Adaptive Drawing Behavior by Visuomotor Learning using Deep Learning

ディープラーニングを用いた視覚運動学習による適応的な描画行為

July 2018

Kazuma SASAKI
佐々木 一磨

Adaptive Drawing Behavior by Visuomotor Learning using Deep Learning

ディープラーニングを用いた視覚運動学習による適応的な描画行為

July 2018

Waseda University,
Graduate School of Fundamental Science and Engineering,
Department of Intermedia Studies,
Research on Intelligence Dynamics and Representation Systems

Kazuma SASAKI
佐々木 一磨

Abstract

Drawing pictures enables humans to represent concepts. Even when an object in the real world is represented a picture that is not exactly the same as the object, human beings possess the ability of recognizing the real-world object. We can also represent concepts by producing motor activities of drawing.

Studies in cognitive neuropsychology have attempted to build models that can explain the observations made by humans in their drawing-related behaviors. However, these built models have their limitations; for example, they need to reproduce the observations because of specific factors, such as individual drawing styles and the non-reproducibility of bodily motions. In contrast to building models by using top-down approaches, the constructive approach provides another way of investigating complex systems by making models that can replicate behaviors. In case the system includes the human body, cognitive developmental robotics typically uses robots to consider the embodiment factors of the human cognitive systems.

The objective of this study is to understand the aforementioned diversity of drawing representations by constructing computational systems that can replicate a human's abilities of recognition and drawing in a robot. In particular, this study focuses on two abilities: recognition and drawing. The recognition ability involves sharing concepts between hand-drawn pictures (called "sketches") and the visual information corresponding to the object in the real world (called "photos"). The drawing ability involves generating bodily motions to depict sketches from the visual information of the given pictures to be copied (i.e., the depiction target).

These two focused abilities are replicated by functionalities of computational systems. These systems are constructed to include very little prior knowledge to

implement the functions because prior knowledge will lead to strong assumptions when the built system is compared with the human aspects. The recognition system is required to recognize both the photo and sketch images using an integrated image-processing function. The drawing system needs to include visual feedback from what the system draws and generate bodily motions.

Conventional computational systems of picture recognition or drawing have been developed based on pre-designed visual processing and path-planning algorithms, such as edge detection or shape primitives. Recently, large-scale neural networks called "deep learning" models have demonstrated improvements in picture recognition and generation. These models did not require any explicit design for the image feature-extraction algorithms or the shape primitives. The functionalities of recognition and generation were acquired through the non-linear optimization process by using large-scale data.

Even though deep learning models do not require explicit feature designs or shape primitives, they did not satisfy the requirements of the recognition and drawing systems. The recognized image was limited to either photos or sketch images, or they needed to prepare two models for each type of image to construct a classifier for both images. The drawing systems did not include both the visual feedback from the canvas and the bodily motion.

To satisfy these requirements, this study proposes to using deep learning and a robot. The proposed recognition system is built by a convolutional neural network (CNN) to share concepts between the photo and sketch images. Existing CNNs could recognize only either the photo or the sketch images because of the visual-gap between these two types of images and the lack of the large-scale image datasets of sketches. Psychological studies performed on children's drawings have suggested that styles of their hand drawing are influenced by non-photorealistic media in their lives, such as comics or cartoons. Therefore, this study proposes to include non-photo-realistic pictures (called "illustrations") in the training dataset. The inclusion of illustrations also contributes to the enhancement of training datasets because they can be easily crawled on the Web. Through the experiments to classify the sketch and photo images, the efficiency of the proposed data argumentation method was confirmed.

The proposed drawing system consists of a recurrent neural network (RNN) that is known as one of the neural network models that process sequential data. According to developmental psychological and neuropsychological studies, drawing-related cognitive abilities may use integrated visuomotor memory that enables us to use information about the production process from the static image of a picture. In fact, we can associate dynamic information to depict what we see by reusing drawing experiences from the past. In this thesis, RNN is trained to retain the integrated visuomotor memory of the drawing process, which involves a visual transition from a drawn picture to certain bodily motions. The depiction ability is realized by an adaptation of its dynamics to generate an appropriate drawing motion from a static image. In the experiments, the proposed drawing systems demonstrate the adaptation of the acquired memory using a simulator and a robot.

This thesis is organized into six chapters. In Chapter 2, the existing studies are surveyed. First, this chapter introduces psychological and neuropsychological studies conducted on a human's drawing ability using a constructive approach. Then, this chapter describes the computational systems of drawing and picture recognition. Finally, the problems of these introduced studies are explained.

Chapter 3 explains the proposed approach to construct computational systems of recognition and drawing. First, the idea of deep learning models called "End-to-End" is introduced. This idea corresponds to satisfying the requirements to avoid the elaboration of image feature extraction and shape primitives. Then, other approaches to satisfy the requirements of recognition and drawing are explained.

In Chapter 4, the experiments on the classification of the photo and sketch images are explained. These experiments were conducted to confirm that the proposed recognition system could share the visual information of sketches and photos by using a single image-processing system. The system was implemented by using a CNN trained by the novel data argumentation method. This data argumentation method includes illustration images into the dataset. In the experiment, the efficiency of this method was evaluated by a comparison of the classification accuracies obtained from several datasets. As a result, the inclusion of the illustration images improved the classification accuracy. Further, the image

features obtained by the proposed method were analyzed by visualizations.

Chapter 5 gives the details of the experiments conducted to learn the drawing process. These experiments include two phases. The first phase is to check the association ability of the drawing system for learning the drawing process in a simulated environment. The second phase includes experiments that use a humanoid robot. The simulator experiments suggest that this association mechanism also enables RNNs to change the drawing scenario depending on the lines added in advance. Through the experiments using a humanoid robot, the proposed RNN model demonstrated association ability for drawing simple shapes. Also, another experiment on learning distorted shape drawings suggested that the proposed model succeeded in recognizing shapes.

In Chapter 6, the contributions of this study to understand drawing ability are summarized. Finally, this thesis is concluded by describing the direction that future research could take.

Acknowledgements

First of all, I would like to thank my supervisor Prof. Tetsuya Ogata for his patient guidance and the invaluable advice he gave me through discussions over the past five years. When I joined his laboratory as a master's course student, I needed to learn engineering topics from scratch because my graduation thesis was on 17th century Spanish artworks, which required very little use of mathematics. Under Prof. Ogata's expert guidance, I learned not only engineering and intelligent robotic systems, but I also developed a scientific perspective that helped me investigate various world phenomena including human intelligence.

I am deeply grateful to Prof. Takashi Kawai and Prof. Yasuhiro Oikawa of the Department of Intermedia Art and Science, Waseda University for reviewing my thesis. This research project is also supported by the insightful comments of Prof. Shigeo Morishima of the Department of Applied Physics (Waseda University), Prof. Shingo Shimoda of the Institute of Physical and Chemical Research (RIKEN), and Prof. Giulio Sandini of the Italian Institute of Technology (IIT). Prof. Sandini very kindly accepted me as a visiting student in the Italian Institute of Technology from May 2016 to October 2016.

I would like to thank Dr. Kuniaki Noda, Dr. Kuniyuki Takahashi, Ms. Madoka Yamakawa, Ms. Kana Sekiguchi, Mr. Hadi Tjandra, Dr. Yuki Suga, Dr. Hiroaki Arie, Dr. Shingo Murata, Dr. Hiroki Mori, Ms. Naomi Nakata, and other members of the laboratory for Intelligent Dynamics and Representation. For the last five years, each of them has provided me tremendous support for my daily research activities through discussions; they have encouraged and assisted me in carrying out my experiments. Dr. Noda gave me a number of comments for my research plans and offered brilliant technical advice. Dr. Takahashi and

Mr. Tjandra helped with my robot and suggested the system framework given in Chapter 5. Ms. Yamakawa and Ms. Sekiguchi helped me to design the experiments in Chapter 4. Dr. Suga, Dr. Arie, Dr. Murata, and Dr. Mori gave many suggestions on the direction of my research activity. Their suggestions helped me learn how to find problems and solve them. Ms. Nakata supported my accounting work. I extend my heartfelt gratitude to each of these persons.

My research activities were also supported by the members and students of the Cognitive Interaction Lab in IIT. Advice and comments given by Dr. Francesco Rea and Dr. Alessandra Sciutti were a great help in my research activities and my daily life in Italy. My friends in this institute, Mr. Marco Pantella (the housekeeper of my apartment in Genova), and his friends helped me enjoy my stay in Italy.

From 2014, I received financial support from "Embodiment Informatics", which is the Waseda University program for leading graduate Schools supported by the Ministry of Education, Culture, Sports, Science, and Technology, Japan. This program also supported me by providing thought-provoking workshops and lectures. I would like to thank all my friends for the interesting discussions that made me consider my research from different angles.

My interest in the drawing abilities of humans started during my days in the laboratory of Prof. Ken Yabuno from 2011 to 2013. I still remember his powerful words in class: "Painters should be intellectuals." During my days in his laboratory, we discussed the essence of the art by relying on the artistic expressions of films, novels, and painting. These discussions were guided by Prof. Yabuno, Prof. Nobuyuki Takahashi, and Mr. Akihisa Inoue. The motivation for this work is based on these discussions.

Finally, I would like to express my thanks to my family and friends for their ongoing and unconditional support. My family provided me the required financial and emotional support along with all the encouragement that I needed to pursue these investigations. I really appreciate my family because even when I was a youngster, they never went against my wishes in the choice of my career path. In fact, my interest in robotics can be traced back to the toy robot development kit that my parents bought me in childhood. They also allowed me to learn painting

at school when I was ten years old. I knew then that I would pursue painting all my life. I would never have written this thesis if my family did not understand and support me to pursue studies in my field of interest.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Drawing is a universal medium used to pictorially represent concepts. Pictures have been used for more than 30,000 years to share concepts [1]. We can interpret the concept of pictures even if these pictures have been drawn at different times in history or in different countries. Also, we can explain various concepts by sketching.

The motivation for this study is to bring clarity in the ambiguity associated with the visual representations of hand-drawn pictures. Pictures can take many visual variations as long as they can be interpreted according to the drawer's intentions. It is hard to find a visual similarity between a photorealistic image and pictures. Also, the drawing process is different for each drawer. Figure 1.1

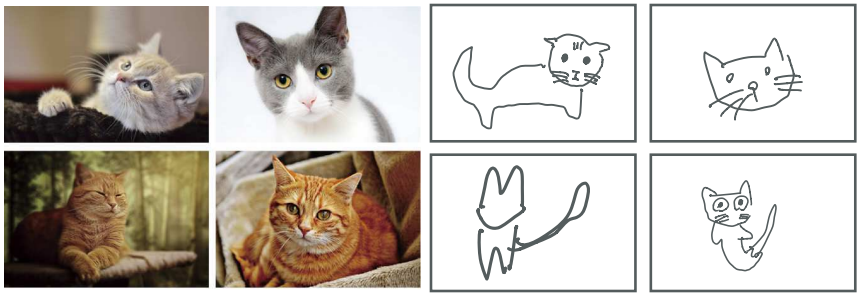


Figure 1.1: Diversity of pictorial representations of cats. (a) Cats in the real world. (b) Hand-drawn pictures of cat.

shows the examples of cats in the real world and the corresponding hand-drawn pictures. Although all these images represent cats, the pictures do not share any visual characteristics with photorealistic images. Further, we can depict what we see by producing motor activities to draw lines on paper or canvas. The produced drawing process has potentially many variations because of the differences in styles, the order of strokes, and our body movements. Even if we repeatedly draw a very simple shape, such as a triangle, the same line will never be reproduced. Therefore, certain questions arise. How can we share the concepts between hand-drawn pictures and the information from our visual perception system? How can we produce a drawing process from our visual information?

A number of studies have investigated the abilities related to interpreting concepts from drawn pictures and producing the drawing process. Many researchers have noticed that the skills to understand or produce pictures are based on the fundamental cognitive skills of visual perception or motor planning. Infants start to draw when they are about one year old [2]. The developmental process of drawing skills has been discussed as the emergence of the ability to symbolize and the emergence of the visual perceptions of space or motor skills [3, 4]. The investigations that explain why we can understand or draw pictures may lead us to approach unsolved problems of human cognition.

1.2 Cognitive Developmental Robotics

Many psychological studies have tried to explore the fundamental principles related to understanding and drawing pictures. These suggested principles can explain the phenomena in experiments of drawing tests. In these studies, picture understanding and drawing abilities are considered as processes that use representations of pictures, symbols, and motions. In this sense, picture understanding is a process that converts the visual representation of a picture into its symbolic information (e.g., category). Depicting what we see is also another process that involves the visual representation of objects, and converts them into lines by using a series of drawing motions via motor planning [5].

Unlike building principles to explain, building computational systems to repli-

cate behaviors provides another way of exploring the underlying principles. This is known as the constructive approach. Investigations of the minimal condition that can realize the phenomenon enables us to understand the mechanism of complex systems. Recently, a new interdisciplinary field named cognitive developmental robotics has been investigating human behaviors related to interactions between our bodies and the environment [6].

The human's drawing-related abilities rely on the body of the subject. The drawing process includes motor planning components that explain the drawing ability [5, 7]. This fact is confirmed by other studies for exploring the fundamental rules of drawn shapes [8, 9] or by replicating the drawing process [10]. The body also affects visual perception. We can imagine how a picture can be drawn [11], and this imagination is used to work to uncover the drawing process in art perception [12].

This thesis reports the investigations of hand-drawn picture representations based on cognitive developmental robotics. In other words, we will try to discuss the fundamental factors that enable a human's drawing-related abilities by constructing computational systems that can replicate the drawing behavior.

1.3 Research Objective and Focus

The objective of this study is to build computational systems that are aimed to replicate the following two drawing abilities:

- Ability 1: Image Recognition. Sharing what is represented by hand-drawn sketches and photorealistic images.
- Ability 2: Picture drawing. Producing motor activities that depict the desired picture similar to the depiction target.

"Image recognition" refers to the sharing of concepts between realistic visual representations of the real world and hand-drawn pictures. We can interpret what is represented by a picture drawn by others and by the realistic visual representation of an object in the real world. Both these interpretations require us to see the representations and recognize them as concepts. This means that we have a

Figure 1.2: Examples of pictures. (a) Study for the Libyan Sibyl by Michelangelo, 1511, Metropolitan Museum of Art, New York. Wikipedia Commons. (b) "Free Curve to the Point - Accompanying Sound of Geometric Curves" by Wassily Kandinski, 1925, Modern and Contemporary Art, New York, Wikipedia Commons. (c) The face of a cat drawn by the author.

visual recognition system that can recognize both types of representations even if it is difficult to find similarities in their visual characteristics. In this thesis, the term "photo" means a realistic visual image that represents anything in the real world. "Sketch" indicates non-professional line drawings to represent some concepts. Figure 1.2 shows a few samples of line drawings. In this study, we have not considered realistic sketches by professionals ("dessin" shown in Figure 1.2 (a)) and abstract paintings whose concepts are rare concrete objects in the real world (e.g., Figure 1.2 (b)).

Another focused ability is "picture drawing." To depict pictures, we need to produce appropriate motor activities. Specifically, we focus on goal-oriented drawing processes to depict another given picture. In this case, the drawer recognizes the depiction target), and produces motor activities of the body that can generate a set of lines that will finally construct the desired picture. As per the model suggested by van Sommers [5], the drawer perceives a visual transition of the drawn picture during the production process so that he or she can flexibly adapt to changing the motions to follow the unexpected changes in the drawn picture, such as the drawing errors.

Developing computational systems that can replicate these two abilities also

contributes to the studies on artificial intelligence. Hand-drawn picture recognition has been regarded as one of the main tasks for image processing systems. In computational systems, the image-recognition ability requires us to consider a function to acquire an invariant image feature between various types of images. Understanding the drawing ability can help us create machines that can draw pictures like artists or can help artists by interacting through the drawing process. Calculating motor programs to control a robotic system to follow the desired path has been the main concern of robotics. However, the method of planning the set of motions from the visual information of the depiction target is still an open problem because of the diversity of stroke orderings and human's bodily motions.

The functionalities of the computational systems should correspond to the aspects of the focused abilities. The requirements of image recognition system are summarized as follows:

- Requirement A1: Less prior knowledge of the images feature detection process
- Requirement A2: Recognizing both the sketch and the photo
- Requirement A3: Sharing the photo and the sketch representation in the recognition process

The requirements of the drawing system are as follows:

- Requirement B1: Less prior knowledge of drawn pictures or the motor planning process
- Requirement B2: Considering image feedback during drawing
- Requirement B3: Considering bodily motions

The requirements A1 and B1 mean assuming less prior knowledge for the implementation of functionalities. In the case of an image recognition system, using prior knowledge corresponds to an elaboration of the image feature-extraction algorithms. It is also possible to consider the shape primitives or the path-planning rules to build drawing robots. However, pre-defined algorithms may lead to strong

assumptions when the built systems are compared with the human cognitive aspects.

The image recognition system has to follow other two requirements. This system is required to recognize both the photo and sketch images (A2). Also, the recognition algorithm is required to be shared among the two types of images (A3). This is because the psychological model of drawing suggests that we use the same process to perceive not only the environment but also the drawn pictures. A drawing system is required to perceive image feedback from the transition of the drawn picture to the reuse of the drawing experiences (B2). Feedback also plays an important role in the drawing process. We adaptively changed the drawing process by looking at what had been drawn in advance. For example, when a circle was given in advance, we added a few smaller circles to depict a face. Further, the drawing process involves generating bodily motions (B3). This requirement indicates that the system does not directly control the pen, but it controls an embodiment system that causes the drawing process.

1.4 Problems of Existing Computational Systems

Conventional image recognition systems have been proposed in studies of artificial intelligence. Researchers have typically used well-designed algorithms to extract image features, such as edge detection or shape primitives. The extracted features were input to a classifier to output a category of the image. The recent success of large-scale neural network models called "deep learning" models have changed this strategy to construct image classifiers. deep learning models do not require any elaborations of the image feature-extraction algorithm, but they obtain the recognition functionality through an optimization process. However, classification targets of these deep learning models are limited to photorealistic images or sketch images, or they need to separately train models for each type of image and merge them into a single classifier.

Conventional computational systems of drawing also require pre-designed algorithms to extract the image features or define simple shapes that will be used as units to represent complex shapes. Drawing systems with fewer elaborations

have been investigated in the fields of cognitive developmental robotics or machine learning for artificial intelligence. However, they also needed to assume the primitives of shapes or the exploration of the drawing motion that will lead to nonhuman-like motion planning. Another study using the deep learning model did not consider any shape primitives, but they did not satisfy the requirements B2 and B3.

1.5 Overview of Approach

The approach of the proposed study is to introduce visuomotor adaptation to build computational systems for image recognition and picture drawing. We propose computational systems that can acquire knowledge for recognition and drawing not by designing rules of image feature extraction or shape primitives, but by using learning samples. Adaptation means that the systems change their behavior to recognize image inputs or generate the motor activities of a robot by reusing the knowledge obtained in the learning process.

For the machine-learning framework, we use deep learning models. One important characteristic of the deep learning model is known as "End-to-End." This means that there is no explicit definition of image feature extraction or motor planning (the requirements A1 and B1). Instead of the designer implementing the algorithms, the deep learning model obtains functionality through optimization methods without any constraints of shape primitives or image feature-extraction algorithms.

The proposed picture recognition system is designed to adapt its classification experiences to both sketches and photo images. This system is implemented by using a convolutional neural network (CNN), which is one of the deep learning models for image recognition tasks. The requirements A2 and A3 are achieved by a CNN classifier for both the photo and sketch images. The classification target of CNN was limited to either photos or sketch images because of two reasons. Firstly, sketch images are very visually different from photo images even if they share the same concept. Secondly, there is a lack of large-scale sketch image datasets. deep learning models typically require a number of images to be generalized for various

inputs. To enable the CNN to classify both the photos and sketch images, we include non-photorealistic pictures by professionals (i.e., "illustrations"). This method reflects the effect of picture representations given as comics or animations in the developmental process of picture recognition.

