Repeatable Folding Task by Humanoid Robot Worker Using Deep Learning

Pin-Chu Yang, Kazuma Sasaki, Kanata Suzuki, Kei Kase, Shigeki Sugano, and Tetsuya Ogata

***Abstract*—We propose a practical state-of-the-art method to develop a machine-learning-based humanoid robot that can work as a production line worker. The proposed approach provides an intuitive way to collect data and exhibits the following characteristics: task performing capability, task reiteration ability, generalizability, and easy applicability. The proposed approach utilizes a real-time user interface with a monitor and provides a first-person perspective using a head-mounted display. Through this interface, teleoperation is used for collecting task operating data, especially for tasks that are difficult to be applied with a conventional method. A two-phase deep learning model is also utilized in the proposed approach. A deep convolutional autoencoder extracts images features and reconstructs images, and a fully connected deep time delay neural network learns the dynamics of a robot task process from the extracted image features and motion angle signals. The "Nextage Open" humanoid robot is used as an experimental platform to evaluate the proposed model. The object folding task utilizing with 35 trained and 5 untrained sensory motor sequences for test. Testing the trained model with online generation demonstrates a 77.8% success rate for the object folding task.**

***Index Terms*—Humanoid robots, learning and adaptive systems, motion control of manipulators, neurorobotics.**

I. Introduction

With declining birth rates and increasing longevity, future labor shortages are anticipated. Labor shortages will result in declining efficiency and increasing costs, and manufacturers who produce essential consumer products may not be able to afford the increased costs, which will eventually affect society as a whole. In addition, as market demands change, the manufacturing industry is switching to small-quantity, multi-variety production. Consequently, more production facilities that can handle various products will be required [1]. Such facilities must be more flexible and able to perform different tasks for different products. To achieve this, robots with multiple degrees-of-freedom arms, image recognition systems, and massive databases with pre-designed motion sequences represent a possible solution. However, this solution may not be suitable for factories that produce or disassemble similar types of products. It is nearly impossible to utilize current pre-design methods to substitute these jobs from human worker. When producing similar types of products, manufacturing tasks are repetitive and tend to require limited human intelligence. Thus, it is anticipated that future labor shortages will affect this type of production.

Recently, artificial neural network learning models with "deep" structure (i.e., deep neural networks) have demonstrated success in image and speech recognition. It is expected that this machine learning method will be applied to robot tasks; however, constructing a machine learning model for an unstructured environment requires considerable training data.

Normally, direct teaching and imitation learning methods are utilized for data collection; however, hardware limitations and control issues can hinder the construction of a system suitable for tasks. The target of this study is a machine learning-based model to control robots that can perform tasks in an uncertain environment, such as a production line with human workers. Therefore, we must consider practical applications, the convenience of an operational system, the ability to perform repetitive tasks automatically, and the ability to perform multiple tasks. To satisfy these requirements, we set the following objectives.

I: **Task capability**:
The robot has sufficient knowledge and is capable of performing a given task.

II: **Reiteration ability**:
The robot can perform tasks repeatedly.

III: **Generalizability**:
The robot can perform multiple tasks and the same task with similar objects.

IV: **Easy applicability to a factory robot (non-backdrivable robot)**:
The proposed method should be easy to apply with few limitations.

To evaluate performance and test practical cases, folding a soft object was selected as the task due to the difficulty of a pre-

Manuscript received September 9, 2016; accepted November 8, 2016. Date of publication November 29, 2016; date of current version December 26, 2016. This paper was recommended for publication by Associate Editor M. Asada and Editor D. Lee upon evaluation of the reviewers' comments. This work was supported in part by the AIST, "Fundamental Study for Intelligent Machine to Coexist with Nature" from the Research Institute for Science and Engineering, Waseda University and in part by an MEXT Grant-in-Aid for Scientific Research (A) 15H01710.

P.-C. Yang is with the Department of Modern Mechanical Engineering, Graduate School of Creative Science and Engineering, Waseda University, Tokyo 169-8050, Japan, and also with the Artificial Intelligence Research Center, Tsukuba 305-8560, Japan (e-mail: kcy.komayang@gmail.com).

K. Sasaki is with the Department of Intermedia Art and Science, School of Fundamental Science and Engineering, Waseda University, Tokyo 169-8050, Japan (e-mail: ssk.sasaki@suou.waseda.jp).

K. Suzuki, K. Kase, and T. Ogata are with the Department of Intermedia Art and Science, School of Fundamental Science and Engineering, Waseda University, Tokyo 169-8050, Japan, and also with the Artificial Intelligence Research Center, Tsukuba 305-8560, Japan (e-mail: suzuki@idr.ias.sci.waseda.ac.jp; kase@idr.ias.sci.waseda.ac.jp; ogata@waseda.jp).

