvisation ensemble support systems for music beginners based on body motion tracking. In: *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Hamamatsu, Japan*. Piscataway, NJ: IEEE; 2017. p. 796–800.

7 Shibata T, Tanaka M. Development of a forearm motion learning-assist system for playing the Japanese shamisen instrument. *Entertainment Comput*. 2023;**46**: 100564.

8 Lee JY, Kim JY, Min JS, Kim HJ. In: *A feature selection technique for hand gesture recognition in music display system, computer science and its applications*. Lecture notes in electrical engineeringvol. 330, Berlin, Heidelberg: Springer; 2015.

9 Kajitani M. Development of musician robots. *J Robot Mechatron*. 1989;**1**(3):254–255.

10 Takagi S. Toyota partner robots. *J Robot Soc Japan*. 2006;**24**(2):208–210. (in Japanese).

11 Lim A, Mizumoto T, Ogata T, Okuno HG. A musical robot that synchronizes with a coplayer using non-verbal cues. *Adv Robot*. 2012;**26**(3–4):363–381.

12 Horiuchi Y, Nishida M, Ichikawa A. Accompaniment system using cue by breath. *Trans Inform Process Soc Japan*. 2009;**50**(3):1079–1089. (in Japanese).

13 Maezawa A, Goto M, Okuno HG. Query-by-conducting: an interface to retrieve classical-music interpretations by real-time tempo input. In: *11th Int. Society for Music Information Retrieval Conf. (ISMIR 2010)*. Universiteit Utrechtp. 477–482.

14 Maezawa A, Yamamoto K. MuEns: a multimodal human-machine music ensemble for live concert performance. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. New York, NY: Association for Computing Machinery; May, 2017. p. 4290–4301.

15 Gann K. Just Intonation Explained *[Internet]*, [cited 2024 Aug 20]. Available from: https://www.kylegann.com/tuning.html.

16 Ishikawa R, Tobita K. Concept and prototype of the pure intonation ensemble system. In: *Proceedings of The 11th TSME Int. Conf. on Mechanical Engineering, DRC001*. Ubon Ratchathani, Thailand: 2020. p. 150–156.

17 MediaPipe *[Internet]*, *MediaPipe Solutions guide* [cited 2024 Aug 20]. Available from: https://developers.google.com/mediapipe.

18 Tobita K, Ishikawa R, Mima K. Study on ensemble between humans and robots: consideration on Einsatz detection method. In: *Proceedings of ROBOMECH2022, 1P1-J02*. 2022 (in Japanese).

19 Tobita K, Ishikawa R, Mima K. Study on human-robot ensemble: experiments on performance timing detection and pure chord playing. In: *2022 JSME-IIP/ASME-ISPS Joint Int. Conf. on Micromechatronics for Information and Precision Equipment (MIPE), ODVP-22*. Nagoya, Japan: August, 2022.