The picture drawing systems reuse their visuomotor memory to produce the robot's motor activities that depict the desired picture image. Humans can associate motor activities to produce the shown pictures even if they did not draw these pictures. Psychological studies have suggested that this association ability is based on the integrated memory of the drawing body motion and the visual information, including feedback from the drawn pictures. Therefore, we propose to build computational systems that can self-organize visuomotor experiences through learning. The association ability of the proposed systems corresponds to generate goal-oriented drawing behavior. The goal of the association is given as a depiction target image or as a sequence of images that are part of the drawing process. Through experiments on a simulator environment, the proposed system demonstrates that the system can change its behavior to depict a depiction target image even when the intermediate steps are given by an experimenter. For example, the system can add a few lines when the experimenter draws the other lines in advance. Furthermore, we demonstrate that the proposed approach can be applied in case a robot draws pictures. The proposed systems are implemented by recurrent neural network (RNN) models. The proposed RNN models are trained to generate visuomotor sequences of drawings from the initial hidden state. The learned sequence consists of image feedback from the drawn picture and the motion data (requirement B2). The association is implemented by an exploration process of the RNN's feature that will decide the produced drawing behavior. The last requirement (i.e., B3) means that the robotic system needs to be used as a body. The proposed RNN model does not produce a sequence of pen position but produces joint angles of a humanoid robot.

1.6 Thesis Organization

This thesis is organized into six chapters as described in Figure 1.3.

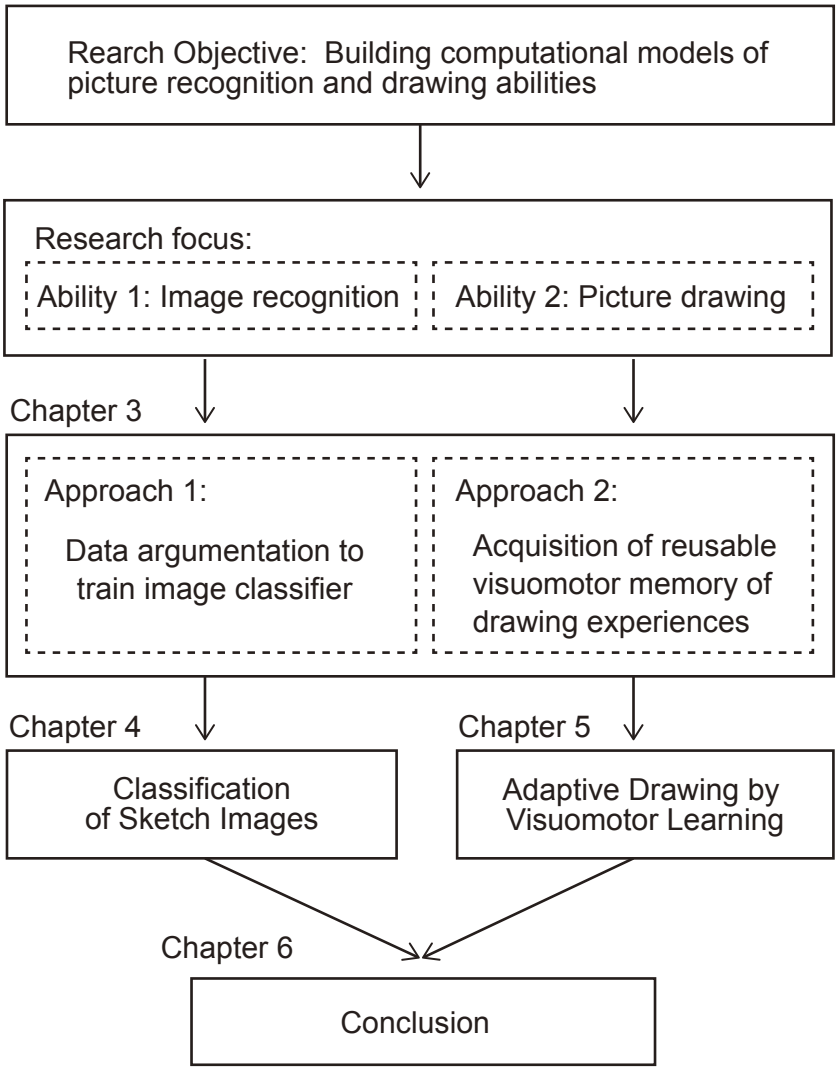


Figure 1.3: Thesis organization

In Chapter 2, existing studies related to this thesis are reviewed. First, the history of psychological and cognitive neuroscience that intends to understand the human's cognitive aspects of drawing are summarized. Subsequently, existing studies that tried to understand the drawing ability by constructive approaches are introduced. Existing computational models for image recognition and drawing generation are reviewed. This section also describes the recent deep learning models not for producing drawing motions, but for directly generating images. Finally, the problems of these introduced computational systems to satisfy the above-mentioned requirements are also explained.

Chapter 3 describes the approaches to build the computational systems that can satisfy the requirements. First, the End-to-End learning model using deep learning is introduced. Then, the approaches to constructing the recognition and drawing systems are explained.

In Chapter 4, we introduce the experiments on classifying the sketch and photo images by using the proposed image recognition system. The task of the experiments is designed to confirm that the inclusion of illustration images enhances the system's classification accuracy. Also, the image features obtained by the trained CNN model are visualized. These visualizations explain how the trained CNN model organizes each type of image for discrimination.

In Chapter 5, several experiments on learning the visuomotor sequences of a drawing are described. First, simulator experiments were conducted to confirm the functionality of the proposed system to memorize visuomotor experiences and associate the drawing motion from a depiction target image or from the image sequence to be copied. Also, this chapter discusses the association ability when using a real robot. In the robot experiments, the proposed system was required to associate a robot's drawing motion of simple shapes. In the final experiment, we demonstrate the recognition ability for distorted shapes by reusing memory.

In Chapter 6, the contributions of this study to understand the drawing ability are summarized. Finally, this thesis concluded by describing the direction of future studies.

Chapter 2

Literature Review

This chapter provides a review of the studies for understanding and replicating the drawing ability of human beings. First, cognitive scientific studies are introduced. Second, the constructive approach to understanding drawing ability is explained. Then, recent computational systems for hand-drawn picture image recognition and generation systems are reviewed. Finally, the problems of the existing computational systems to satisfy the requirements of the proposed system are explained.

2.1 Understanding Drawing Ability

Until the 20th century, there were very limited studies to investigate the mechanism that enabled humans to understand and draw pictures. Rather than understanding the mechanism, the methodology for drawing pictures was documented as the memo for painters. Leonardo da Vinci, who was an Italian Renaissance polymath, recorded observations of human emotional expressions, the structure of the human body, and the depiction of objects for portraits in memo [13].

Through studies of sensory responses to stimuli by esthesiophysiology in the 19th-century [14], Gestalt theory tried to describe the qualities of wholes by introducing self-governing laws. This theory inspired studies on ecological optics by Gibson [15]. He investigated not only the mechanisms of visual perception systems for photorealistic images but also the developmental process of children's drawing [16]. Gibson's manner to propose the principles of visual perception affected the

art theory by Gombrich [17].

Psychological studies in the 20th century have investigated the developmental process of a child's drawings [2, 18, 3]. These studies described the developmental process of children's drawings, especially scribbling because unsophisticated motor skills gradually acquired realism by the introduction of the shape primitives' semantic knowledge of objects. A theoretical approach to the adult drawing process was discussed by Cohen et al. [19]. They explained the factors that led to the drawing of inaccuracies when we tried to depict what we saw.

Mechanisms of visual perceptions or drawings have been mainly proposed by the studies on cognitive neuroscience. Researchers have tried to build model cognitive mechanisms that can explain the phenomena caused by drawing disorders [5, 20]. Drawing tests are used for clinical psychology researchers to measure intelligence [21] or constructive apraxia that is a disability of synthetic activities [22]. In the studies on constructive apraxia, drawing was considered as a process that involved the construction of motor activities by building the spatial structure of the depiction target and considering the description of the structure through semantic systems concerned with word-related representations [7]. The model is assumed to be a complex system built by the components assigned to each neuroanatomical regions. The visual perception process for artworks is discussed in neuroesthetic studies [23, 24, 25]. They suggested that the sense of beauty is realized by the organization of neural activities.

Cognitive psychological drawing models follow the study by van Sommers [5] described in Figure. 2.1. He tried to build models based on Marr's framework [26] for deriving the shape information of objects from the image shown in Figure 2.1.

- Primal Sketch (2D): Structure of the intensity value at each point in the image (edge, blob, bras, etc...)
- 2 1/2-D Sketch: Orientation and rough depth of the surfaces and contours (surface orientation, depth, distance from viewer, etc.)
- 3-D model representation: Shapes and spatial organization of 3-D shape primitives

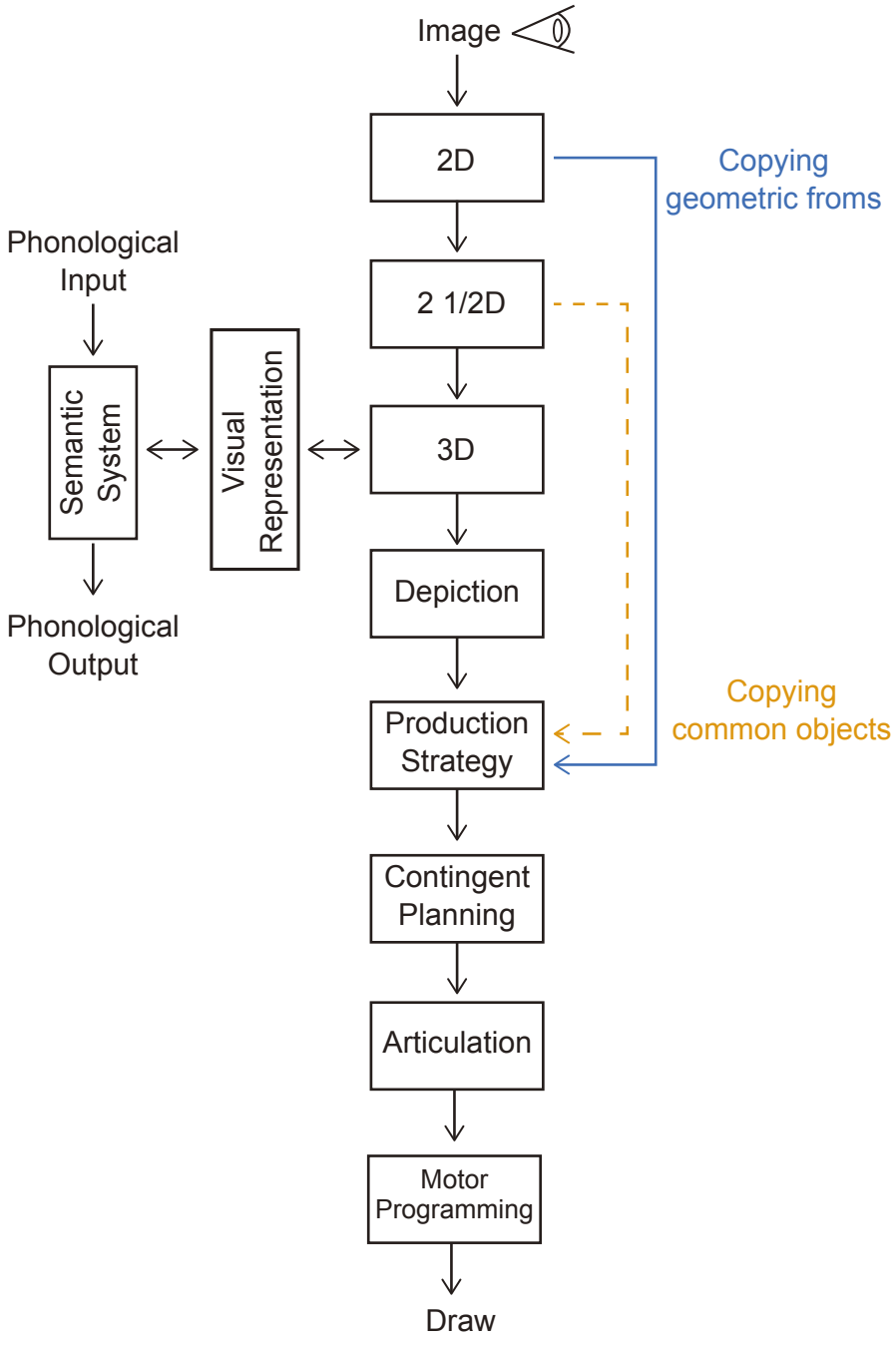


Figure 2.1: Drawing model by van Sommers. Adapted from Fig. 22 in [5].

Van Sommers added a drawing production component into modules by using Marr's framework. The production components are summarized as follows:

- Depiction: Higher-order decisions (types of objects, viewpoints, the levels of detail)
- Production Strategy: Chunking parts of figures (composition of lines)
- Contingent Planning: Motor planning, such as the ordering of sequences
- Articulation: Starting position, stroke direction, order, circle schematics, paper contact, geometric grouping, anchoring, and routing planning
- Motor Programming: Motor movements

Van Sommers also considered pathways from the semantic systems to a visual representation of the object. His model allows us to consider pathways from each visual component by using Marr's framework for the drawing production components. Thus, there are a few variations depending on the drawing scenario. The model most related to the drawing model proposed in this thesis is "copying geometric forms." In this case, the depiction target (i.e., the picture to be copied) was not 3D objects but simple shapes such as a circle, a square, a triangle, and a combination of a few of these shapes.

Besides the image of the depiction target, the drawn picture image must be used in the production process. The model proposed by van Sommers did not clearly mention (but he did suggest) the existence of a temporary store for later production; this store was thought to be a memory of the visual representation (mental imagery) [27].

2.2 Constructive Approach for Drawing Behavior

Besides building concrete models based on the components that can explain the phenomenon of drawing tests or neural activities, building models to replicate the

abilities provides another way to explore the underlying principles. By investigating the minimal conditions that can realize this phenomenon, we can understand the models better. One example related to the drawing ability is the "Power Law," which explains the relationship between the movement speed and the curvature of simple line drawings, as proposed by Lacquaniti et al. [8]. This rule has influenced other studies of human locomotion [28].

The power law is very simple and makes it possible to easily analyze mathematical characteristics by simulations. Drawing ability consists of a wide range of cognitive functions; therefore, this rule needs to be more complex when the model is designed. In fact, the more general principle of human writing makes us consider not only the points of lines but also the motion of the drawer's arm. Wada et al. proposed the minimum jerk principle-based model that could represent handwritten characters [9]. This study suggested that the characteristics of the body effect the drawing ability. The embodiment effects in the drawing are also mentioned in the researches of the developmental process of drawing [3]. The model proposed by Wada et al. is limited to continuous motions and trajectory planning; therefore, this model does not consider the construction process to acquire the idea of shapes; this model also does not consider how the drawer interactively revises the trajectory depending on what is drawn.

2.3 Computational Systems of Drawing

Recent developments in the computational theory has allowed us to calculate the image and the sequential motion data; therefore, it has become possible to build computation systems to recognize or generate images. Picture recognition systems have been developed as sketch interface systems [29, 30, 31]. These systems were intended to accept a human drawing as the user's command to the software [32]. In this case, a sketch is given as sequences of two-dimensional points. These points are abstracted to the structure of 3D primitive shapes [33, 34] or directly used for classification [35, 36, 37] and the beautification of lines [38]. In case the picture is given as raster images, the features to describe images will be typically extracted by the edge detector [39, 40]. The extracted features become inputs to

the recognizer part.

Computational systems of picture production have been investigated as applications of computational arts or robotics challenges. In the 1980s, Harold Cohen developed a system to generate artistic images by using a computer program called ARRON [41]. In its early stages, AARON was developed as a computer program to produce images. Subsequently, Cohen built printing machines to produce pictures on a canvas. Currently, there are two main methods to produce pictures. The first method is to directly generate the image data. This method has been investigated by studies of non-photorealistic rendering [42, 43]. Typically, these studies have proposed new algorithms to convert the pixels of an input image into the desired artistic images. In contrast, to calculate the pixels of the output, other studies on artistic robots have tried to implement systems that can control robotic plotters to obtain pictures [44, 45, 46]. These systems typically extract the edges of a given depiction target through a robot's camera or 3D sensors. Then, the extracted shape information is converted into a trajectory a brush or a pen controlled by the robot's hand. The main problem with this trajectory generation was the generation of accurate motions to effectively handle contacts between the tools and the canvas [47, 48]. Image feedback during the production process was regarded as an input to update the generated trajectory. Some of the existing systems updated the planned trajectory to consider unexpected changes of the picture because of complex physical interactions of the contacts that were difficult to simulate in advance.

Developmental systems for replicating drawing abilities were studied in the field of cognitive developmental robotics [6] or machine learning for artificial intelligence. Mohan et al. proposed a model that was trained to reproduce given lines as a combination of the assumed primitives of the robot's motions [49]. Their method employed primitives for handwritten shapes based on the catastrophe theory [50]. However, the designing primitives have a strong association with psychological studies. In fact, it is difficult to show evidence that a human's drawing process follows these assumed primitives. Unsupervised training methods do not assume any concrete primitives of shapes [51], but the obtained drawing order is not shared with humans.

2.3.1 Deep learning Models for Drawing

The idea of neural networks originated from a linear model for neural activities [52]. Minsky showed that the linear model could not be applied to data that is not linearly separable [53]. This limitation was overcome by introducing a hidden layer whose parameters were acquired by back propagation [54]. Neural networks are allowed to have an arbitrary number of hidden layers to maintain nonlinearity that increases the complexity of the acquired function. However, the size of networks is limited by the capacity of the calculation process because the backpropagation process requires computational memory as large as the size of the networks and larger than the size of the dataset.

Recent developments in computation capacity and the appearance of rich and large datasets have allowed machine-learning researchers to test larger-scale networks. In 2012, Krizhevsky et al. won an image classification challenge using a CNN [55] that had 60 million parameters [56]. Networks with many hidden layers came to be called "deep learning" because these layers developed a deep structure because of the stacking of layers [57]. The success of deep neural networks (DNNs) is based not only on the evolution of calculators but also on the introduction of techniques to avoid the optimization problems of over-fitting and gradient vanishing, such as batch normalization [58], dropout [59], refinement of activation function [60], and residual connection [61]. The details of deep learning models related to this thesis are provided in Appendix A.

2.3.2 Sketch Recognition Models

In the context of deep learning, sketch recognition tasks are often achieved by CNN models. Yu et al. proposed a multi-scale network to classify sketch images by considering the ordering of the drawing process [62]. They tuned parameters of the CNN model proposed by Kerizhevsky et al. and prepared five parameters to build a huge ensemble network. Image retrieving tasks have also been conducted using CNN. Sketch image recognition considering the drawing process was also studied by Sarvadevabhatla et al. [63]. Seddati et al. proposed a CNN-based framework to find sketch images similar to the user's sketch image input [64].

Retrieving photos from sketch images is a more challenging task because of the large visual gap between these images and photos-a photo image has color and details that are invariant in the same object. However, sketch images do not have any color and have less textural information. Sangkloy et al. tried to concatenate two CNN models that were independently trained to classify each type of image [65].

2.3.3 Picture Image Generation models

The first image generation by a DNN model is called "autoencoder." It encodes input images into low-dimensional image features and decodes them to the input image [57]. An autoencoder is constructed by stacking many hidden layers of a feed-forward neural network. The hidden layers are replaced by convolutional layers to improve the efficiency of image generation by sparse connectivity [66]. The CNN model has recently started being used not only for autoencoding images but also for sketch image generation from photo images [67] or by the simplification of rough sketches [68].

The above-mentioned image generation models are trained by loss functions defined by a norm of the difference between the generation and the target image. In this case, the generated image tends to have a blurry expression [69]. A solution to overcome this problem is to avoid calculating the loss value with respect to the output. Gatys et al. proposed a method to replace the texture representations of photorealistic images by using another artistic image [70]. Their approach did not calculate the losses at each pixel but at the intermediate layers of the pre-trained classification CNN model. Another way to generate high-quality images by CNN models is adversarial training protocol proposed by Goodfellow et al. [71]. This protocol requires two CNN models called "generator" and "discriminator." The generator network tries to produce images that can fake the discriminator network that adversely penetrates the fake. Radford et al. introduced CNN into an adversarial network to create a large-scale CNN model for image generation called "deep convolutional generative adversarial network" (DCGAN) [72]. DCGAN is applied to a variety of image generation tasks including sketch image generation [73].

2.3.4 Drawing Motion Generation Models

Time series data, such as drawing motion, is efficiently handled by RNNs [74]). The RNN model has feedback in its hidden layer beyond the time step; therefore, the model retains the sequential memory. One of the problems of RNNs is that they easily lose memory because the information vanishes by the propagation process through feedback connections. Hochreiter et al. proposed gates for the internal state of neurons to avoid memory loss [75]. Their model is called "long short-term memory" (LSTM). LSTM shows good ability to learn complex time-series data for language processing [76], image generation from captions [77], and continuous sequence generation for a reinforcement agent [78].