S. Sugano is with the Department of Modern Mechanical Engineering, Graduate School of Creative Science and Engineering, Waseda University, Tokyo 169-8050, Japan (e-mail: sugano@waseda.jp).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2016.2633383

designed pipeline method to accomplish this type of task. The folding task is a well-known and difficult manipulation task that can be used to address unpredictable changes during the folding process with robots due to dynamic environmental information, as mentioned in the previous section. The generalization and reiteration abilities can be tested with objects that never be used in the training data. This task will be used in situations without specific setup and calibration to demonstrate the effectiveness of the proposed model. Experimental results indicate that the proposed model holds promise for realizing a smart robot worker, and it is expected that the model can be applied to various tasks.

The rest of the paper is organized as follows. Section II introduces related works that have been done. The challenges and the contributions of this paper are also described in this section. Section III introduces the approach of this study, which includes the introduction to the process of gathering data, architecture of models and how it be trained. Section IV describes the experiment setting including robot's motions, objects to be used for collecting training dataset, parameters of model training and the results. Section V discusses the robustness and the possible method to boost the performance of task doing. Section VI concludes the result and demonstrates the effectiveness of proposed approach.

II. Related Work

The ability of robots to perform manipulation tasks, such as object grasping and inserting bolts, has been investigated [2]. More complicated tasks such as folding clothes [3], [4] and wiring cables [5], [6] have also been studied. These studies have demonstrated that such tasks can be performed by various approaches; however, the success rate of these methods typically depends on human-designed control, image feature extraction, and environment. The approaches proposed in previous studies might be difficult work to apply to an environment characterized by complex and uncertain situations.

Researchers have attempted to develop methods for manipulation tasks and incorporate some type of smart control. The deep learning method has been applied to static image recognition [7]. This method achieved a recognition rate considerably greater than human recognition ability. Super-resolution convolutional networks [8] inspired the Internet-based waifu2x super-resolution service [9]. Deep learning has also been used to train sequential data, e.g., in self-driving car research [10], and for undefined object grasping [11]–[13].

A machine learning method has also been applied to robot tasks [14]. This research proposed a reinforcement learning method to train a trajectory policy and robot arm actuator torque signals; however, tasks that require different amounts of time and are performed under different operational conditions must be trained separately. Noda *et al.* [15], proposed a model that considered multiple-behavior learning and information from multiple modalities automatically. Their model combined two fully connected neural networks, i.e., an image feature extraction network and a dynamic learning model. Noda *et al.* successfully applied their model to multiple periodic motion behaviors using NAO, a hobby-sized humanoid robot. Suzuki *et al.* adopted a structure similar to that proposed by Noda *et al.* and successfully used a PR2 robot for the soft object folding task [16]. However, their method cannot perform task repeatable and [16], [15] both use a direct teaching method that can only be applied to backdrivable robots. In addition, with these methods, it is difficult to determine the effectiveness of sensory-motor data.

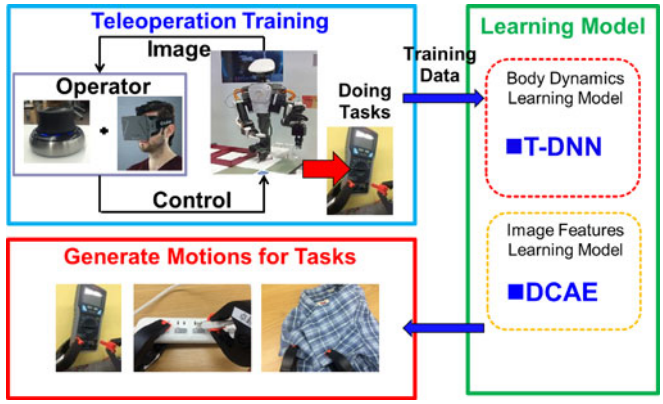

Fig. 1. Process of the proposed approach in 3 main phases.

The current study focuses on the effects of dynamic information on interactive robot-environment information (sensory-motor information) and uses a humanoid robot and a folding task to evaluate the proposed method. Based on our approach, it is effective to make industrial level humanoid robot be able to do tasks that requires high adaptability instead of pipeline method. Moreover, through our architecture, the effort of designing evaluation function for reinforcement learning can be eliminated and substituted by direct learning from experimenter's operating experience.