The RNN-based model for drawing motion generation was studied by Ha et al. [79]. They proposed an autoencoder model whose encoding and decoding parts were implemented by LSTM. The model was trained to accept a sequence of a pen's position and status, which specified whether the pen touched the paper or whether the drawing process had ended. The encoded feature of the input drawing sequence was reconstructed by using another decoder LSTM. Researchers have also proposed to add a loss to force the encoded feature to form a Gaussian distribution [80]. After training the model with thousands of human drawing sequences, the model showed its generalization ability by producing new samples or by completing a given drawing process.

RNNs have also been used for drawing robots. Mochizuki et al. proposed a RNN-based model that could improve a robot's drawing ability through interactions between humans [81, 82]. Their model was inspired by the developmental process that improved the drawing skills of children. In their experiments, a humanoid agent randomly moved the arm as the researchers observed the position of the pen. Then, they conducted an incremental imitation learning for drawing simple shapes.

2.4 Problems of Existing Computational Systems of Drawing

2.4.1 Primitive Design

As mentioned in the last chapter, the methodology of algorithm design plays an important role when a computational system is built to replicate the human's cognitive systems. Most of the existing computational systems are constructed by subcomponents that require the elaboration of algorithms or explicit representation design. Hand-drawn picture recognition systems have required edge detectors [29, 30, 31] or well-defined shapes primitives [35, 36, 37]. Also, most drawing generation systems have assumed that the drawn shapes consist of primitives or are required to detect the edge detection subsystems. The existing robotic systems use edge detectors to determine the path of a pen [47]. Mohan et al. did not conduct any explicit shape detection, but they assumed that the drawn shapes consisted of primitives [49].

The assumption of concretely shaped primitives makes the consequent discussions difficult for comparison with the human's cognitive aspects. We can design image classifiers by using the pre-designed edge detectors or shape primitives. But these pre-designed algorithmic primitives requires us to assume the existence of the primitives in the human cognitive systems. Also, it is challenging to assume that we use concrete primitive features to depict abstract visual perceptions. The drawing style is acquired during the developmental process in our childhood; therefore, the style can have variations according to the individual factors of the drawers, such as the cultural or development characteristics of drawers [83].

2.4.2 Picture Recognition Systems

CNNs are used to build deep learning models that can recognize both the sketch and photo images. As reviewed in the last chapter, the recognition targets of the existing CNN models are limited to either photo images or sketch images. It is challenging to train a CNN recognizer with multiple types of image. Sangkloy et al. proposed an integrated CNN model that consisted of two sub-CNN models

to classify either photo or sketch images [65]. However, there were two image feature extraction processes according to the image types. The difficulty comes from the visual difference between the sketch and photo images. Furthermore, the existing sketch image datasets had very few samples to substantially train large-scale CNN models. There is an open image database set that has 20,000 pairs of sketch images and photo images [31], but this size is still much less than other photo image datasets [84, 85].

2.4.3 Drawing Systems

The recent success of deep learning has also contributed to the development of sketch generation systems without any explicit shape primitive designs. However, researchers involved in these studies did not consider the visual feedback from both the drawn picture image and the bodily motion. The RNN model proposed by Ha et al. did not consider the drawn picture image; therefore, the model does not accept image input as the depiction target [79]. Also, they did not consider bodily motion in the generation process of the drawing. The lack of bodily motion means that the system ignores the effect of embodiment in the drawing process. Mochizuki et al. conducted drawing experiments using a humanoid robot [81, 82], but their system did not accept any image feedback.

Chapter 3

Approach and Methodology

To satisfy the requirements mentioned in Section 1, this study uses the deep learning and a robot. The requirements for the sketches were as follows:

- Requirement A1: Less prior knowledge of the image-feature detection process
- Requirement A2: Recognizing both the sketch and the photo
- Requirement A3: Sharing the photo and the sketch representation in the recognition process

The requirements for the drawings were as follows:

- Requirement B1: Less prior knowledge of drawn pictures or the motor planning process
- Requirement B2: Considering image feedback during drawing
- Requirement B3: Considering body motion

Firstly, the approach to the avoidance of elaborations (A1 and B1) is achieved by introducing deep learning models. The deep learning model is known to directly perceive or generate large dimensional, sequential data without any prior knowledge of them. Two requirements for the image recognition system (A2 and A3) are satisfied by building a DNN model that can recognize both the sketch images and the photo images. The requirements (B2 and B3) are achieved by the adaptation of an integrated memory of the image and motion sequence. This adaptation process is realized by RNN models.

3.1 End-to-End Method in Deep Learning

End-to-End is a deep learning model because the model's algorithm is obtained by an optimization whose objective function requires a relationship between the input and output, which specify the ends of the model's structure. End-to-end models have been limited to small-dimensional data [55]. However, the recent developments in the computation capacity and large-scale datasets (i.e., big data) enable us to build large-scale neural network models by composing many non-linear components to directly calculate high-resolution images or movies. This thesis proposes to use End-to-End learning to implement image recognition and drawing functions. In particular, the image recognition model is implemented by a CNN. In contrast to the conventional feature extractors for images, CNN acquires good parameters of convolutional kernels and linear connections by solving the classification problem via gradient descent. The drawing model is required for considering the sequential data of the drawing motion and the visual feedback from the drawn picture. In this case, the model is implemented by using RNN. The model is trained to generate visuomotor sequences also by gradient descent.

3.2 Data Argumentation to Train Image Classifier

Unlike deep learning models, human beings do not have any difficulty in sharing the concepts of sketch and photo images. Psychological studies of children's drawings have suggested that hand drawing styles are influenced by non-photorealistic media in their life, such as comics or cartoons [86, 87]. In this thesis, the image recognition CNN model is trained not only for sketch and photo images but also for another type of image called *illustration images*. Illustration images are non-photorealistic color pictures as shown in Figure 3.1 (b). Regarding the optimization process of CNN, illustration images also contribute to the argumentation of the sketch dataset because they can be easily crawled on the Web. The proposed data argumentation method also includes the color transformed version of the photo and sketch images, rather than the conventional data argumentation

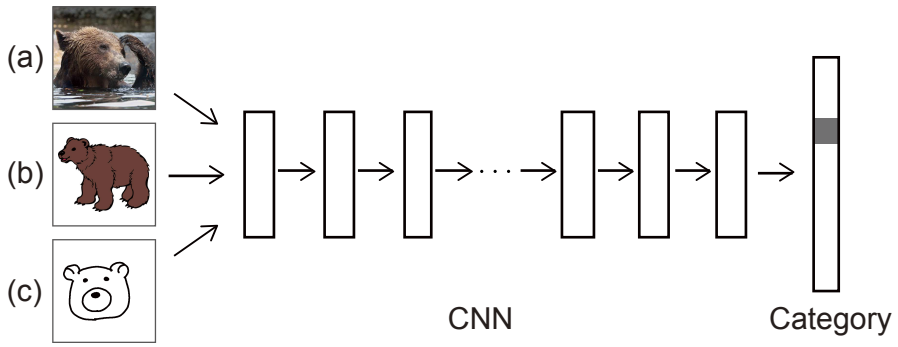


Figure 3.1: An overview of the proposed approach to construct a computational system that replicates the recognition ability for sketch and photo images. (a) Photorealistic images. (b) Professional and colored sketch images (illustration images). (c) Hand-drawn sketch images.

technique that typically increases the data size by the affine transformation of images.

3.3 Acquisition of Reusable Visuomotor Memory of Drawing Experiences

To satisfy the requirements of the picture drawing systems B2 and B3, we introduce the RNN learner to obtain a reusable memory of the drawing. The human's ability to draw from a picture by associating motor activities was indicated by Freyd et al. [88, 11]. Their studies showed that we could use production information from a static form such as a picture. The information about a letter's production was also used for recognition [89]. Pignocchi suggested that the ability to acquire this information was key to interpreting artworks [12]. Waterman et al. suggested that the acquisition of information on how to produce pictures from an image is based on a memory of the visuomotor experiences of drawings or letterings [90]. In this study, we propose RNN-based systems that obtain visuomotor memory consisting of visual transitions of the drawn picture as image sequences and motions.

Figure 3.2 describes the proposed approach to build the computational system to obtain memory. In this study, we assume that "association" ability means

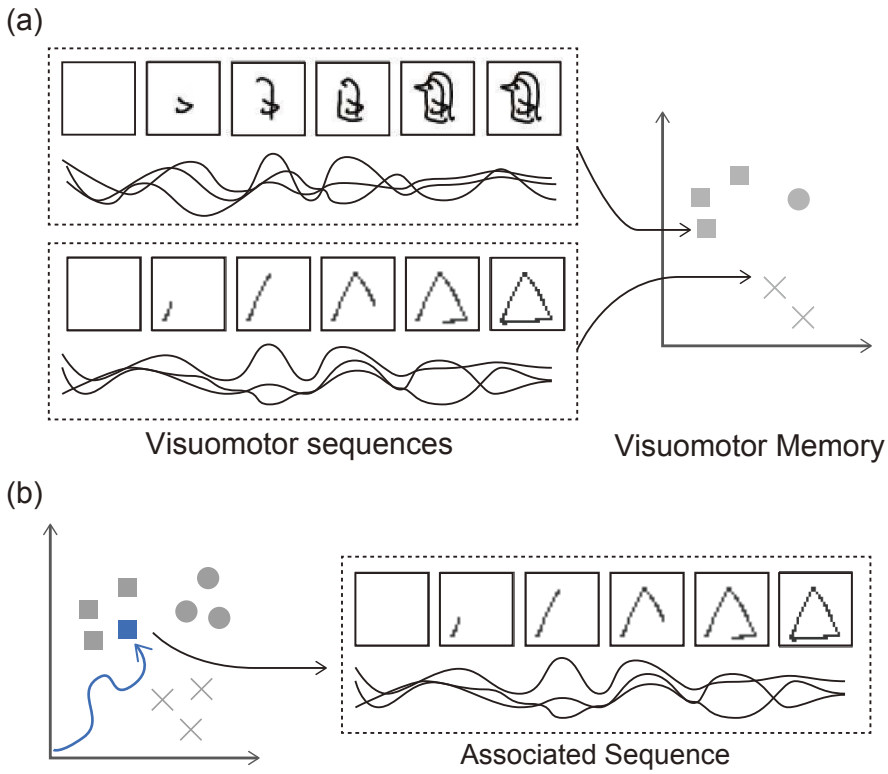


Figure 3.2: Visuomotor adaptation for drawing. (a) Training process to acquire an integrated memory of the drawn picture image and motion. (b) Adaptation of the memory for associating new drawing sequences.

to adapt the obtained visuomotor memory of drawing to generate appropriate another drawing motion to depict a given picture image to be reproduced. To enable the association ability, the proposed drawing system obtained a memory that could be used for adaptation. Adaptation means to interactively change the system's state to produce motor activities by using the visual feedback and the depicted target. First, the RNN model was trained to generate visuomotor sequences from a learnable space. Each sequence consisted of drawing motions, such as a sequence of the joint angles of a robot and the corresponding visual feedback from a drawn picture. Image feedback corresponds to the visual transition of a canvas or paper. After the training process, we conducted an exploration of a point in the obtained RNN space. This exploration process tried to minimize the difference between the drawn picture image and the depiction target. Finally, the RNN generated the drawing motion by perceiving the image feedback from the currently drawn picture. Further, the implementation details of RNN are described in Chapter 5.

Chapter 4

Classification of Sketch and Photo Images

4.1 Introduction

This chapter presents experiments to check the efficiency of the data argumentation method for enabling a system to recognize a sketch and a photo. Experiments were performed to evaluate the contributions made to the recognizer by the pictures drawn by professionals (i.e., illustrations). In the experiments, several variations of image recognition models using deep learning were evaluated by the classification tasks on the photo and sketch images. The classifiers were implemented by a CNN, which did not require pre-designed shape primitives or feature extraction algorithm. The datasets for evaluation consisted of hand-drawn sketch drawings and photo images.

The rest of this chapter is organized as follows. In Section 4.2, the proposed data argumentation method is explained. This method is intended to improve the discrimination accuracy of a CNN model for both the sketch and photo images. Then, the proposed method is evaluated in experiments by using a CNN model. Section 4.3 introduces experiments to classify 20 classes of animal images. We describe the results of the experiments in Section 4.5. Also, we report an analysis of the image features obtained by the CNN model trained by the proposed method and discuss the possibility of the proposed method being used for sketch image retrieval. Finally, this chapter concludes in Section 4.6.

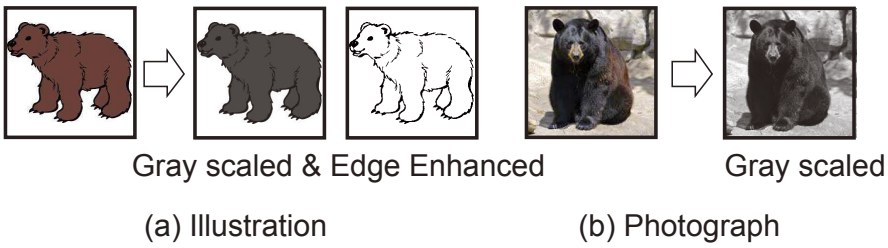


Figure 4.1: The proposed method to prepare training dataset for CNN. (a) Illustration images with grayscale and sketch-like variations of images. (b) Photo images with grayscale variations.

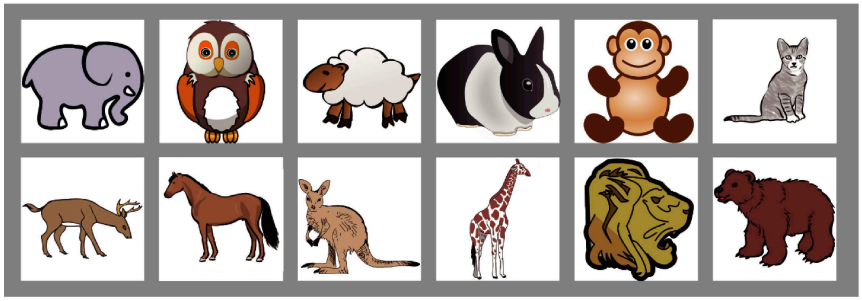
4.2 Data Argumentation Method

Figure 4.1 shows how to prepare a training dataset for CNN. We include illustration images into the set of photo and sketch images. Illustration images are expected to work as intermediate visual representations between photos and sketches because they can be easily collected by image search engines. The added illustration images included color-transformed versions. We included edge-emphasized versions of the illustration images to imitate the sketch images. The edges of the illustration image were extracted by using a canny detector [39]. Furthermore, the dataset had grayscale images of the photo. The sketch images were drawn by five Japanese participants.

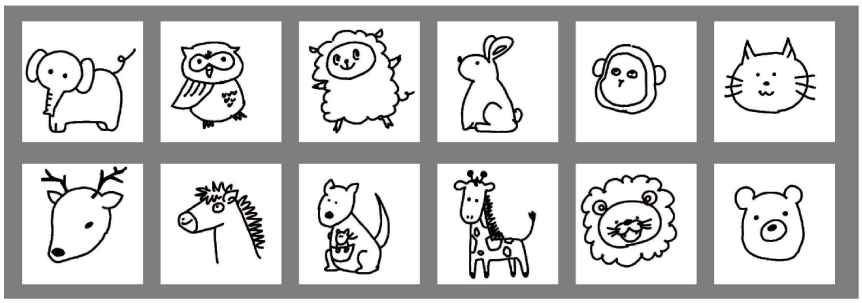
4.3 Image Classification Experiments

To confirm the proposed data argumentation method, experiments were conducted for classifying 20 classes of animal images. In these experiments, four CNN models were evaluated by using four training datasets. To prepare these training datasets, the source of image types was changed. The discrimination abilities of the trained CNNs were evaluated by classification accuracies for the untrained images of photos and sketches.

(a) Examples of photo images



(b) Examples of illustration images



(c) Examples of sketch images

Figure 4.2: Examples of the collected images for experiments. (a) Photorealistic images. (b) Illustrations. (c) Sketch images drawn by humans.

4.3.1 Details of Datasets

Figure 4.2 depicts a part of the image samples used in the experiments. The database was created by collecting 27,927 images of animals. These images belonged to 20 classes of animal names and were collected by an image search engine. To obtain sketch images for testing, we asked five participants to draw 100 sketches. To input to CNN models, we resized the collected images into color images of 256 256 pixels. When these images were used for training CNN models, they were randomly cropped into 227 227 pixels. Note that all the images were horizontally flipped to increase the number of images in each dataset.

Table 4.1: Training datasets. (A) Illustration-only dataset. (B) Photo-only dataset. (C) Illustration and photo dataset. (D) Dataset using the proposed method.

	(A)	(B)	(C)	(D)
Photo	-	-	x	x
Illustrations	x	x	x	x
Photo (Gray)	-	-	-	x
Illustrations (Gray)	-	-	-	x
Illustrations (Edge)	-	-	-	x
Num. of Image	12,734	38,468	51,202	115,138

Table 4.3.1 describes the detail of the four datasets for the experiments. The first two datasets A and B have only one type of image from the illustration or photo. Also, the dataset C has both types of images. We prepared the dataset (D) by using the proposed method. As a test dataset used to evaluate the classification accuracy of untrained images, we also made a dataset that had 100 images of sketches and photos.

4.3.2 Training CNNs

We use Alex-net as a deep learning model for the experiments. The architecture of this model is described in Figure 4.3. This model has five convolutional layers and corresponding activation functions. The input images are convoluted by these layers and down-sampled by pooling layers. Finally, the image features are

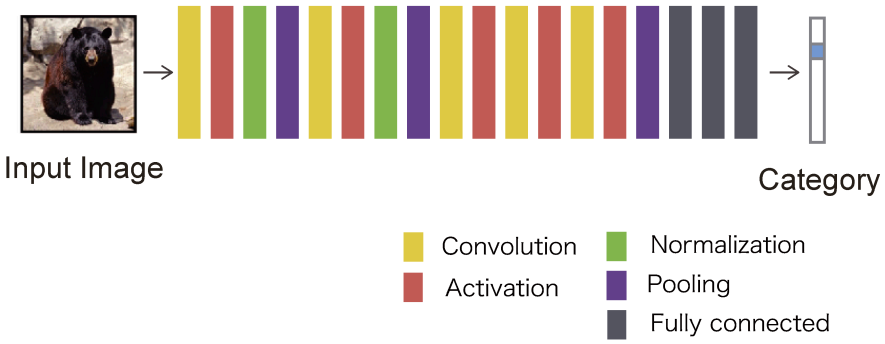


Figure 4.3: Architecture of CNN for experiments

converted to the probability of class by the three fully-connected layers.

The learnable parameters of the model are obtained by a stochastic gradient descent [91]. The gradients are given by the back propagation algorithm [92], which is a method for calculating the derivatives of the loss function with respect to the vectors of the composed, smooth function. The loss function of the model has cross entropy between the predicted probability of the category and one-hot representation of class information:

$$L = -\sum_{b} \hat{y}_b \log y_b. \tag{4.1}$$

Here, $\hat{y}_b$ is the category probability of b-th sample, and y is the probability predicted from the corresponding input. At each iteration, a number of image samples are randomly picked up from the training dataset. Then, the loss value is calculated based on Equation 4.1. Finally, the parameters are updated by using the derived gradients.

All the models were trained for 1400k iterations by gradient descent. At each iteration, the gradients of the learnable parameters of the models were obtained by using a mini batch having 100 images. We use Caffe [93] as a framework to implement the optimization process.

4.4 Results of Experiments and Discussion

4.4.1 Discrimination Ability for Photos and Sketches

Table 4.2: Classification accuracy on the test dataset. Each line corresponds to the best score obtained during the training by using one of the datasets.

Dataset	Photo	Sketch	Mixed
(A) Illustrations	26%	41%	33%
(B) Photo	99%	11%	55%
(C) Illustrations and Photo	99%	42%	71%
(D) **Proposed Method**	99%	76%	**85%**

The training results of the four CNN models are summarized in Table 4.4.1. Each line of the table indicates the best classification accuracy during the training iterations. These accuracies are calculated by using the test dataset that consists of photo and sketch samples. In case a model is trained using the dataset (A), the model classified 33% of the test dataset images, and the sketch images were discriminated more successfully than the photo images. When another CNN model was trained with only the photo images in the dataset (B), the model discriminated among almost all the test photo images, but the accuracy of the sketch images became worse than the accuracy of the dataset (A). Classifying both the illustration and the photo improved the accuracy of the sketched images by maintaining high accuracy for the photo images. Finally, the dataset (D) made by the proposed method outperformed the other datasets. By adding color-transformed illustrations and photo images, the accuracy of sketched images was improved.

4.4.2 Acquired Image Features of CNN Models

Also, we analyzed the image features of the trained CNN's layers by visualizations using principal component analysis (PCA). PCA was performed on the output of the fully connected layer, which was second from the output layer when the images of the training dataset were given as input to the CNN model. Figure 4.4 shows how the features were gathered concerning the category. The contribution values were 1.84%, 1.06%, and 0.97%for the first, second, and third layers, respectively.

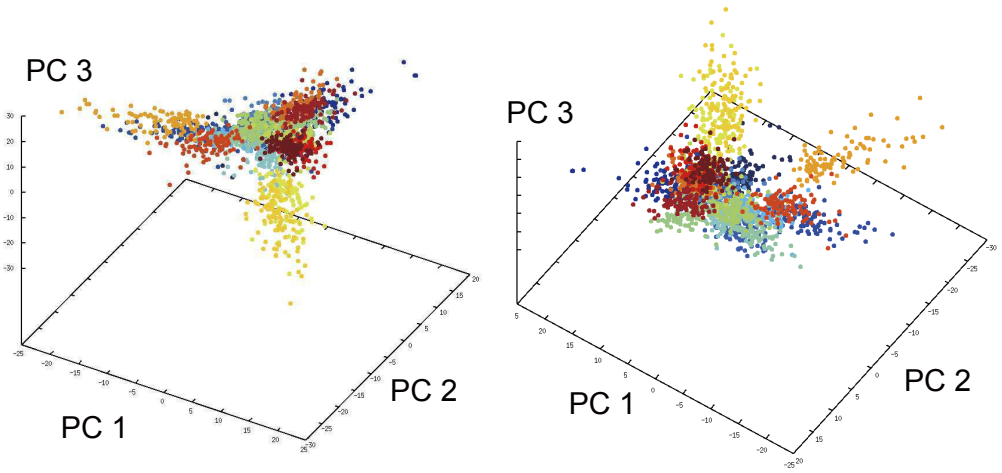


Figure 4.4: Visualization of the image features colored by the category. The image features were obtained in one of the layers of the CNN model, which was trained using the proposed method. Two columns corresponded to the same plot from different angles of the view. The axes indicate the acquired principal components. Each color of the point corresponds to the category of image.

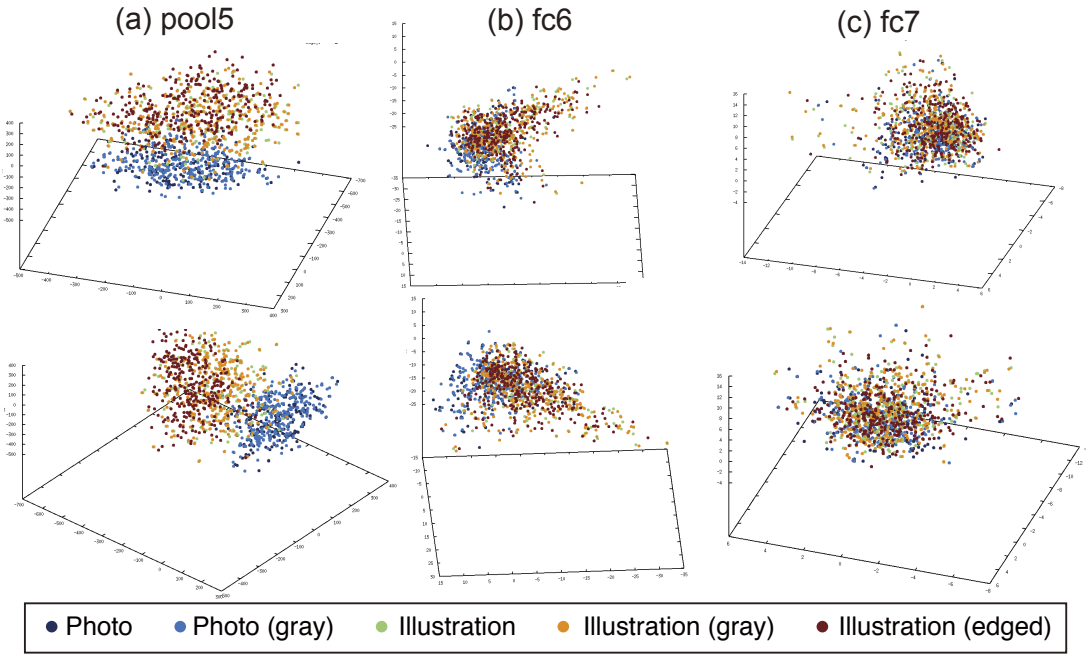


Figure 4.5: Image features of the images categorized as bears. Each column represents the same plot from a different angle of the view. The color of the plots indicates the type of image.

To interpret how the trained CNN model organizes the types of images, we conducted another visualization process. In this case, the features were obtained from three consecutive layers after the convolutional layers (see Figure 4.5): (a) pool5 means the first pooling layer of the model; (b) fc6 indicates the fully connected layer, which is next to the layer (a); and (c) fc7 means second from the last fully connected layer. These visualization results were obtained by PCA and Figure 4.4, but the colors corresponded to the types of images. As shown in Figure 4.4, the features cluster by the types of images at pool5 ((a) in Figure). By shifting to the deeper layer, they gradually gather and form a single cluster.

We quantitatively evaluated how the image features were gathered. To measure the distances between each type of image, we used the metric S, which was the ratio between the two covariances s_w and s_b:

$$s_w = \frac{1}{N} \sum_{i \in class} \sum_{m_i \in m} (m - \overline{m_i})^{\mathrm{T}} (m - \overline{m_i}) \tag{4.2}$$

$$s_b = \frac{1}{N} \sum_{i \in class} (\overline{m_i} - \overline{m})^{\mathrm{T}} (\overline{m_i} - \overline{m}) \tag{4.3}$$

$$S = \frac{s_b}{s_w}. \tag{4.4}$$

Here, m is a vectorized image feature; s_w is the between-class covariance of the image features whose mean is indicated as $\overline{m_i}$; s_b corresponds to the within-class covariance; and $\overline{m}$ indicates the mean of all the features. We obtain S of the dimensionally compressed image features depicted in Figure 4.5. In this case, the class of the image indicated the image type, not the type of animal. Consequently, S scored 0.53, 0.11, 0.03 at the pool5, fc6, and fc7 layers, respectively. These scores imply that the obtained features were gradually gathered by shifting the target layer.

4.5 Discussion and Future Work

Through our experiments, we confirmed that the inclusion of illustration images contributed to an improvement of the CNN classifiers. These experimental results

reflect the contributions of culture-specific visual representations for a human's sketch recognition ability. However, the scope of the discussions is limited because the sketch images were drawn by five Japanese participants. In future studies, it is important to investigate cultural differences and variations across age and gender.

One way of using the proposed method is by using an application that searches images by sketching. This kind of application is called "content-based image retrieval" (CBIR) systems [94]. CBIR systems enable us to find images whose discriminating metadata do not exist. Instead of using keywords as a query, CBIR systems require images as the user's input. The use of hand-drawn sketch images as queries were investigated for sketch image-retrieval systems [95, 30]. To measure the similarity between hand-drawn sketch images and the images in the database, well-elaborated image feature extraction methods have typically been adopted [96, 30].

The recent success of deep learning has led to the appearance of CNN models for CBIR systems [97, 98, 99]. Classifying both the sketch and photo images by using a CNN model is challenging because of the two reasons mentioned in Section 4.2: large visual gap and the lack of large-scale dataset. To overcome the difficulty in training CNN, Sangkloy et al. proposed the use of two CNNs corresponding to each type of image [65]. They used a constraint to share the image features obtained from multiple CNN layers [100]. However, their method required a large computational memory to train two CNN models. The method proposed can be simpler than integrating the photo and sketch images because it did not require multiple CNN models.

4.6 Conclusion of Chapter

In this chapter, we described the experiments on the classification of the sketch and photo images by using CNN models. The proposed data argumentation method includes illustration images with color-converted versions into the training dataset of photo images. The proposed method was evaluated by performing experiments for discriminating the sketch and photo images of animals into 20 classes. Con-

sequently, the CNN model trained by the proposed method demonstrated the best classification accuracy rather than the models trained by other combinations of image types. The classification accuracy was also improved by including edge-emphasized sketch images. Furthermore, we tried to interpret how the CNN model organizes each type of image by the visualizations of the intermediate layers of the model. Finally, a possibility to use the proposed method for sketch image-retrieval systems was suggested in the last section.

Chapter 5

Visuomotor Adaptation for Drawing

In this chapter, several experiments on learning visuomotor drawing sequences are described. First, the proposed method to construct models based on deep learning for learning visuomotor sequences are introduced in Section 5.1. In this section, we also describe the method to enable generate-appropriate drawing motions from an image by reusing the obtained memory. Section 5.2 explains the experiments on a simulator environment. These experiments were conducted to demonstrate the ability of visuomotor adaptation for several scenarios of drawing interactions. In Section 5.3, the target of the experiments is expanded to the real world by using a humanoid robot. Finally, this section is concluded in Section 5.4.

5.1 Introduction

In this section, the proposed method to enable visuomotor learning of drawing experiences is described. First, a visuomotor sequence is defined. Then, the method for a model based on deep learning to acquire an integrated memory is explained. After that, a method to reuse the obtained memory is described.

5.1.1 Visuomotor Sequence of Drawing

In this Section, the drawing process is assumed to be a finite process. The process X has a sequence of images from the visual feedback received from the drawn picture and the corresponding drawing action state vector as follows:

$$X = (x_1, x_2, \cdots, x_t, \cdots, x_T) \tag{5.1}$$

$$x_t = (i_t, a_t), \tag{5.2}$$

where i_t is an image at the time step $t = (1, 2, \ldots, T)$, and a_t corresponds to the drawing action state. The drawing action state means the status of the drawer. For example, in the simulator experiments, the drawing action state means the position of a pen. In the case of real robots, the state will be a vector that represents the physical status of the robot, for example, the joint angles. We assume that all the processes start from the same state. This means that i_1 is an image without any lines, and a_1 represents a fixed starting position.

5.1.2 Learning Model

As the time step t increases, the picture is visually altered by drawing actions until the maximum step T. We present a function to evolve the process as a forwarding function f [101] as follows:

$$x_{t+1} = f(x_t). \tag{5.3}$$

We approximate this forwarding function by a RNN, which is a neural network model for learning sequential data, by retaining the memory as a state of the hidden layer. We formalize the RNN function as an appropriated forwarding function of a given drawing sequence as follows:

$$x_{t+1}, h_t = F(x_t, h_{t-1}), \tag{5.4}$$

where h_t is the state of the RNN's hidden state. To acquire an appropriate approximation of f under a given X, the learnable parameters of the RNN are acquired

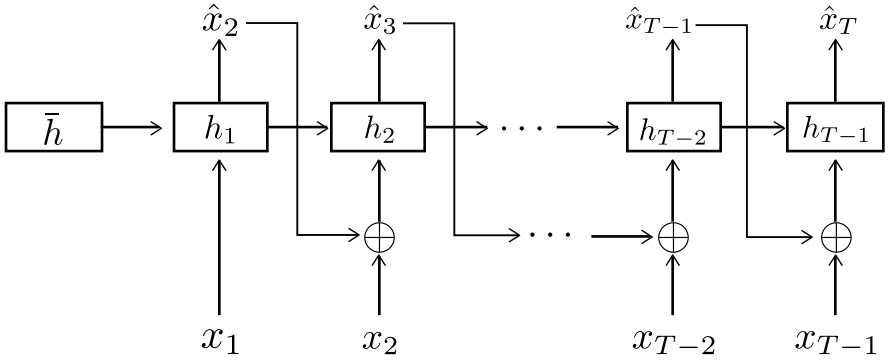


Figure 5.1: Forward propagation of RNN

by an optimization process to minimize the prediction error. The prediction error means the difference between the predicted state $\hat{x}_t$ and the target state x. The errors at each time step are accumulated into the loss of training L. L is given by

$$L = \sum_{t=2}^{T} l(\hat{x}_t, x_t). \tag{5.5}$$

Here, $\hat{x}$ is obtained by the forward propagation of RNN, and l is a loss function which refers to the similarity between $\hat{x}_t$ and x_t. The input to the RNN model can be taken from the target state x_t or the prediction at the previous step $\hat{x}_{t-1}$. When the model is fed the target state for all steps, we call it "open generation."

Another case is called "closed generation." In this case, the input to the model is replaced by the previous prediction. These modes of generation are presented as follows:

$$x_t = \begin{cases} x_t & (t = 1) \\ g_t^i x_t + (1 - g_t^i)\hat{x}_{t-1} & (t > 1), \end{cases} \tag{5.6}$$

where $g_t^i = (1, 0)$ is the gating value that determines the mode of the forwarding process. When $g_t^i = 1$ for all t, it means open generation; and when $g_t^i = 0$ for all t, it is closed mode.

We can consider setting the learnable state of the hidden state at the first step $\bar{h}$ as the "initial state." The initial state is input to the hidden state at the time step h_0. When the RNN model is trained to generate a number of sequences,

we assume multiple initial states for all the trained sequences. The sequence generated by the trained RNN model is affected by not only the input state but also by the initial state [102]. In particular, when the generation mode is closed, the generated sequence is determined only by the initial state because the first input is shared.

In this section, the training process of the RNN model was a minimization of the total loss of all the sequences:

$$L^{total} = \sum_{s} L^{s}, \tag{5.7}$$

where L^s is the accumulated loss between the s-th sequence of the target data and the predicted sequence generated by using the initial state $\hat{h}^s$. The initial states are updated by gradient descent and other learnable parameters of the model.

5.1.3 Reusing obtained memory

When the system draws a picture after the training process, the forwarding process of the trained RNN is the open-generation mode, but the model accepts the input not from the target data but from the current sensory input. The sensory input is assumed to consist of visual feedback from the drawn picture and the action status vector. After acquiring the inference from the model, the robot starts drawing according to the predicted action status.

Adaptation of the obtained visuomotor memory for associating drawing motion is implemented by a process to explore the initial state that can make the model draw the expected picture. To explore a good initial state, the gradient descent method was used. We acquired a sequence by using closed generation from an initial state vector initialized by using a random or a zero value. Then, the initial value was updated by the gradients given by the error between the image part of the generated sequence and the target picture. The error L^{exp} was accumulated in part of the time steps as follows:

$$L^{exp} = \sum_{t=2}^{T} g_t^o l^{img}(\hat{i}_t, i_t), \tag{5.8}$$

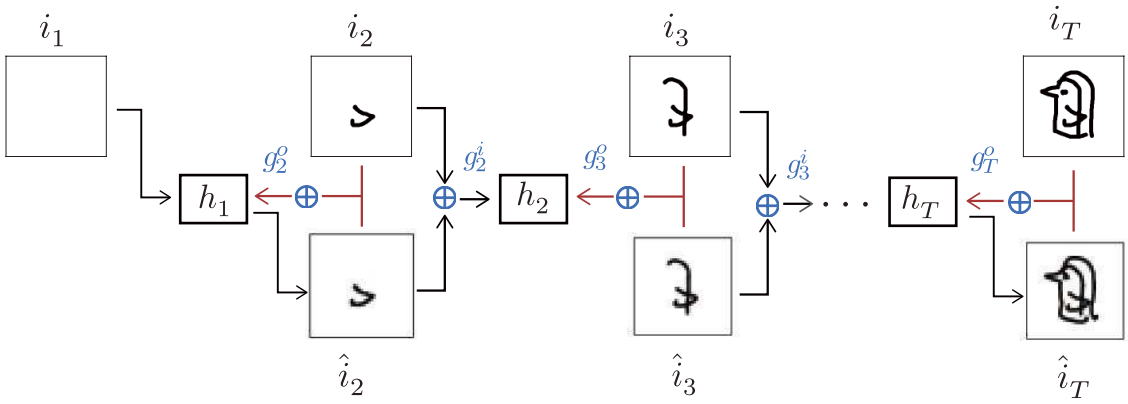


Figure 5.2: Gating for accumulating loss to explore an initial state

where L^{img} is a function to measure the similarity between the predicted image $\hat{i}_t$, and the image to be associated i_t. $g_t^o = (1, 0)$ is a gating value to determine whether or not the loss will be accumulated at the time step t.

The gating values for the forwarding process g^i to get i_t and the accumulating error g^o are determined by the association scenario, as shown in Figure 5.2. When the depiction target picture is given as a static image, g_t^i is zero for all the steps. Only at the last step, g^o is one because the images for the rest of the time steps are not given; therefore, $g^o = (0, 0, \cdots, 0, 1)$. When the first line is given in advance, g_t^i and g_t^o are one at the steps where the given line is drawn. Note that g_t^o at the last step is also one even if the first line is given.

5.2 Simulator Experiments

This section presents experiments to demonstrate the association ability of the proposed RNN model in a simulator experiment. First, we describe the architecture of the model in 5.2.1. The detail of the simulator environment is presented in 5.2.2. Subsequently, learning experiments conducted on 30 classes of drawing sequences are described in 5.2.3. In these experiments, we proposed that the RNN model demonstrates the ability to associate drawing motion from an image and another image conditioned by the first given line. In 5.2.4, we compare the proposed model with a model without vision.

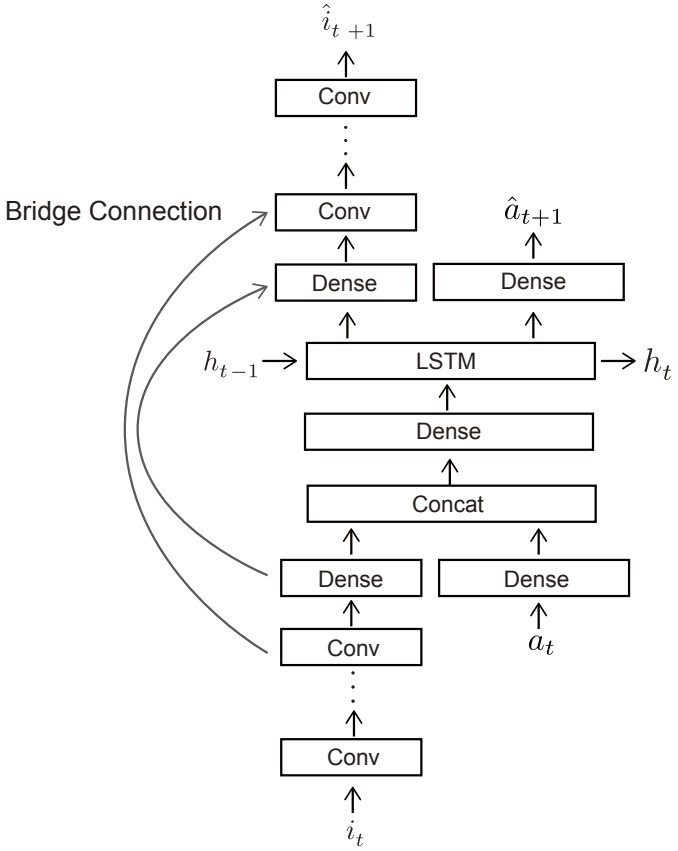


Figure 5.3: The architecture of the proposed RNN model for simulator experiments. Each box represents a RNN layer. Conv: Convolution2D or TransposedConvolution2D with the activation function. Dense: Linear mapping with activation function. LSTM: Long short-term memory cell.

5.2.1 Model Architecture

The model used in the simulator experiments is a RNN with convolutional layers, as shown in Figure 5.3. The model accepts the input consisting of an image i_t and the corresponding drawing action state a_t. To reduce the dimensionality of the image input, the convolutional layers are stacked to construct the encoding part. The encoded image features are concatenated with the feature of the action state encoded by the dense, connected layer. Then, the recurrent connection layer determines the output by using the concatenated feature as the input. We use LSTM [103, 104]) as the recurrent layer. The output of the recurrent layer is split into the image and action state features. The action state features become the

predicted action state by the decoding part implemented by a dense, connected layer. The image decoder part has transposed convolutional layers that accept the intermediate features of the image encoding part in addition to the previous layer's output. The bridge connections between the encoding and decoding part helps the decoding part to reconstruct the spatial information of the image [105]. The initial state of the recurrent layer is calculated by a dense, connected layer without bias term:

$$h_{t=0} = tanh(W^{init}\bar{h}), \tag{5.9}$$

where W^{init} is the weight matrix, which is one of the learnable parameters of the model. We apply this dense connection layer to reduce the dimensionality of the initial state. The small dimensionality of the hidden layer determines the capacity of RNN's memory.