III. Approach

We present an approach to achieve a deep learning method that can be applied to an adaptable task-performing humanoid robot operating in an uncertain environment. For objectives **I** and **III**, to achieve sufficiently high generalizability suitable for an uncertain environment, deep learning is applied to learn the sensory-motor information acquired from the robot. That information grants proposed model to perform task operation. In order to achieve objective **II** with deep learning, all sequences are designed to begin and end at the same robot pose. For objective **IV**, the teleoperation technique is used to acquire data that can ignore the robots configuration.

To apply deep learning to a robot task, (1) data collection, (2) training, and (3) task generation phases are required. The flow of the approach is shown in Fig. 1 where "Teleoperation Training," "Learning Model," and "Generate Motions for Tasks" correspond to the data collection, training, and task generation phases, respectively.

A. Data Collection Phase

Data collection is an important step in deep learning and is particularly important for tasks that require precisely timed motions. Some deep learning research has used direct teaching to

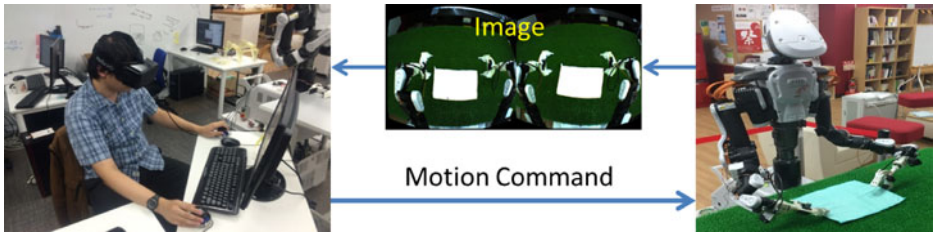

Fig. 2. Sensory-motor experience sharing: Using remote monitoring or a head-mounted display, the operator can operate the robot directly and sharing same sensation to acquire proper sensory-motor experience (data). This ensures that the collected sensory-motor data are able to operate by the sense of human and are expected to be effective for deep learning.

guide robot arms to perform tasks; however, direct teaching can only be applied to backdrivable robots that are typically used to manufacture specialized products. Such robots can be prohibitively expensive for factory that usually only equipped with non-backdrivable robots and requires capabilities for practical task operation. The objective of the current study is to propose an effective data collection method without robot limitations.

1) ***Teleoperation Training****:* Awano *et al.* applied the teleoperation method to collect training data and successfully realized humanrobot cooperative behavior [17]. The current study utilizes same technique makes it can be applied to almost all kinds of robots, particularly for non-backdrivable robots.

The teleoperation can involve any amount of control, from fully autonomous to complete manual control, as well as mixed-initiative interactions. Such a combined command procedure can provide advantages for data collection. A fully autonomous command system can utilize pre-designed motions and behaviors to reduce programming time for predictable behavior. A semi-autonomous command refers to self-autonomous behavior that requires guidance information, such as a guided missile used by the military. A semi-autonomous command process can provide satisfactory robot behavior for motions that require high precision. With manual control, a human operator directly controls the robot's actuators to perform tasks.

Some data, such as sensor signals and image data can be collected during teleoperation. These sequential data are collected directly from the robot with different autonomous command levels. Furthermore, sensory-motor data that contain robot motor angles and image data captured by the robot-mounted camera are collected for the training phase, as shown in Fig. 2.

B. Traning Model Phase

The collected data are used to train the deep learning model proposed in this study. The model comprises two parts, as shown as Fig. 3. With this two-step end-to-end training process, the proposed model can handle raw input data adaptively to deal with small changes in the environment and perform corresponding motions from the output command signal.

1) ***Deep Convolutional Autoencoder (DCAE)****:* Convolutional neural networks (CNNs) are powerful image-processing tools, particularly for image recognition. A CNN contains sliding filters, which are similar to biological cells that can exploit a strong response to a spatially local input pattern and cover the entire input image. CNNs can handle considerably more input dimensions than fully connected neural networks while

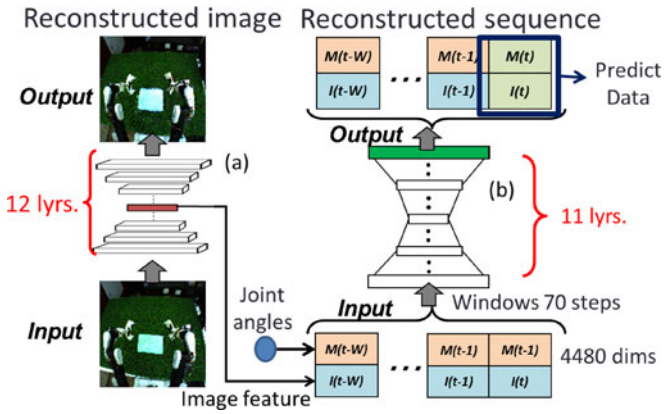

Fig. 3. Overview of the model: (a) a DCAE can extract the image and compress feature information. Half of the models structure (middle layer to output layer) is used to decode (reconstruct) information to extracted image features. (b) a TDNN learns and generates the time sequential data with window-size steps of the extracted image features and motion data acquired from the robot.