The parameters of the RNN model are optimized by the stochastic gradient descent (SGD) method that uses the Adam optimizer [80]. The model generates sequences in the closed mode when it is trained. The implementations of loss functions for the image and the action state are described in each subsection.

5.2.2 Simulator Environment

The simulator used in the experiments described in this section has the function of drawing black lines with fixed thicknesses. The simulator gives binary image data as the image feedback from the canvas. All the drawing processes start from the center of the canvas. The drawing action state vector consists of the position of the pen and the binary representation that specifies whether or not the pen touches the canvas; another binary data specifies whether or not the drawing process has ended. This representation was introduced by [106], but this study uses the absolute position of the pen to avoid exceeding the size limits of the canvas.

Figure 5.4: Example of the training dataset

5.2.3 Experiments on Associating Drawing Motion From an Image

Experimental Setting

To demonstrate the visuomotor adaptation ability by using the proposed model, we conducted experiments on learning 30 classes of drawing samples. Firstly, the RNN model produced the drawing sequences from the initial state obtained in the training process. The objective of this experiment was to confirm the forwarding function obtained by the training process. Then, we checked the visuomotor adaptation ability in three cases: 1) when completed images were given, 2) when a completed image with the first line was given, and 3) all the lines excluding the last line were not given in the trained ordering. In the second case, the model was required to adapt not only to depict the given image but also to follow the given line. The final case associated the drawing motions to complete the given drawing process whose line ordering was unknown.

Figure 5.4 shows the pictures drawn by the training sample sequences. The sequences for training the proposed RNN model was collected by the experimenter. These sequences consisted of 30 categories. Pictures that had the same number of lines (e.g., 3-5 lines) were placed in one category, and the length of all the lines was less than 100 steps. The image obtained by the simulator i_t had 128×128 pixels.

Training a RNN model for a very long sequence is typically a difficult task because the gradients of the hidden state become unstable [107]. To enable training RNN for very long sequences, we split the training task into two sub-tasks that required two RNN models. We firstly trained a RNN model to learn a single line (LineRNN). Then, another RNN model was trained to control the LineRNN (PicRNN). The structures of these RNN models were shared, but they had different learnable parameters.

LineRNN accepted the input that consisted of the drawn picture image i_t^L and the drawing action state a_t. The drawing action state was the set of pen positions and the status that specified whether or not the pen was touching the paper:

$$a_t = (p_t^1, p_t^2, q_t^L), \tag{5.10}$$

where p_t^1 and p_t^2 are the absolute positions of the pen on the x-y axis. The variable q_t^L indicates the status of the pen's lifting. If $q_t^L = 1$, a line will be drawn when the pen moves following the next position. The variable i_t^L has information of only a single line. Therefore, i_t^L does not include the previous lines in which the simulator added another line to the canvas.

The optimization loss of LineRNN is given by accumulating the losses of the image and the action:

$$L^{Line} = \sum_t [l^{img}(\hat{i}_t^L, i_t^L) + l^{act}(\hat{a}_t^L, a_t^L)], \tag{5.11}$$

where l^{img} is the loss function that accepts the target image i_t^L and the predicted image $\hat{i}_t^L$; l^{act} is another loss function for the target action state a_t^L and the prediction $\hat{a}_t$. These loss functions are defined as follows:

$$l^{img}(\hat{i}_t^L, i_t^L) = i_t^L \log \hat{i}^L + (1 - i_t^L) \log(1 - \hat{i}_t^L). \quad (5.12)$$

$$l^{act}(\hat{a}_t^L, a_t^L) = -\log \mathcal{N}(p_t^1, p_t^2 | \hat{\mu}_t^1, \hat{\mu}_t^2, \hat{\sigma}_t^1, \hat{\sigma}_t^2, \hat{\rho}_t) - q_t^L \log \hat{q}_t^L. \quad (5.13)$$

Here, l^{img} corresponds to the cross-entropy of the binary image between i_t^L and $\hat{i}_t^L$; l^{act} is given by the negative log likelihood of a bivariate normal distribution and the cross-entropy of the pen's status; $\hat{\mu}_t^1$ and $\hat{\mu}_t^2$ are the mean values of the x- and y-coordinates, respectively; $\hat{\sigma}_t^1$ and $\hat{\sigma}_t^2$ are the variances of the x- and y-coordinates, respectively; and $\hat{\rho}_t$ is the covariance is the .

LineRNN has the initial state $\bar{h}^L$ for each trained sequence. These initial states and the learnable parameters are obtained by stochastic gradient descent.

PicRNN also accepts the image and the drawing action status. However, the input image i_t^P of PicRNN differs from that for LineRNN. The variable i_t^P has previous lines to include the information of line ordering. The drawing action status for PicRNN a_t^P consists of the probability of determining whether or not the drawing process has ended; the initial state of LineRNN $\bar{h}^L$ (instead of the drawing action status) is

$$a_t^P = (\bar{h}_t^L, q_t^P), \quad (5.14)$$

where q_t^P is the probability of end of the drawing. The loss for training PicRNN is given as follows:

$$L^{Pic} = \sum_t [l^{img}(\hat{i}_t^P, i_t^P) + l^{Pic}(\hat{a}_t^P, a_t^P)], \quad (5.15)$$

where l^{Pic} is a loss function of the action status of PicRNN. The value l^{Pic} is given by adding the mean square loss of $\bar{h}^L$ and the cross-entropy of the probability.

To train PicRNN, LineRNN should be trained to acquire the initial states $\bar{h}^L$. In the experiments, the obtained value of $\bar{h}^L$ is normalized into the specific range because the activation function of the PicRNN's drawing action state is *tanh*. When the predicted initial state of LineRNN is given to draw a line, the state is normalized again by using the *artanh* function. At the beginning of the training process of LineRNN, $\bar{h}^L$ is initialized by using a zero vector. After

Table 5.1: Parameters of LineRNN

PART	NUM
Image Encoder	4 (conv), 2 (dense)
Image Decoder	4 (conv), 2 (dense)
Action Encoder	1 (dense)
Action Decoder	1 (dense)
RNN	200 cells
Initial State	30 dims

Table 5.2: Parameters of PicRNN

PART	NUM
Image Encoder	4 (conv), 2 (dense)
Image Decoder	4 (conv), 2 (dense)
Action Encoder	-
Action Decoder	1 (dense)
State Encoder	1 (dense)
State Decoder	2 (dense)
RNN	200 cells
Initial State	10 dims

training LineRNN, PicRNN is trained by using the normalized initial states and the probability q_t^P. PicRNN is also allowed to have its initial state $\bar{h}^P$.

The parameters of LinerNN and PicRNN are presented in Tables 5.1 and 5.2, respectively. We use *tanh* as the activation function for all the layers excluding the part for the probability. The probabilities are given by the softmax function. The position of the pen and value of the image pixels are normalized in the range $(0, 1)$.

The hyperparameters of Adam Optimizer were $\alpha = 0.001$ and $\beta 1 = 0.75$. First, LineRNN was trained by using all the lines of the dataset. Then, PicRNN was trained to produce the obtained initial state of LineRNN, the probability of drawing process, and the picture image. To extend the dataset, we increased the number of samples by changing the ordering of lines. However, single ordering for each category was not added to the dataset.

Association of drawing motion was implemented by exploring the initial state

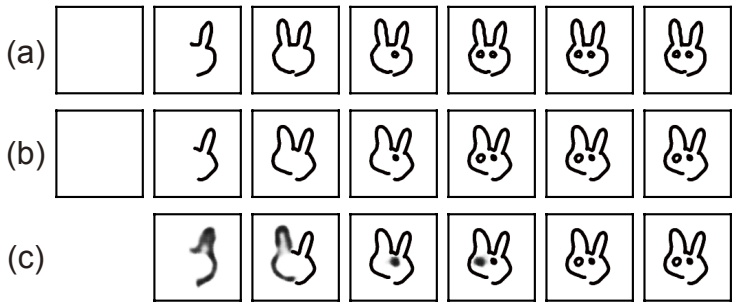


Figure 5.5: Results generated by the trained RNNs. (a): Ground truth from the training dataset. (b): Pictures drawn by the model. (c): Predicted drawing images.

of PicRNN. For each exploration, we initialize 50 candidates of $\bar{h}^P$ by using a normal distribution with zero means and a variance of one. After every 100 updates, half of the worst candidates were replaced by a new normal distribution whose mean of the better candidates. The candidates were measured by the loss value of exploration. After exploration, the models drew a picture using the best candidate.

For the scenario of adaptation, we performed three cases of association: 1) when the image in the last step was given, 2) when the images in the last step and the steps of the first lines were given, and 3) when the images in all steps were given excluding the part of the final line.

The gating values used to obtain the association loss were given by the experimenter. For associating a motion from a picture image in only the last step, we used $g_i = (1, 0, 0, \cdots, 0)$ and $g^o = (0, 0, \cdots, 0, 1)$. When the first line was also given, the gating value was one at $t = 2$.

Generating from the obtained initial state

Figure 5.5 shows the drawing results obtained by the trained model with one of the trained initial state values. Figure 5.5 (a) presents the snapshots of the training dataset. The predicted image sequence and the history of the drawn picture are shown in Figures 5.5 (b) and (c), respectively. At each step of the PicRNN's prediction, LineRNN draws a line by using the initial state predicted by PicRNN. As shown in the figure, the trained model draws lines in the same order as the

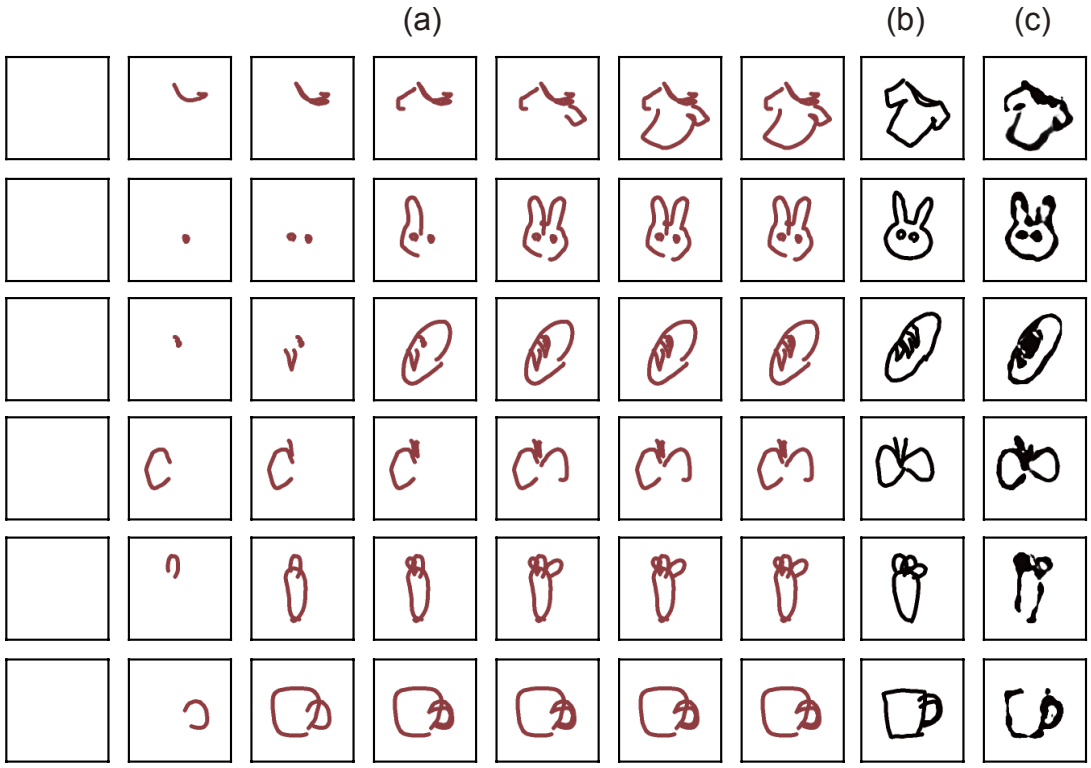


Figure 5.6: Results of the initial value exploration. (a) Pictures drawn by the model. (b) Depiction target of the images. (c) Predicted drawing images at the final step of the sequence associated with the initial value.

target sequence. Finally, the picture drawn by the model is a picture similar to the final state of the target sequence.

The association results from the picture images are summarized in Figure 5.6. We conduct explorations of the initial state of PicRNN for each picture image shown in Figure 5.6 (b). The loss of exploration is given by the prediction error between a given depiction target and the image prediction made by PicRNN. Note that the target picture images are not included in the training dataset. The model draws pictures that are similar to the given pictures. This suggests that the proposed model demonstrates the association ability of the drawing process from a picture image.

To demonstrate the adaptation ability of the association by the proposed model, other exploration experiments were conducted. Figure 5.7 presents the drawing process for depicting the images shown in (c). These exploration pro-

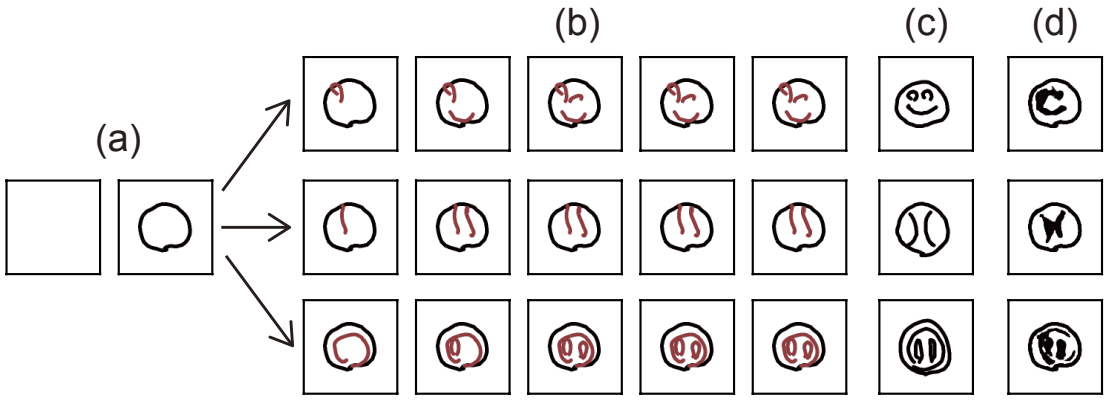


Figure 5.7: Results of the initial value regression on sketches starting with a line. The black lines are given by the experimenter. The red lines are drawn by the proposed model.

cesses were conditioned by the given first line. In this case, a loss of the initial state exploration was given by prediction errors at not only the final step but also the first step when the model drew the first line. Consequently, the picture images predicted by the model were similar to the given picture image and were depicted by adding appropriate lines to the first line.

To confirm the generalization ability for line ordering, we conducted initial value exploration tasks whose results are described in Figure 5.8. This task requires the model to add the last line to the incomplete drawing process. The lines of the given drawing process are given in an untrained order. The given drawing process is not trained as well as the other experiments. As shown in Fig. 5.5, the model adds the last line, which is similar to the ground truth from the untrained data.

5.2.4 Comparison with RNN Model Without Vision

To clarify the effects of including image feedback, we conducted experiments to compare the proposed model and RNN without vision. The task was to depict faces when the first line was given in advance. Further, we also conducted experiments in case an additional line was included in the given sequence. The aim of adding lines was to confirm the robustness against "noisy" lines. The first line was given as an outline of the face, and each RNN model was required to depict a

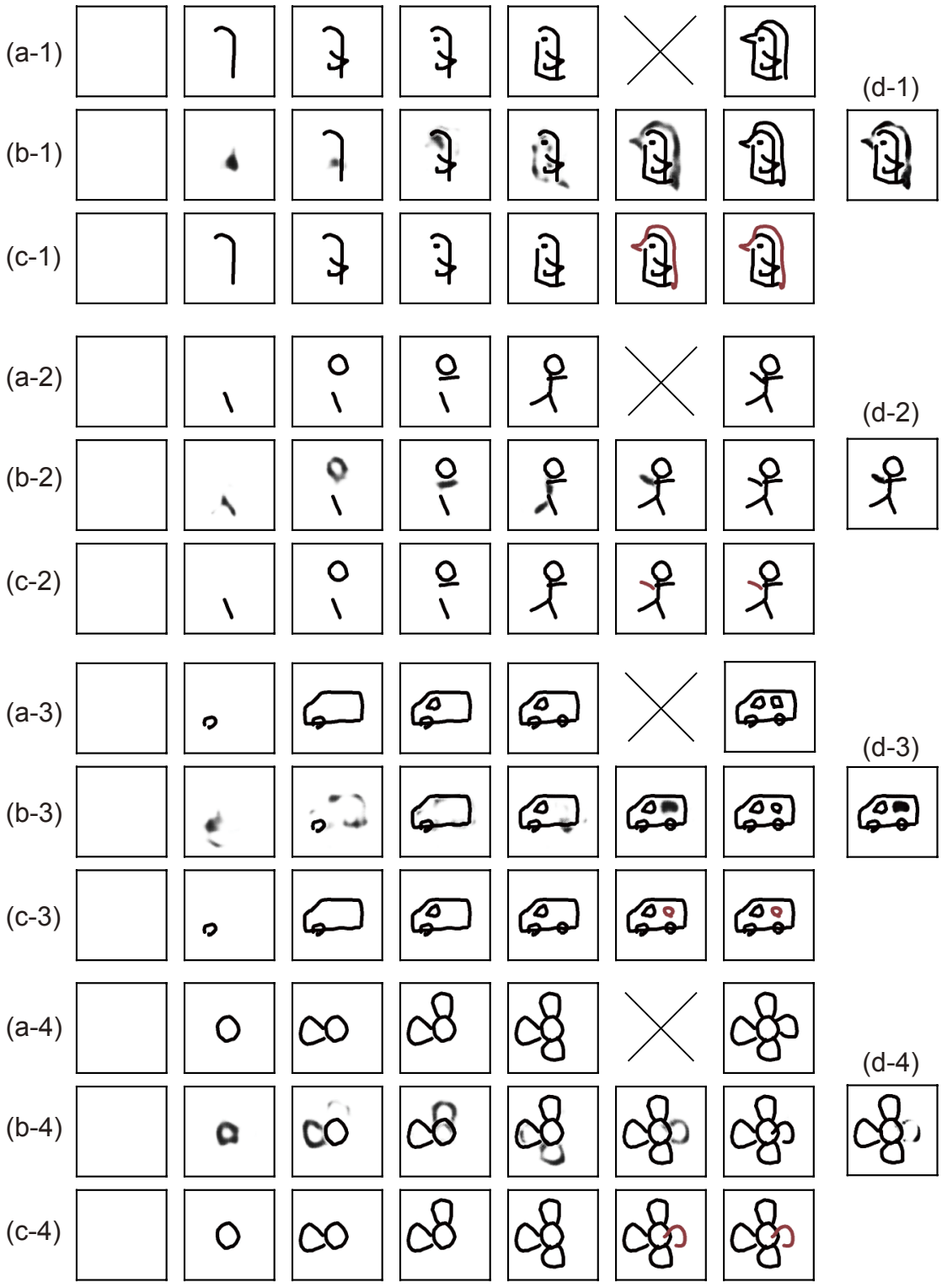


Figure 5.8: Results of initial value regression on untrained combinations of lines. Black lines were given by the experimenter. Red lines were the lines drawn by the model.

Table 5.3: Parameters of Model

PART	NUM
Image Encoder	4 (conv), 2 (dense)
Image Decoder	4 (conv), 2 (dense)
Action Encoder	1 (dense)
Action Decoder	1 (dense)
RNN	256 cells
Initial State	10 dims

face by adding more lines. The noisy lines were placed outside the given outline in all cases.

Experimental Settings

For the training dataset, we randomly selected 1000 samples from the sequences of the QuickDraw Dataset [108] in the Magenta project [109]. To reduce the computation cost to train RNNs, we used only the sequences whose length was less than 100 steps. To acquire sequences of drawn pictures, we used the simulator that was used in previous experiments. The images were 64×64 binary pixels. Note that the position of the pen was converted from the relative point to the absolute point to avoid going beyond the canvas.