using fewer parameters. This greatly decreases the training time and enhances performance for image processing or similar data input. Furthermore, a model with a deep convolutional layer structure can extract data to different levels of features from edges to partial parts of the image.

This study utilized convolutional layers to present a DCAE that can handle a high-resolution image to the small size of feature map. Convolutional layers with a stride can extract features and down-sample the dimension of information. Deconvolutional layers are used to reconstruct images from the encoded feature map. With a trained DCAE, half of the models structure (the input layer to the middle layer) is used to encode (compress) the information to small-dimension image features compared to the original input image. These encoded image features can represent the state of an input image and provide high-resolution input information with fewer dimensions. Batch normalization is used to optimize learning and reduce the possibility of overfitting problems. The DCAE structure is shown in Table I. Networks are trained to reconstruct input image data at the output layer. In this study, training data for DCAE utilize sequential images acquired from the robot-mounted camera. The target of each input image is the original input data, and the mean square error (MSE) is used to modify the weight of neural networks by using Adam optimization [18].

2) ***Time-Delay Neural Network****:* TDNN is a fully connected feed-forward neural network trained for temporal sequence data

TABLE I
DETAILED STRUCTURE OF DCAE

DCAE	conv@3chs-conv@32chs-conv@64chs-conv@128chs-conv@256chs-fullBN@1000-fullBN@50-fullBN@1000-dconv@256chs-dconv@128chs-dconv@64chs-dconv@32chs-dconv@3chs

conv : Convolutional Layer
dconv: Deconvolutional Layer
fullBN: Fully connected layer + Batch normalization.

TABLE II
DETAILED STRUCTURE OF TDNN

TDNN	4480-4524-2262-1131-565-150-565-1131-2262-4524-4480

All links are fully connected with batch normalization.

with time step windows [19]. Noda *et al.* proposed a model that utilized a TDNN with multi-modality signals and achieved multi-behavior with NAO, a hobby-sized robot [15]. A TDNN with deep-structured layers can successfully reconstruct sequential data, and it is possible to generate a continuous sequence by shifting input information. A TDNN can be used for online generation that is executed by shifting the input window over time and repeatedly inputting the extracted image (camera image) features and motions (motor angles) in real time. The structure of the TDNN used in this study is shown in Table II.

The TDNN can learn sequential information with multiple sensory-motor signal inputs. Image features extracted from a DCAE and robot motion are applied in the TDNN model. The input of TDNN is a fixed windows size steps of data from dataset. During training, the target of each input data is the original input data, and the MSE is used to modify the weight by using Adam optimization. The TDNN training dataset is created by sliding the training data over time.

C. Task Generation Phase

The trained deep learning model is used with real-time sensory-motor information to make the robot perform a task. This is referred to as online generation. During every execution, the camera image firstly fed into the DCAE to compress the image information to the feature vector. Second, we combined these feature vector with joint angles in a designed window size trough time for inputting TDNN. Finally, we slide this window through time and continuously substitute last step combined information and input this slid sequential information into TDNN in order to generate predicted step, as shown in Fig. 4(a), where I denotes an image feature and M denotes motion at the corresponding time T.

These predicted step data can be used for the next step execution command sent to the robot. Notice that when proceeding sliding action to the window sized sequential information, the ex-motion that executed the last step is copied and then fed into the TDNN to address the problem of missing motion when executing sliding (Fig. 4(b). Moreover, for more stable and smooth

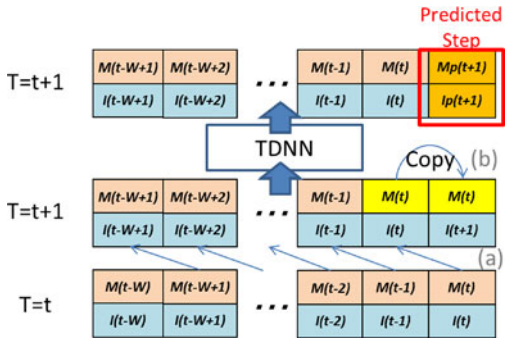

Fig. 4. Online generation is executed by sliding window through time with continuous inputting extracted image features and motions into TDNN in real time. (a): Combined sequential information slides through steps (b): Copy the previous motion to address missing motion problem.