As a RNN model to be compared with the proposed model, sketch-rnn by Ha et al. [79] was used. This model is a variational autoencoder [110] for reconstructing the drawing process. To accept and generate sequential data, the decoder and encoder parts were implemented by LSTMs. Ha et al. also reported unconditional generation by using only the decoder LSTM. In this study, we used only this decoder LSTM model as the learner of the drawing sequences. This model reconstructs the drawing sequence that consists only of the pen status. Thus, it cannot accept visual feedback from the drawn picture. The recurrent layer of the model had 1024 LSTM cells and five-dimensional latent variable expressions as the initial state. The model predicted the position of the pen based on multiple Gaussian distributions, and parameters were presented to control the randomness of selection. We set this parameter as 0.25 because it produced stable drawing results.

The model proposed in this study was implemented by using a RNN model with image prediction. Unlike what we did in the previous experiments, we did not split the training task because the trained sequences were short. The loss of the training process was given as follows:

$$L^* = \sum_t [l^{img}(\hat{i}_t, i_t) + l^{act*}(\hat{a}_t^L, a_t^L)] \tag{5.16}$$

$$l^{act*} = -\log \mathcal{N}(p_t^1, p_t^2 | \hat{\mu}_t^1, \hat{\mu}_t^2, \hat{\sigma}_t^1, \hat{\sigma}_t^2, \hat{\rho}_t) - q_t \log \hat{q}_t. \tag{5.17}$$

where q_t is the probability of the pen status used in sketch-rnn, and q_t consists of a three-dimensional vector. The parameters of the layers are described in Table 5.3. Associating lines to complete the picture of the face was also done by exploring the initial state. The exploration loss was given by the prediction error at the final step and at the steps when the conditional lines were drawn. Fifty candidates of the initial state were initialized by uniform distribution whose value range was $[-0.1, 0.1]$. The candidates were updated by using a normal stochastic gradient descent with momentum. Note that the candidates were not reinitialized in these exploration processes.

Results

Figure 5.9 describes the results by of both the sketch-rnn and the proposed model. Each column of the figure corresponds to the input lines; the pictures are drawn by sketch-rnn (without vision) and by the proposed method. The initial state of the proposed method was obtained by explorations for associating the picture image shown in the upper right side of the "target image". Each row corresponds to a different input line. Specifically, the rows from (b-1) to (b-5) indicate the results of noisy line inputs. As shown in the figure, when the beginning portion of the process is given by only a single line, both the models draw parts of the face inside the output line. Facial expressions by sketch-rnn have variations because its generation process cannot be conditioned by picture images. The pictures using the proposed model have distortions, but the faces drawn have mostly the same structure as the depiction target. Sketch-rnn tends to put the parts outside the

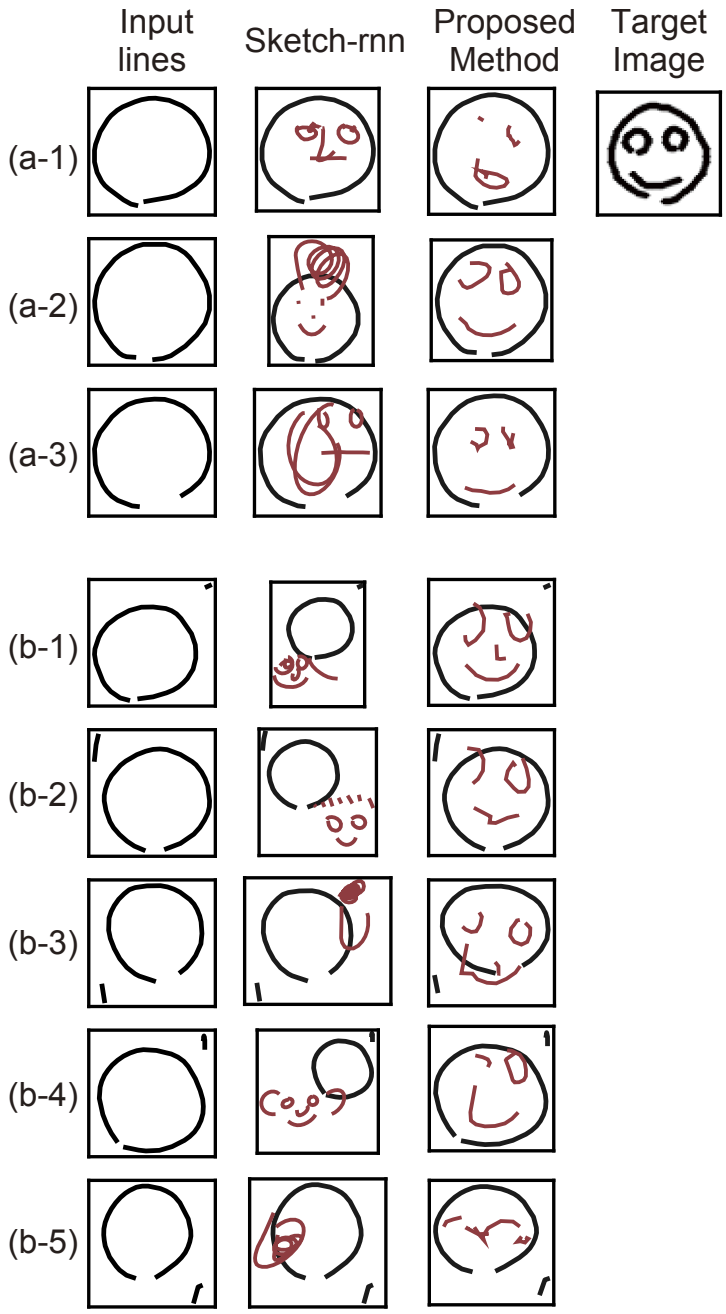


Figure 5.9: Results of drawing adaptation to the given input lines. Black lines are given by the experimenter. Red lines are drawn by the models.

given outline in the case of the noisy line input. In comparison with the sketch-rnn results, the proposed model succeeded in drawing the outline inside.

5.2.5 Discussions for Simulator Experiments

In the simulator experiments, the proposed RNN models demonstrated the visuomotor adaptation ability of drawing. The adaptation ability means to associate appropriate drawing motions from the picture image. This association process is conditioned by the prediction errors. The functionality of the implemented drawing system includes not only accepting visual feedback from the drawn picture but also predicting the future state of drawing. The image prediction enables the models to obtain the error that can be used for exploring the initial state of the RNN's hidden layer. In the proposed architecture, the initial state functions as a conditional input to the system.

The association ability demonstrated in the simulator experiments allows us to add another condition to determine the model's behavior. The loss for the initial value exploration could have been a loss for the picture image not only in the last step, but also in the steps during the process. In other words, by changing the exploration loss, the models show flexible adaptation to the given picture image sequences. First, the model generates all the drawing motions to depict a given picture image. This situation of association corresponds to remembering the depiction process and recovering them [90]. By adding image prediction losses, we can include a more complex association scenario-conditioning by using given lines. The proposed RNN-based system demonstrated that it could complete the given process by drawing the final line. Also, the system could change its behavior depending on the depiction target even if the beginning part of the process was shared.

In the simulator experiments, we confirmed that the drawing behavior by a proposed RNN model could be conditioned by using picture images. Although the original sketch-rnn can reconstruct the given drawing motions by using another LSTM as the encoder, the condition can be given as raster image data. Furthermore, the experimental results suggest not only the necessity of visuomotor learning for association but also the possibilities for drawing applications. The ro-

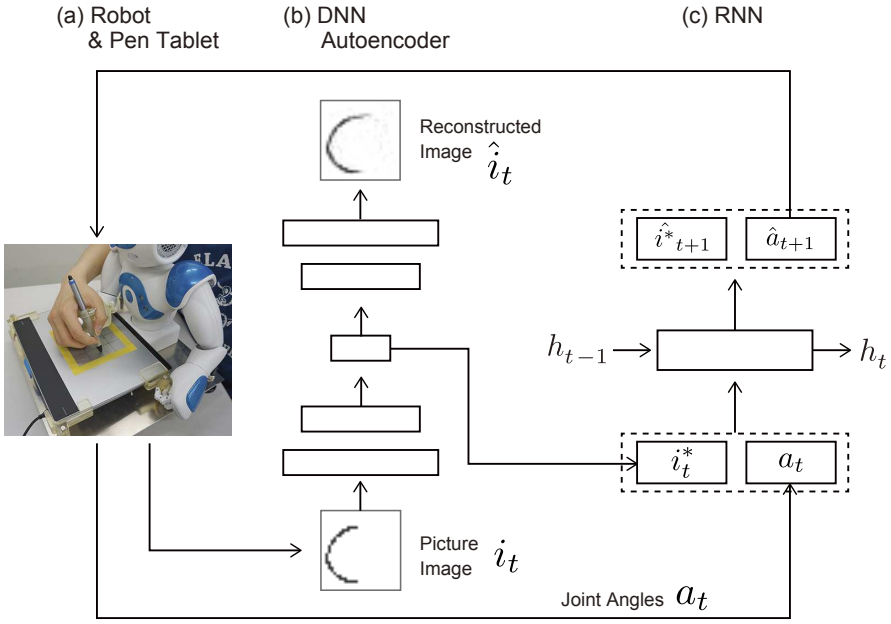


Figure 5.10: The architecture of the RNN-based system for robotic experiments (a) A humanoid robot with a stylus pen. (b) A DNN autoencoder. (c) The RNN model for modeling visuomotor sequences.

bustness against noisy lines is an important factor to consider while constructing systems for understanding or helping the drawing process.

5.3 Robot Experiment

In this section, several visuomotor adaptation experiments using a humanoid robot are described. In the subsection 5.3.1, the implementation of a RNN-based model for visuomotor sequential learning was given. Subsequently, two experiments on learning the drawing process of simple shapes were explained. In the subsection 5.3.3, the first experiment that associated drawing motions from a picture image was described. The second experiment confirmed that the obtained visuomotor memory could be used for recognizing picture images. Finally, the adaptation abilities described in this section are discussed in subsection 5.3.5.

5.3.1 Model Architecture

Figure 5.10 describes the architecture of a RNN-based system for learning a visuomotor sequence of the robot's drawing process. The use of neural network models was inspired by multimodal integration learning using autoencoders [111]. An autoencoder is a DNN model for acquiring low-dimensional features by unsupervised learning. Hinton et al. proposed to construct an autoencoder model by stacking many layers of feedforward neural networks [57]. Their model can be applied to large-dimensional data, such as raw image pixels. We use this DNN autoencoder to acquire the feature of an image from the drawn picture. The RNN model accepts and predicts the dimensional compressed image feature instead of the raw image data. This technique reduces the computation cost to optimize the model parameters.

A DNN autoencoder consists of the encoder e that reduces the dimensionality of the input i_t and the decoder d to reconstruct the input from the encoded feature. The forward propagation of the reconstruction is formalized as follows:

$$i_t^* = d(i_t) \tag{5.18}$$

$$\hat{i}_t = e(i_t^*), \tag{5.19}$$

where i_t^* is a low-dimensional feature of i_t. Both the encoder and the decoder are implemented by using a linear map with a sigmoid as the non-linear activation function. To train this network, we use a truncated Newton-optimization method [112]. The cost function is defined as the mean square error between the reconstructed input data $\hat{i}_t$ and the corresponding input i_t.

To implement the RNN model, we use a continuous time-scale recurrent neural network (CTRNN) [113]. CTRNN is a RNN model typically used for learning continuous sequences such as a robot's motion. The state of the hidden layer of CTRNN is computed not only by the input but also by its own previous state. The time responsiveness of the hidden layer is determined by a time constant value, which is one of the hyperparameters. Specifically, the CTRNN model has hierarchical connections between hidden layers that have different time constant

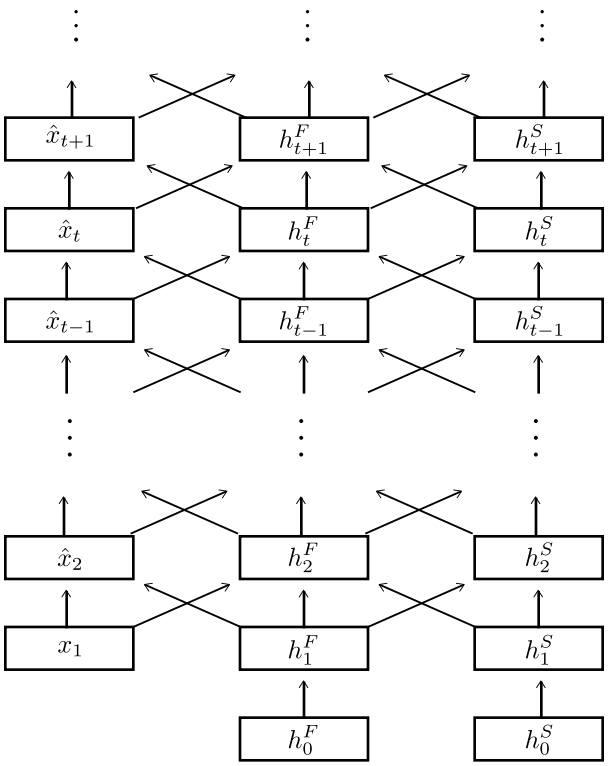


Figure 5.11: Hierarchical connectivity of CTRNN in the case of the closed generation mode.

values, as shown in Figure 5.11 (Figure 5.11). The inference by this model $\hat{x}_t$ at the time step t is given as follows:

$$
\begin{aligned}
u_t^F &= (1-\frac{1}{\tau^X})u_{t-1}^X + \\
&\quad \frac{1}{\tau^X}(W^{XX} \cdot \hat{x}_{t-1} + W^{XF} \cdot h_{t-1}^F + b^X) \\
u_t^F &= (1-\frac{1}{\tau^F})u_{t-1}^F + \\
&\quad \frac{1}{\tau^F}(W^{FX} \cdot \hat{x}_{t-1} + W^{FF} \cdot h_{t-1}^F + W^{FS} \cdot h_{t-1}^S + b^F) \\
u_t^S &= (1-\frac{1}{\tau^S})u_{t-1}^S + \\
&\quad \frac{1}{\tau^S}(W^{SS} \cdot h_{t-1}^S + W^{SF} \cdot h_{t-1}^F + b^S),
\end{aligned}
\tag{5.20}
$$

where $u_t^X, u_t^F, and u_t^S$ are the internal states of the hidden layers named IO, CF, and CS, respectively. The variables $\hat{x}_t, h_t^F$, and h_t^S indicate the states of the hidden layers activated by the nonlinear activation function σ as follows:

$$\hat{x}_t = \sigma(u_t^X) \tag{5.21}$$

$$h_t^F = \sigma(u_t^F) \tag{5.22}$$

$$h_t^S = \sigma(u_t^S). \tag{5.23}$$

The hidden layers each have their time constant value τ. The initial states of these layers are given as follows:

$$\hat{x}_0 = 0 \tag{5.24}$$

$$h_0^F = 0 \tag{5.25}$$

$$h_0^S = \bar{h}, \tag{5.26}$$

where $\bar{h}$ is a learnable parameter for the s th trained sequence. The model generates a sequence by using the closed mode, and we give $\hat{x}_1 = x_1$ as the initial

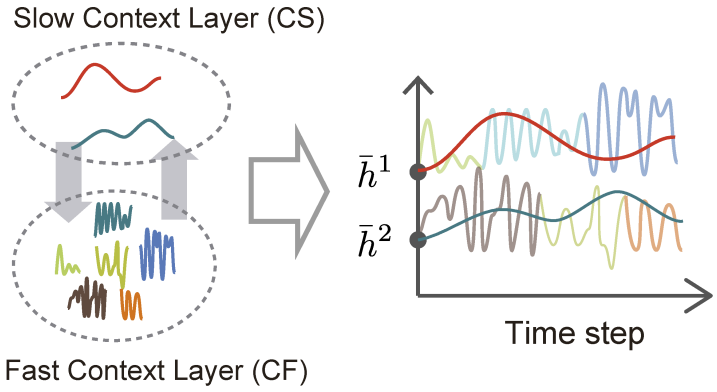


Figure 5.12: An example of the dynamics of CTRNN with hierarchical connections. Two sequences are generated from difference initial states.

input.

The time constant values and the hierarchical connectivity between the hidden layers makes it possible for the model to organize a complex sequence as a combination of multiple sequences having different time scales (Figure 5.12). The generated sequence is determined by the initial state of the CS layer $h_0^S = \bar{h}$ because the other input to the model is shared. The initial state of CS layer is allowed to be changed during the training process. Each initial state of CS corresponds to the trained drawing sequences.

The total loss was calculated by accumulating the loss for all the sequences to be trained. For the loss function to train the CTRNN model, we used the gradient descent method. To acquire the gradients of each parameter, the loss function was calculated for every iteration. The loss function was defined as the error between the generated sequence and the target data:

$$L = \sum_{s} \sum_{t=2}^{T} ||x_t^s - \hat{x}_t^s||_2^2, \tag{5.27}$$

where x_t and $\hat{x}_t^s$ are the states of the s-th learnt visuomotor sequence at the time step t.

Although CTRNN can remember multiple sequences, its capacity for generating large-dimensional inputs, such as raw image pixel data, is limited. Also, the imbalance of dimensionality between the image and drawing action states (joint

angles) causes poor regression accuracy for low-dimensional data, i.e., the drawing actions. Therefore, we use the dimensionally compressed image feature by the DNN autoencoder. In other words, the CTRNN is trained to generate the vector (i_t^*, a_t) instead of (i_t, a_t), as x_t.

To collect the drawing data for constructing the training dataset, we conducted a direct teaching experiment. An experimenter held the robot's arm, and let the robot move its hand to depict shapes. When the robot's arm was moving, the drawn picture image was recorded as a raster image. To capture the images when the robot was drawing, we used a pen tablet. After collecting the dataset, the DNN autoencoder was trained to reconstruct all the collected images. Then, we trained the CTRNN model by using the dimensionally compressed feature of the trained DNN autoencoder and collected the robot's joint angle vector as the drawing action state.

Figure 5.13 describes the overview of the initial state exploration method for the visuomotor adaptation using the trained models. After training the models, they can associate the drawing process by reusing the obtained parameters. First, the encoder part of the trained DNN autoencoder converts the depiction target image and the image at $t = 1$ to the dimensionally compressed image features. Then, we can calculate the loss between the predicted feature produced by the trained CTRNN model and the image feature of the depiction target at $t = T$. The new initial state of the CS layer is obtained by back-propagating this prediction error using recurrent connections. Note that the candidates for the explored states are initialized by the zero vector, and we use one of the candidates, which gives the minimum error.

5.3.2 Robot Experimental Setup

The experiments using the proposed models were conducted by using a humanoid robot NAO [114]. The robot drew a picture using an Intuos pen tablet as shown in Figure 5.14. The image feedback is acquired by rendering the history of the pen's position during the drawing process. The images are rendered as 30×30 pixel binary images of single lines whose width was fixed for all sequences. The pen was attached to the end effector of the robot. To avoid capturing the error

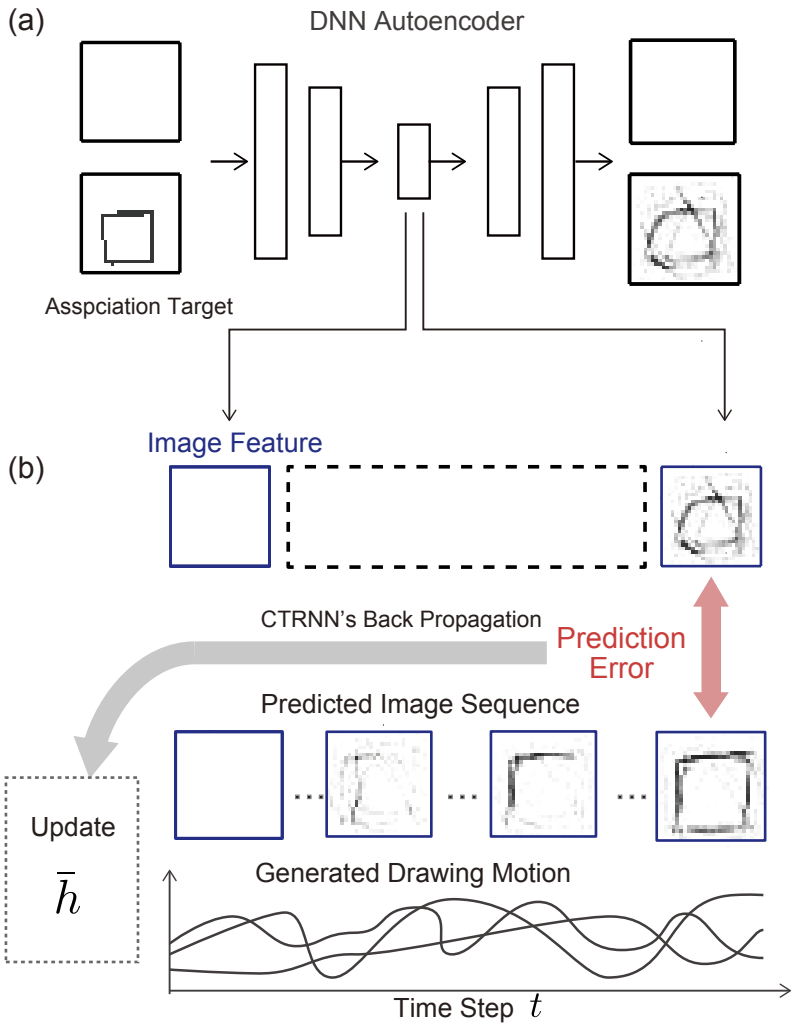


Figure 5.13: Overview of the initial state exploration. (a) The process to acquire image features at the first and final step encoded by the DNN autoencoder. (b) Path of backpropagation through time to calculate the gradient of an initial state $\bar{h}$.

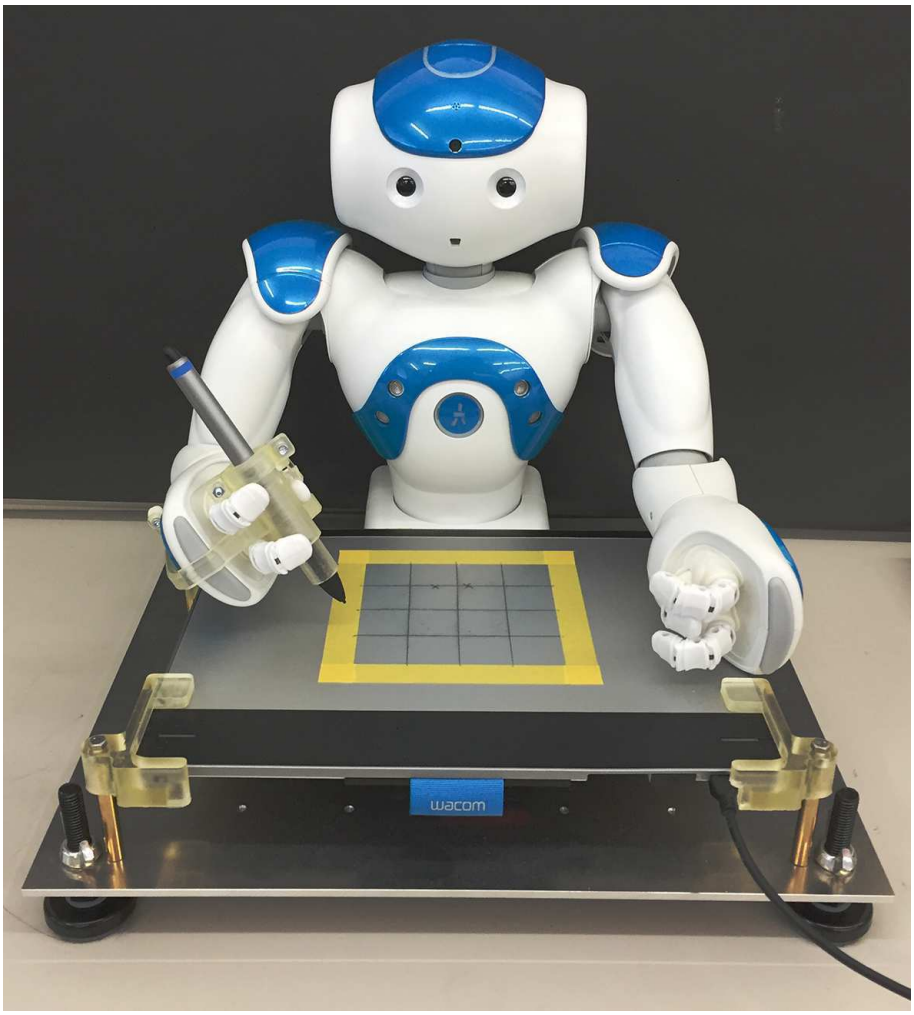


Figure 5.14: Snapshot of the robot experimental setup.

Table 5.4: The number of collected images and hyperparameters of the proposed model. IO-CF-CS and DIMS give the dimensions of the layers in CTRNN and DNN, respectively. DATA refers to the number of images recorded as the training dataset. TRANS and ROTATED are the numbers of translated and rotated versions, respectively, of the originally recorded images. Training Iter corresponds to the number of training iterations.

Param	DNN	CTRNN
IO	900	15($\tau^X = 1$)
DIMS	900-400-180-80-30-10	30(CF, $\tau^F = 12$), 20(20, $\tau^S = 60$)
DATA	494	494
TRANS	31940	-
ROTATED	2910	100
Training Iter	100	15000

and breaking the robot, the pen was allowed to move vertically (Refer Appendix B for further the experimental setting.) In the drawing action state, the five joint angles of the right hand were recorded at each time step.

5.3.3 Experiments on Robot's Drawing of Simple Shapes

The first experiment was to demonstrate the visuomotor adaptation ability of the proposed models. We collected 15 drawing sequences as the training dataset. These collected sequences consisted of squares, circles, and triangles. All the pictures were drawn by a single line, and each type had five variations. These variations shared approximately the same initial starting point. The duration of the drawing process was approximately between 5 and 10 s for 15 to 50 time steps of the sequence. All the lines were drawn clockwise.

Table 5.4 describes the experimental settings of the DNN autoencoder and CTRNN. We followed the model by Noda et al. to design the structure of the DNN autoencoder. DNN accepts 900 vector data as the input, and the encoder part gives a ten-dimensional feature of the input. The dimensional structure of the layers in the decoder and the encoder are shared. Therefore, the decoder gives 900 vector data as the reconstructed input. To avoid the vanishing of gradients with respect to either modalities, we set the dimensional size of the image feature to be

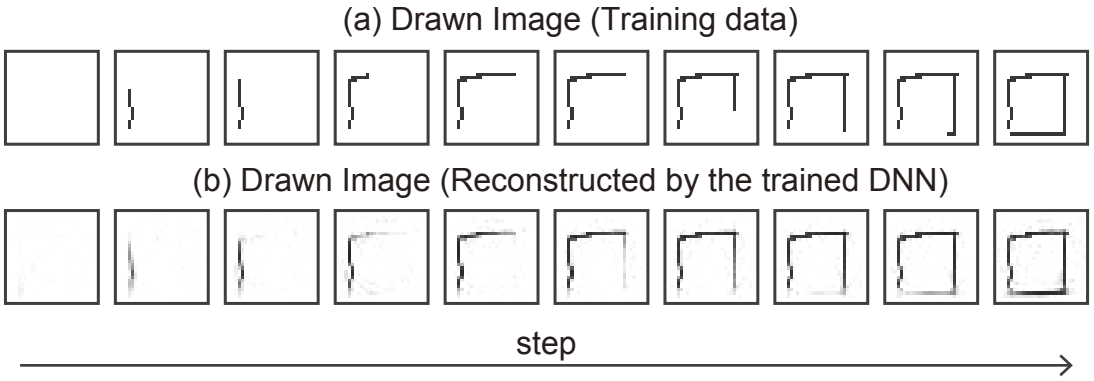


Figure 5.15: An example of reconstruction results by the DNN autoencoder. (a) The picture image sequence sampled from the training dataset. (b) The reconstructed results.

close to the dimensional size of motion. We trained the DNN autoencoder for 100 iterations by using the dataset of the captured drawn picture image frames. To avoid overfitting, the size dataset was increased by translation and rotation. The CTRNN model accepts a vector that includes the ten-dimensional image feature and the five-dimensional vector of joint angles as the input. The model had 30- and 20-dimensional hidden layers. The sizes of the CTRNN's hidden layers were selected from the candidates by checking the generation loss after 15,000 iterations.

Generating Trained Sequences

To confirm that the proposed models remembered the collected visuomotor sequences, the trained model generated drawing sequence data by using the parameters obtained in the training process. Figure 5.15 provides an example of the reconstructed picture image sequence generated by the DNN autoencoder. Figure 5.15 (a) shows the snapshots from one of the drawn picture image sequences used for training. Figure 5.15 (b) represents the images reconstructed by the trained DNN autoencoder. As shown in the figure, the DNN model successfully reconstructed the input data.

The generation results of the CTRNN model are summarized in Figure 5.16. To obtain the drawn pictures by using the trained models, we reuse the obtained initial state of the three trained sequences that had different shapes. Figure 5.16

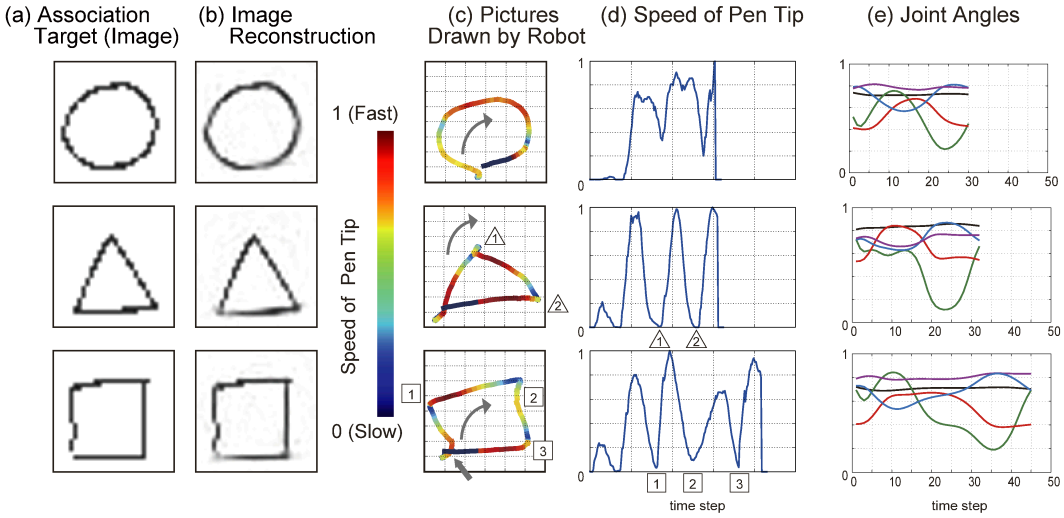


Figure 5.16: The generation results from the obtained initial state. (a) Images at the end of the drawing process in the training dataset. (b) Reconstruction results by the DNN autoencoder. (c) Line drawn when the robot draws using the motion sequences obtained by CTRNN. The lines are colored by the value of the normalized value of the speed of the pen tip (d). (e) Normalized joint angle sequences generated by the CTRNN. The numbers in (c) and (d) correspond to the corners of the shapes.

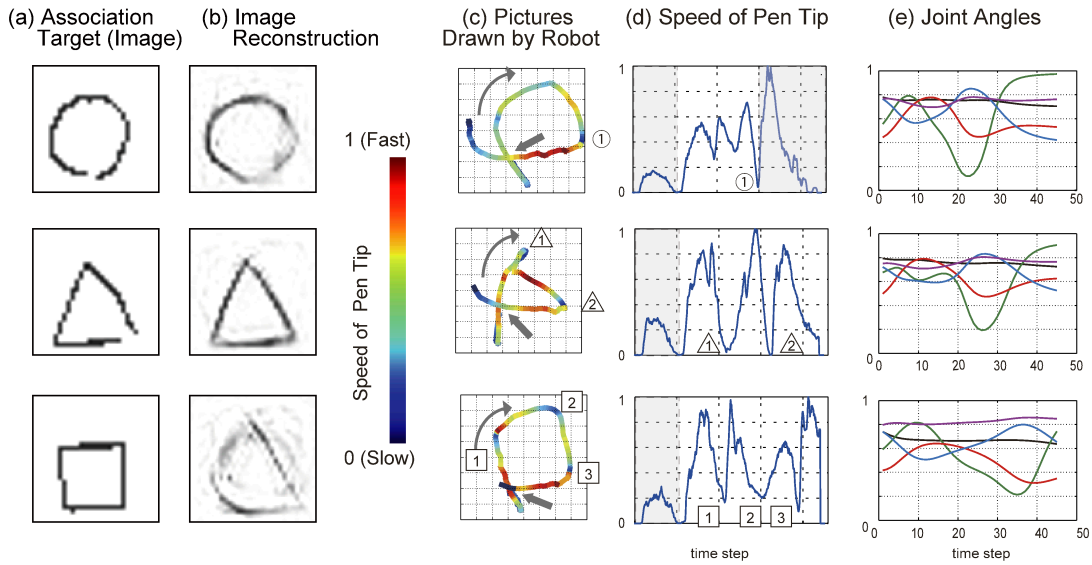


Figure 5.17: The association drawing results of not-trained picture images. (a) Not-trained images as the depiction target. (b) Reconstructed image (a) obtained by the trained DNN autoencoder. The robot's drawing results from the explored initial state of the slow, hidden layer are given in (c), (d), and (e).

(a) represents the final state of the selected sequences. Figure 5.16 (b) shows an image of (a) reconstructed by the trained DNN autoencoder. The CTRNN model generated drawing sequences by using each of the obtained initial states of the slow, hidden layer (CS). Figure 5.16 (c) presents lines that were drawn by the robot. Each line is colored by the speeds of the pen tip. This speed was calculated by the recorded positions of the pen during the drawing process. The value of the speed is shown in Fig. 5.16 (d). The joint angle sequence to draw lines of Fig. 5.16 (c) is shown in Fig. 5.16 (e). The shape of the drawn pictures maintains the shape characteristics of the training pictures given in Fig. 5.16 (a).

Associating drawing motion from an image

For obtaining drawing results using the initial state obtained by the explorations, we allowed the trained CTRNN to generate 45 step sequences. The initial input x_1 was given to the joint angles of the initial pose and the image feature at the first step of the drawing process. The joint angles of the initial pose were recorded when the robot's arm was set to place the pen at the average of the initial points

of the training dataset.

Figure 5.17 describes the association drawing results. In contrast to the results of the trained pictures, the pictures drawn by the exploration had distortions at the edge points. Also, all the lines did not stop at the starting points. We assume that these drawing errors were caused by the characteristics of CTRNN. For drawing motions to depict the shapes with edges, CTRNN is required to generate discontinuous sequences; however, the state of CTRNN's neural activity changes continuously. The mismatch between the start and the end points was also caused for the same reason. The pen was moved in the opposite direction at the beginning of the process. To reach the start point from the initial position also involved a discontinuous change of the joint angle trajectory.

Although the drawn lines have the above-mentioned errors, the associated drawing motions have the same characteristics of the pen tip's speed as Figure 5.16. The pen tip moves slowly around the corners of the shapes. When the model associated a motion to draw the circle, the pen stagnated at the right side of the drawn picture because the learned circle's drawing motion sequences were shorter than the other sequences.

Visualizing Feature of Learnt Sequences

Figure 5.18 shows the visualization results of the image features obtained by the trained DNN autoencoder. To project the ten-dimensional image feature vector to the 3D space, we use PCA. The axes PC1-PC3 in the figure correspond to the three principal components whose contribution values are larger than the values of other components. Each point of the plot indicates a picture image frame. We found that the sequence of image features for a single drawing process form a line in the projected space. Further, all these lines shared a common starting point because the image at the initial time step was the same, that is, white images.

We also visualize the acquired visuomotor features in the CS layer of the trained CTRNN model, as shown in Figure 5.19. The PCA was used to visualize the features. The components were selected in the same manner as in the Figure 5.18. The plot in the figure corresponds to the state of the CS hidden layer at each time step. For the trained DNN, the projected features of the trained CTRNN

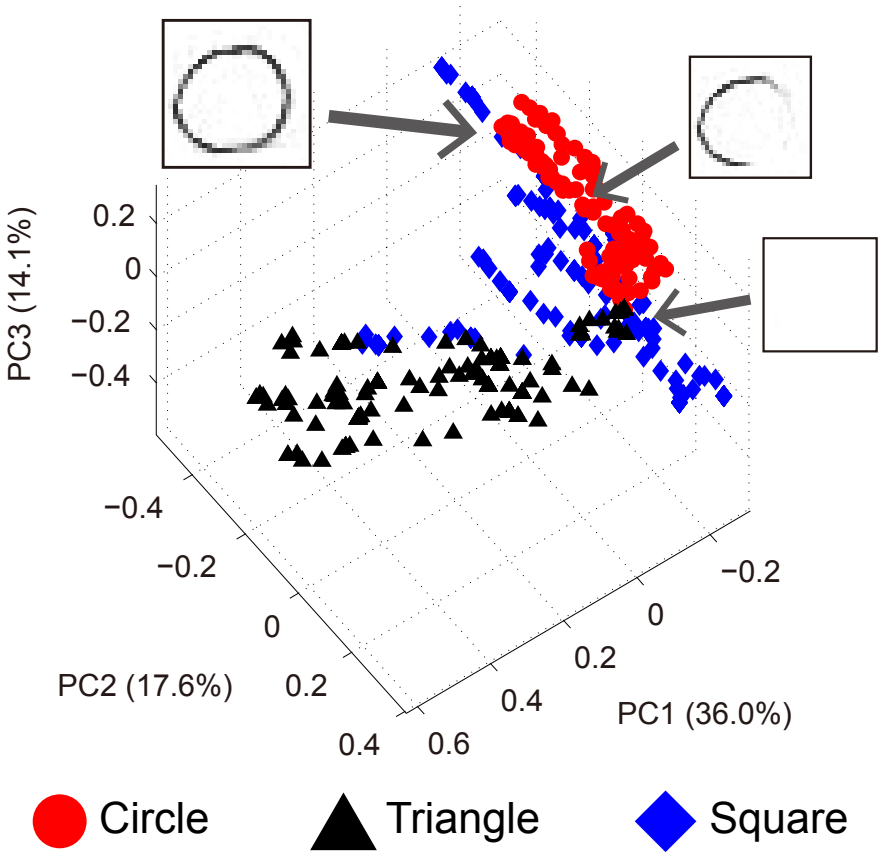


Figure 5.18: Visualization of image features by the DNN autoencoder (Training dataset). PC1-PC3 are the principal components that mark the three highest contribution values.

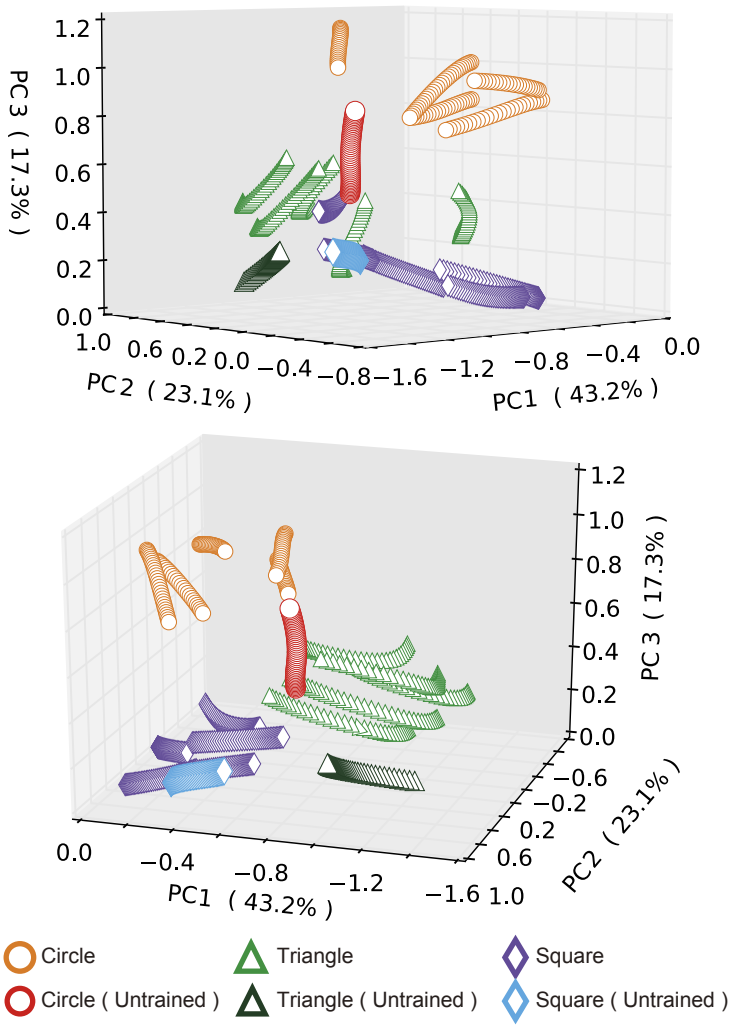


Figure 5.19: Values of CS when the trained CTRNN model generated a visuomotor sequence from the trained and explored initial states. The two figures share the same features but are shown from different viewing angles.