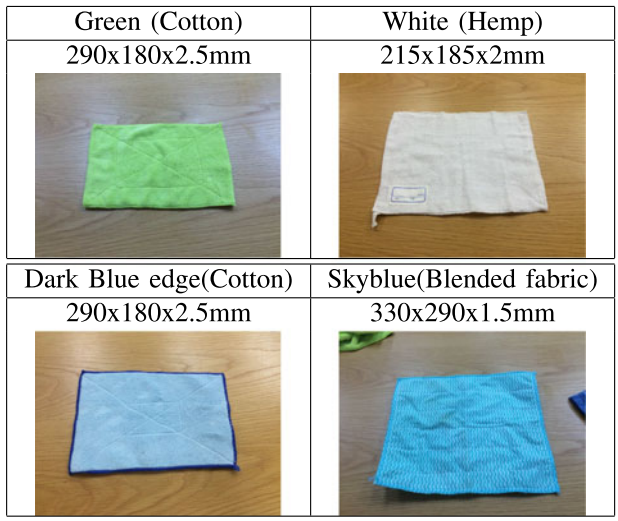

Fig. 5. Training Object Configurations.

motions, the executed command combines the outsourcing signals and the predicted signal rather than using the predicted steps directly. The combination of signals expressed in percentage can be calculated as follows:

$$\mathrm{Signal}_{\mathrm{exe}} = P \times \mathrm{Signal}_{\mathrm{out}} + (1 - P) \times \mathrm{Signal}_{\mathrm{pre}} \quad (1)$$

where P is the input percentage parameter of the combination, $\mathrm{Signal}_{\mathrm{exe}}$ is the execution command, $\mathrm{Signal}_{\mathrm{out}}$ is outsourcing signal, and $\mathrm{Signal}_{\mathrm{pre}}$ is the predict signal from TDNN. Finally, we evaluate the performance of the online generation result using a task-dependent estimation method.

IV. EXPERIMENT

The Nextage Open Robot from KAWADA Robotics is used as the experimental platform in this study [20]. This robot has two non-backdrivable six DOF arms and a mounted camera for precise task manipulation. The robot is placed in front of a grass sheeted table. The artificial grass sheet provides a buffer area to prevent damage when the robot performs much beyond the limited range. Here, the experimental task is a cloth-folding task where the cloth is placed randomly by experimenter. The motion behavior of the folding task is shown in Fig. 6 with four training

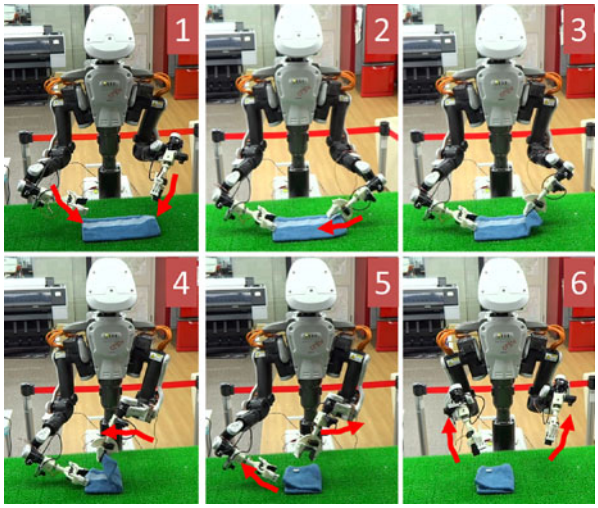

Fig. 6. Folding task behavior generated using three different teleoperation modes. About the details of motions, first a grab point (x,y position) is determined and the motion is executed automatically by an operator inputting an operation command (1–3). Second, when the robot performs a grabbing motion (generate the motion and close the gripper), the process control automatically switches to manual control mode to directly control the end-effector position using a 3D mouse (3–5). Finally, when the operator completes the operation, the "END" command button is pressed to allow the robot to finish the task and return to the home position to complete the actions (5–6). The sequence ends at the home position to create a limit cycle attractor to satisfy the task reiteration ability requirement.

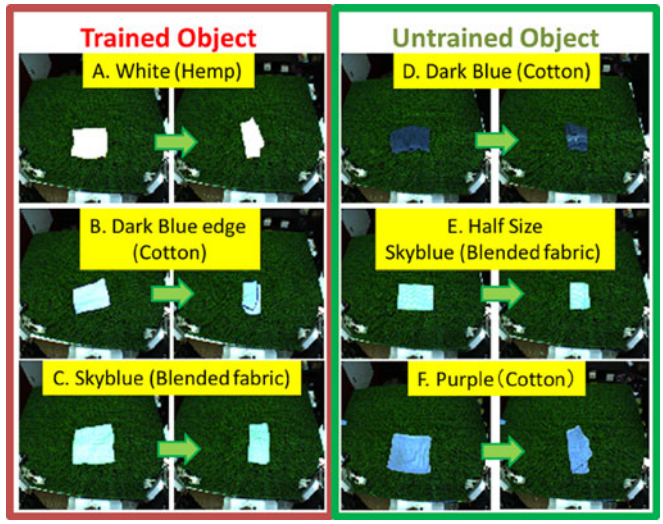

Fig. 7. Results of online generation with trained and untrained objects.

TABLE III
SUCCESS RATE EVALUATED BY BEHAVIORS

	Performed Behavior	Success Rate
Trained Cloth	Grabbed	88.9%(8/9)
	Grabbed+Folded	88.9%(8/9)
Untrained Cloth	Grabbed	77.8%(7/9)
	Grabbed+Folded	66.7%(6/9)
Total	Grabbed	83.3%(15/18)
	Grabbed+Folded	77.8%(14/18)

objects, and its configurations are shown in Fig. 5. Since the placing position, orientation and size of clothes are different, the visual information takes great contribution for robot performing task.

The training and test data for the training model are the sensory-motor data, which include motor angles and camera images captured from the robot. The camera image resolution is $112 \times 112 \times 3$chs (37,632 dimensions, RGB) and the motor angles with 12 DOF, each gripper signal per DOF (two DOF). The data are recorded at 10 FPS and each task sequence requires approximately 70 s. Approximately 28,000 steps of data are used for the training model.

A. Model Training

The DCAE is trained with learning rate $\alpha = 0.0002$, $\beta_1 = 0.75$(ADAM parameter), and mini batch size $= 200$ for the training and test data. The training takes 13,849 s (approximately four hours) using Chainer [21] with GPU calculation support for training 50,000 iterations. The TDNN is trained with learning rate $\alpha = 0.0002$, $\beta_1 = 0.7$ and mini batch size $= 250$ for the training and test data. Training takes 7,864 s (approximately two hours) using Chainer with GPU calculation support for training 70,000 iterations.

B. Motion Generation

First, we generate the trained and untrained sequences utilized during the training process. Continuous input in the image in training data is utilized to verify the performance of the trained model in online generation. During this process, the MSE is used to estimate prediction performance. The average prediction errors of motions are 0.00501 and 0.10682 per step for each sequence by associating 35 sequences of trained image data and five sequences of test image data, respectively. Second, we verify the success rate of tasks through online generation with trained and untrained objects. Here, three types of trained cloths with untrained position data and three untrained cloths are used for testing. Each cloth is placed randomly (shifting with small rotation) three times within the robots reach. The results are shown in Fig. 7.

To assess the task performance, we define the success rate using (a) performed behaviors and (b) area changed ratio.

a) Performed Behaviors: In this evaluation, "grabbed" and "folded" behaviors are evaluated to determine if the robot performs the task during online generation. The success rates for different behaviors are shown in Table III.

b) Area Changed Ratio: This evaluation utilizes area detection on a cropped image, where the area represents pixels and the cropped image always covers the whole cloth. Here, we define the Area Changed Rate (A.C. Rate), which detects area in both the start state (before Fig. 6(1)) and the end state (after Fig. 6(6)), and the difference between these two states is utilized to evaluate the success rate. The details of area change percentage are listed in Table IV, and the success rate due to different area changed percentages is shown in Fig. 8.

TABLE IV
AREA CHANGED RATE

Type	Num.	A.C. Rate	Type	Num.	A.C. Rate
A	(1)	65.95%	D	(10)	58.84%
	(2)	61.88%		(11)	60.06%
	(3)	62.13%		(12)	Failed
B	(4)	62.85%	E	(13)	58.32%
	(5)	Failed		(14)	Failed
	(6)	63.97%		(15)	62.57%
C	(7)	54.21%	F	(16)	59.32%
	(8)	54.82%		(17)	Failed
	(9)	58.84%		(18)	61.69%

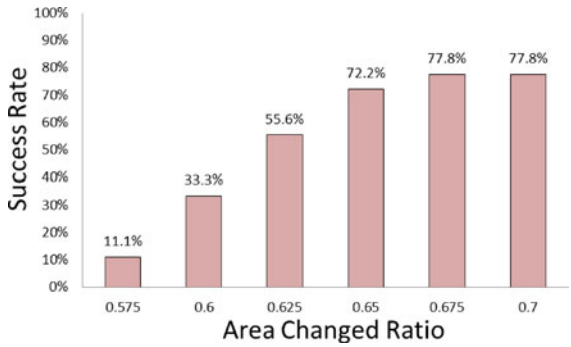

Fig. 8. Success rate relative to area changed percentage.

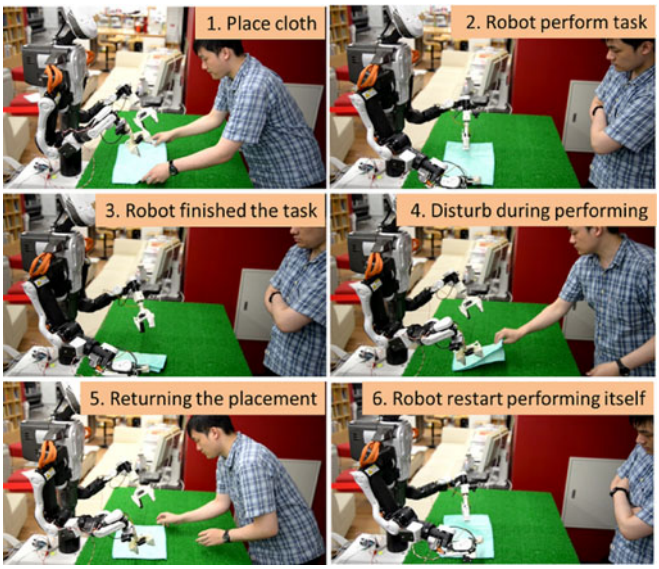

Fig. 9. Reiteration ability test experiment through human-robot interaction.

C. Reiteration Ability Test

To evaluate the reiteration ability, an experimenter stands in front of table facing to robot and disturbs the task while the robot performs the folding task (Fig. 9). It is confirmed that the robot can repeat the task even when disturbed during online generation, which proves the robustness of the proposed model.

Fig. 10. Online generation test with untrained object: Book-closing test.

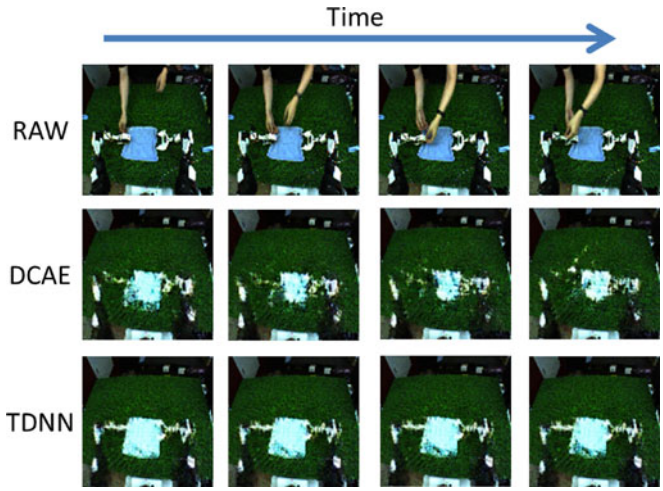

Fig. 11. Image feature reconstructed with untrained objects during human interference. The image information contains the human interference, and the DCAE + TDNN model can manage this information and perform the task.

D. Online Generation with Untrained Object

The cloth-folding task is very similar to the book-closing task; therefore, a book is used for testing with an untrained object. This task is performed successfully, as shown in Fig. 10.

V. DISCUSSION

Compared to previous research [16] with similarly structured training models, this study uses higher-dimension images and dual arms with 14 DOF, which greatly increase the difficulty in online generation. Their approach could only perform simple tasks but not well for long dynamical behaviors due to limitation of teaching manner and model. Therefore, we introduce a novel model based on teleoperation training manner and DCAE-TDNN model that could manage this problem. Through our experiments, the proposed model has shown a powerful ability for managing higher-dimension image data, and it has been proven that the model can provide a relatively stable signal for TDNN online generation. From experiment, the visual information is sufficient for task doing with the stable environment settings; however, it is expected the accuracy and adaptability for vary environment that can be improved by increase the modality as future prospection.

A. Robustness

Relative to robustness, the robot can perform tasks even when a human experimenter's hand disturbs the process, e.g., by changing objects or shifting object positions during experiment. This demonstrates the robustness of the proposed model in that it is capable of sustaining performance in a noisy environment. The proposed model can address the dynamics of sensory-motor information during generation (Fig. 11).

B. Increasing Operating Speed

Task performance speed is an important factor, especially for industrial applications. During online generation, this speed strongly depends on prediction and execution speed and the original training data sequence. The task performance speed is tested, and the task is performed successfully with four times less data than the original training sequence by reducing the teaching data sampling speed (one-half the original sequence length) and adjusting the prediction and execution speed (predict twice and operate once). We successfully test with four times faster generation speed (700 steps were reduced to approximately 180 steps; average of 70 s is reduced to approximately 18 s with a 10-FPS execution rate). As a result, it is possible to increase the robot's task performing speed.

VI. Conclusion

This study's goal is to propose a useful way to achieve a humanoid robot worker that can perform the folding task repeatedly with good generalizability. The soft object-folding task is chosen to evaluate the adaptability and effectiveness of proposed approach. In our experiments, training data are successfully collected in a teleoperation, and the proposed approach successfully allows a non-backdrivable humanoid robot to complete the folding task, which is verified by our experiments. As a result, our objectives are achieved and the proposed method is proven as a workable model for addressing difficult task problems.

In the future, the proposed model will be applied to multiple task sequences. More practical applications, such as picking up indeterminate shapes objects or more dynamic tasks, may be achieved. In addition, task performance speed is expected to equate that of predesigned pipeline methods and we expect to increase the variety of multiple task sequences.

References

[1] S. Y. Nof, *Handbook of Industrial Robotics*, 2nd ed. New York, NY, USA: Wiley, 1999.

[2] Y. Yamamoto *et al.*, "Task performance tests on inserting the bolts by an experimental system for power distribution line maintenance - grope action under compliance control," *Proc. 2012 Int. Symp. Micro-NanoMechatronics Human Sci., Nagoya*, Japan, pp. 290–293, 2012.

[3] J. Sindler, "World representation of a dual-arm robot manipulating with clothes," Master Thesis, Center Mach. Perception, Dep. Cybernetics, Faculty Elect. Eng., Czech Tech. Univ., Czech Republic, 2013.

[4] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbel, "A geometric approach to robotic laundry folding," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 249–267, 2012.

[5] Y. Koishihara and K. Yamazaki, "Wiring with hooking of a string by a dual-armed robot," in *Proc. 2016 JSME Conf. Robot. Mechatronics*, 2016, no. 16-2, Paper 2P1-03b3.

[6] K. Mukai, T. Matsuno, A. Yanou, and M. Minami, "Shape modeling of a string and recognition using distance sensor," *Proc. IEEE 24th Int. Symp. Robot Human Interactive Commun.*, 2015, pp. 363–368.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[8] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Dec. 2014.

[9] "waifu2x," 2016. [Online]. Available: http://waifu2x.udp.jp/index.html

[10] M. Bojarski *et al.*, "End to end learning for self-sriving cars," arXiv:1604, pp. 1–9, 2016. [Online]. Available: http://arxiv.org/abs/1604.07316

[11] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. Symp. Exp. Robot.*, 2016.

[12] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50 K tries and 700 robot hours," in *Proc. 2016 IEEE Int. Conf. Robot. Autom.*, Stockholm, Sweden, 2016, pp. 3406–3413.

[13] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, no. 4/5, pp. 705–724, 2015.

[14] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-End training of deep visuomotor policies," *J. Mach. Learn. Res. 17*, vol. 17, no. 35, pp. 1–40, Apr. 2016.

[15] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Multimodal integration learning of robot behavior using deep neural networks," *Robot. Auton. Syst.*, vol. 62, no. 6, pp. 721–736, 2014.

[16] K. Suzuki, K. Takahashi, C. Gordon, and T. Ogata, "Motion generation of flexible object folding task applied on humanoid robot using deep learning," 78th Nat. Convention Inf. Process. Soc. Japan, 2016.

[17] H. Awano, T. Ogata, S. Nishide, T. Takahashi, K. Komatani, and H. G. Okuno, "Human–robot cooperation in arrangement of objects using confidence measure of neuro-dynamical system," in *Proc. 2010 IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2010, pp. 2533–2538.

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2014, pp. 1–13.

[19] K. Lang, A. Waibel, and G. E. Hinton, "A time-delay neural network for isolated word recognition," *Neural Netw.*, vol. 3, pp. 23–43, 1990.

[20] Nextage.kawada.jp, "Kawada Robotics: Nextage Open," 2016. [Online]. Available: http://nextage.kawada.jp/

[21] Chainer.org, "Chainer," 2016. [Online]. Available: http://chainer.org/