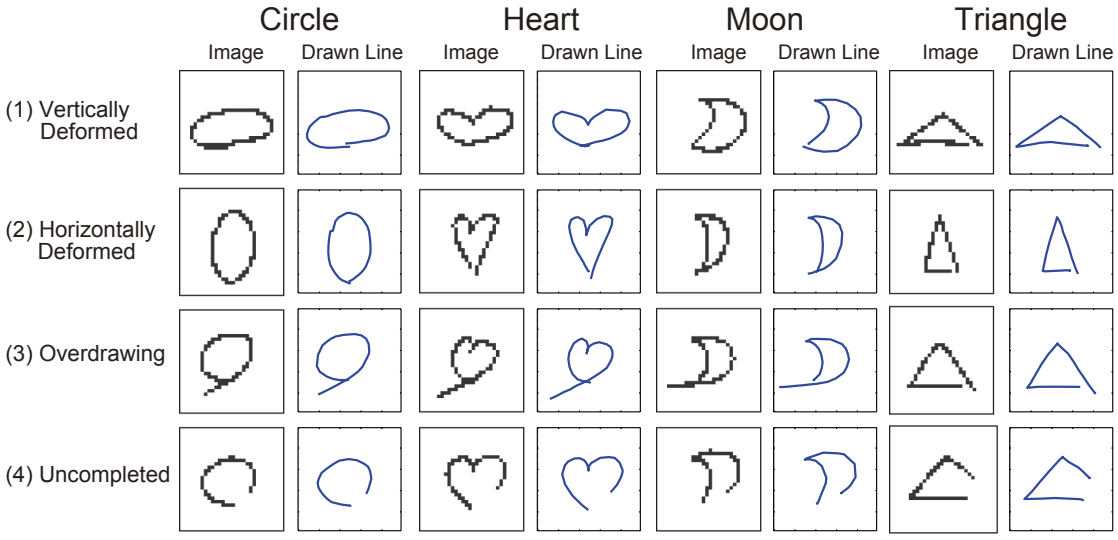


Figure 5.20: The training dataset of the distorted shapes. Four shapes are shown: a circle, a heart shape, a moon, and a triangle. These shapes are distorted in the four variations.

also form a line for each drawing process. Unlike the case for the DNN, the initial points of these lines were not averaged because they were allowed to be changed during the training process.

5.3.4 Experiments on Distorted Shape Recognition

In the previous experiments, the proposed models were required to remember 15 pictures without any explicit distortions. The experiment described here corresponds to the case of learning distorted shapes, as shown in Figure 5.20. In this experiment, we focus on investigating the possibility of replicating a human's recognition ability using the robot's drawing experiments.

As the training dataset, we collected 16 sequences having four sequences for each shape. The variations of each shape were determined by the degree of distortions. The vertically and horizontally deformed shapes corresponded to the size variations. The other distortion types were temporally deformed shapes. As in the previous experiment, the robot drew the shapes in a clockwise direction with a single stroke, and the starting points were mostly shared. The length of the collected sequences was from 30 to 40 steps. The size of the dataset and the

Table 5.5: The number of collected images and parameters of the proposed models. IO-CF-CS and DIMS give the dimensions of the layers in CTRNN and DNN, respectively. DATA means the number of images recorded as the training dataset. TRANS and ROTATED refer to the number of translated versions and the number of rotated versions of the originally recorded images, respectively. Training Iter specifies the number of training iterations.

Param	DNN	CTRNN
IO	900	13($\tau^X = 1$)
DIMS	900-400-180-80-30-8	30(CF, $\tau^F = 3$), 5(CS, $\tau^S = 30$)
DATA	631	631
TRANS	40384	-
ROTATED	3786	-
Training Iter	100	15000

hyperparameters of the models are described in Table 5.5.

Comparison of Shape Features

To evaluate the recognizing ability for the distorted shapes, we compared the distribution of the picture's features in several cases. The hypothesis for this comparison was that considering visuomotor data would lead to higher recognition accuracy as compared with using only the image data to discriminate. To measure the contribution to the picture-type recognition, we used PCA. The discrimination ability was measured by changing the input to PCA analysis. The comparison inputs are summarized as follows:

- IMG-RAW: Raw pixel values
- IMG-DNN: Image features dimensionally compressed by the trained DNN
- IMG-MOT-CTRNN: Initial state of the CTRNN model.

"RAW" in Fig. 5.21 indicates the use of pixel values of the picture to discriminate. These pixel values are obtained only from the final step of the drawing process, that is, the drawing process is not considered. The input in this case includes the translated and rotated images for generalizing the spatial variations and the

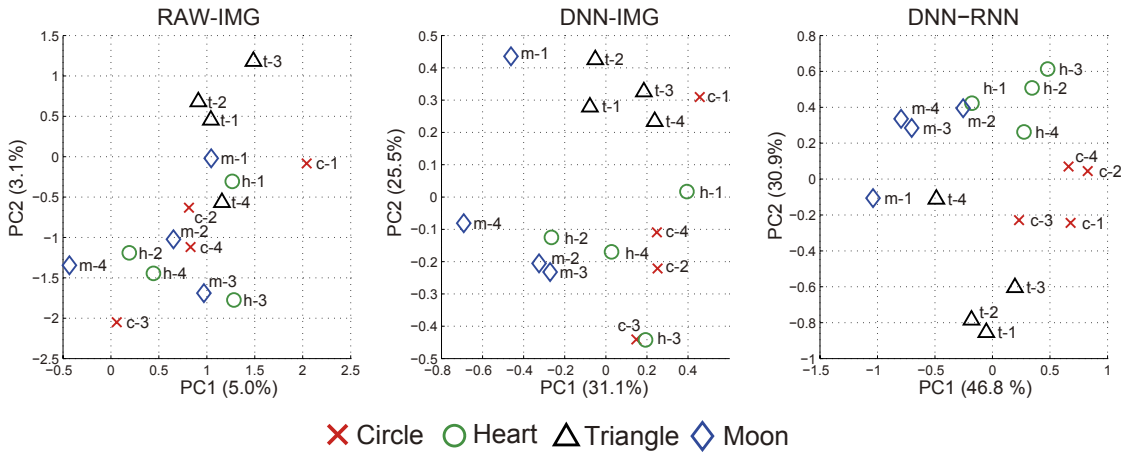


Figure 5.21: The results of PCA analysis. PC1 and PC2 indicate the principal components whose contribution values are the highest (The values are noted on the axes). The labels of the plots correspond to the shapes in Figure 5.20, for example, m-1 indicates the moon ("m") drawn in the vertically deformed manner ("1").

training dataset in the previous experiment. IMG-DNN corresponds to the use of the image features by a DNN autoencoder trained to use the images of IMG-RAW. Finally, IMG-MOT-CTRNN means the use of both the DNN autoencoder and CTRNN models. In this case, the PCA's input is the initial state of the CS hidden layer of the CTRNN that was trained by using the robot's drawing action, which corresponded to the image feature from the trained DNN autoencoder (IMG-DNN).

Figure 5.21 shows the results of the PCA analysis that changes the input. In each case, we chose the two principal components with the largest contribution values. Each plot indicates the picture in Figure 5.20. The projected features of IMG-DNN are organized as shapes that are better than IMG-RAW. However, a few pictures are considered to have shapes similar to IMG-DNN. For example, c-3 and h-3 are located close together. Unlike the IMG-DNN features, the CTRNN features form clusters based on the similarity of shapes. Although the analysis of IMG-MOT-CTRNN provided feature structures that were easier to discriminate than other cases, there were still paired features of different shapes. We found that these features were paired because of the similarity of the joint angle sequences learned by the CTRNN model whose values are shown in Figure 5.22.

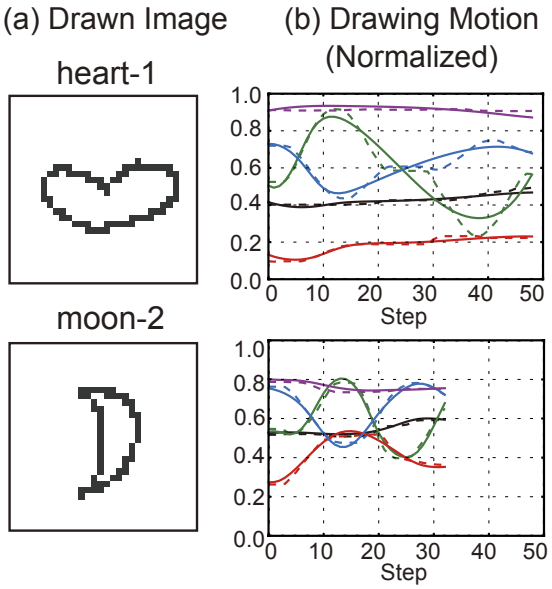


Figure 5.22: Generated drawing sequence of the distorted shapes. (a) Recorded image at the end of the drawing process. (b) Normalized joint angles. The dotted lines are the angles recorded by direct teaching. Solid lines correspond to angles generated by the trained CTRNN model.

Table 5.6: Class covariances of features by the comparison inputs.

	s_w	s_b	S
IMG-RAW	0.18	0.01	0.05
IMG-DNN	0.19	0.03	0.17
IMG-MOT-CTRNN	**0.19**	**0.11**	**0.56**

To quantitatively evaluate how these features form clusters based on the various shapes, the ratio of the covariances were used along with the analysis given in the last chapter (See Equation 4.2.) In this case the class corresponds the type of shape. The obtained ratios are described in Table 5.6. The covariance S of IMG-MOT-CTRNN is large in all cases. In particular, the between-class covariances s_b, which indicate the degree of separation between the types of shapes contribute to these differences.

5.3.5 Discussion for Robot Experiments

Using the two experiments involving drawing robots, the proposed robot-drawing system demonstrated visuomotor adaptation ability for associating the motions from a picture image and recognized shapes. In the experiment on learning simple shapes, the proposed CTRNN model demonstrated the ability to remember the training visuomotor sequences of 15 pictures. Further, the association results of drawing for the not-trained picture images suggested that the proposed system changed its behavior to generate drawing motions that could produce the given picture images. However, the pictures drawn by the associated motions had several distortions in the corners because of the CTRNN characteristics. One solution for these distortions was to make the drawing sequence longer. The fitting accuracy by CTRNN was limited by not only the time constant values but also by the time resolution of the learned sequences. Longer sequences enabled the pen to stop at a corner, change direction, and then proceed.

In the second experiment on the distorted shapes, the proposed system demonstrated the best recognition performance in the case of input data modality. This result suggested that the drawing experience contributed not only to the drawing ability but also to the recognition ability of shapes. Specifically, the memory of the drawing experience can be reused for recognition, which is a human's cognitive functions related to drawing or lettering [88, 11, 12, 90]. Even though the proposed system replicates the functionality of human's visuomotor memory, the replication is limited. For example, the robot experiment confirmed the functionality of the system only in the case of the black-and-white images that use a single stroke. Further, there were many factors pertaining to the shape, which

were not considered, such as the size, the direction (clockwise or anti-clockwise), the location, and the combination of shapes. To enhance the proposed system to consider these factors also, the capacity of the learners should be improved.

5.4 Chapter Conclusion

In this chapter, we introduced RNN-based models to enable the adaptation of drawing behavior by reusing the visuomotor memory of drawing. To add memory to the neural network, the RNN was designed to generate multiple visuomotor sequences of drawing processes. The drawing process consisted of the drawing action status and a corresponding image of the drawn picture. The adaptation of the drawing behavior reusing the obtained memory was implemented by exploring the RNN's initial state that determined the generated sequence for closed generation. The exploration was implemented by using the gradient descent method to optimize a new initial state that could lead the model to produce the desired sequence.

The proposed visuomotor adaptation method was confirmed for both the simulator and the real-robot environments. In the simulator experiments, we demonstrated that the LSTM-based model could remember hundreds of visuomotor sequences for drawing simple pictures. The association experiments suggested that the proposed model could generate the entire drawing process from the picture images in several scenarios: a completed picture image, a completed picture image with the first line given, and an image sequence of not-trained line ordering. Furthermore, the contribution of the picture image was confirmed in the comparison with another RNN without visualization.

In the robot experiments, the proposed model was applied to a robot environment. The drawing action state was replaced by the joint angle vector. A CTRNN-based model demonstrated the memorizing capacity of the visuomotor sequence by learning 15 pictures. The association ability was confirmed with a completed picture image. The final experiment did not focus on the association to draw, but on recognizing pictures. Consequently, we confirmed that considering visuomotor memory for drawing led to better performance in recognizing the

distorted shapes.

Chapter 6

Conclusion

6.1 Contribution of this Study for Understanding Drawing

The aim of this study was to understand a human's drawing abilities by constructing computational systems. The focused aspect of these abilities was the diversity of representations regarding the concepts depicted in pictures. In the case of visually recognizing hand-drawn sketches, the recognizer was required to share concepts between a hand-drawn sketch and photorealistic images. Also, hand-drawn sketches were produced by the drawer's body motion that could have many variations because of the differences in drawing styles or the low reproducibility of bodily motions. This study investigated these abilities based on the approach used in cognitive developmental robotics. The computational systems were constructed to replicate the two focused abilities: recognition and drawing. These abilities were described by their requirements. Recognition is the ability to recognize hand-drawn sketches and photorealistic images by a shared visual processing function. Drawing indicates the ability to produce bodily motions that could alter the drawn picture into a given target picture by accepting visual feedback from the picture. As the fundamental requirement, these systems were required to be constructed based on a limited amount of prior knowledge of the pictures.

This thesis proposed computational systems for recognition and drawing. These

systems were implemented by using deep learning and a robot. The recognition system uses the image classifier of a CNN. A CNN is a deep learning model whose functionality does not require explicit knowledge of pictures. In this study, we propose to include sketch images to enable a CNN to discriminate both a sketch and a photo in contrast to the existing CNN models whose recognition target is limited to a single image type. The CNN model was trained to classify illustration images, photos, and color-converted versions. Experiments in the classification of 20 class animal images showed the contribution of illustration images in the training dataset.

The proposed drawing systems were also implemented by using DNN models. To enable the generation of sequential data, RNNs were used. This thesis proposed a RNN model for organizing visuomotor memory of the drawing process. The drawing process involves the robot's motion and images and uses visual feedback from the drawn picture. The RNN is trained to predict the drawing motion of the joint angles robot and the image information of the drawn picture. The drawing ability was realized by an adaptation of the acquired memory. The behavior of trained RNN can be changed by the optimization process to minimize the difference between the prediction and the depiction target. In the simulator experiments, the proposed drawing system demonstrated the ability to associate the motions from an image. The association ability was also demonstrated in the robot experiments for learning simple shapes. The proposed RNN model generated images similar to a humanoid's drawing motion from an image. Also, other experiments on distortions suggested that the visuomotor memory contributes to the recognition of shapes by reusing drawing experiences.

The proposed systems of drawing and recognition demonstrate the aspects of drawing ability suggested by cognitive science. The contribution of the data argumentation method shows that adding professional drawings (i.e., illustrations) leads a CNN classifier to improve the recognition accuracy of the sketch and photo images. This result reflects the influence of professional drawings in children's drawing development. The experiments using the drawing systems suggested that a visuomotor memory of the robot's drawing can be acquired by the RNN model. The visuomotor memory acquired in these systems enabled the robot to associate

the drawing motion from an image and to recognize pictures by considering the drawing experiences. These results of visuomotor memory corresponded to the phenomena indicated by experimental psychology: the ability to generate motor activities to depict another picture and the use of dynamic information to produce pictures in picture recognition.

6.2 Limitation and Future Studies

One limitation of the proposed systems is the variations in the pictures. This study assumed that drawn pictures are black and white images that have a few lines. The property of lines is fixed in contrast to pictures drawn by different types of tools, such as brushes, pencils, or pens. The proposed recognition system could discriminate any type of image as long as these images could be given as raster images. A technical problem to improving the recognition system was the size of the dataset. However, as mentioned in Section 4, the existing dataset of the picture or the illustration was smaller than the dataset of the photo images. The drawing system may also require many drawing experiences if we wish to extend the drawing tools. Further, we did not consider erasing any part of the drawn picture. In this case, the transition of the tools used needs to be discussed.

Another limitation is the variety of depicted concepts. In the experiments on image recognition and drawing, the pictures were intended to represent a single concept. However, we could depict complex concepts that included some concepts. For example, a scenery would include many objects, such as a house, a tree, and humans. In addition, sometimes the drawer required to repeat the drawing of windows of a house. To recognize or depict complex visual concepts, the idea of primitives may be considered. In fact, drawing is considered as the ability to construct visual concepts [7]. One way enhancing the recognition target is to consider that these complex pictures will be discriminated not as a probability of a single category, but as sentences (e.g., a cup is on the desk). The drawing system will require decomposition and composition of drawing experiences. In this case, the drawing model is required to select primitives of visuomotor memory obtained by drawing experiences and compose them by considering the description of what

the model intends to draw.

Finally, one aspect that future studies can consider is a combination of the recognition and drawing systems. This combination will realize the depiction of photorealistic images. A possible implementation of this combination is to use the feature of the input photo image in the intermediate layer of CNN in the recognition system. This feature can be input to RNN in the drawing system as an initial state of the generation process. The acquisition of the functionalities to process the photo image and generate corresponding drawing motion will require a dataset of the paired photos and the robot's drawing processes.

Bibliography

[1] N. Humphrey, "Cave art, autism, and the evolution of the human mind," *Cambridge Archaeological Journal*, vol. 8, no. 2, pp. 165–191, 1998.

[2] J. Piaget, *Play, dreams and imitation in childhood*. Routledge and Kegan Paul, 1951.

[3] G.-H. Luquet, *Le Dessin Enfantin*. 1927.

[4] L. S. Vygotsky, *Mind in society*. Combridge MA: Harverd University Press, 1978.

[5] P. V. Sommers, "A system for drawing and drawing-related neuropsychology," *Cognitive Neuropsychology*, vol. 6, pp. 117–164, Mar. 1989.

[6] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robotics and Autonomous Systems*, vol. 37, pp. 185–193, Nov. 2001.

[7] S. McCrea, "A neuropsychological model of free-drawing from memory in constructional apraxia: A theoretical review," *American Journal of Psychiatry and Neuroscience*, vol. 2, no. 5, p. 60, 2014.

[8] F. Lacquaniti, C. Terzuolo, and P. Viviani, "The law relating the kinematic and figural aspects of drawing movements," *Acta Psychologica*, vol. 54, pp. 115–130, Oct. 1983.

[9] Y. Wada and M. Kawato, "A theory for cursive handwriting based on the minimization principle," *Biological Cybernetics*, vol. 73, pp. 3–13, June 1995.

[10] H. Fu, S. Zhou, L. Liu, and N. J. Mitra, "Animated construction of line drawings," *ACM Transactions on Graphics*, vol. 30, no. 6, p. 1, 2011.

[11] M. K. Babcock and J. J. Freyd, "Perception of dynamic information in static handwritten forms," *The American journal of psychology*, vol. 101, no. 1, pp. 111–130, 1988.

[12] A. Pignocchi, "How the intentions of the draftsman shape perception of a drawing," *Consciousness and cognition*, vol. 19, no. 4, pp. 887–898, 2010.

[13] C. Pedretti, *Leonardo Da Vinci on Painting. A Lost Book (Libro A)*. University of California Press, 1964.

[14] H. von Helmholtz, *Science and Culture: Popular and Philosophical Essays*. University of Chicago Press, 1995.

[15] J. J. Gibson, *The Ecological Approach to Visual Perception*. Psychology Press, 1986.

[16] J. J. Gibson, *The Senses Considered as Perceptual Systems*. Praeger. Revised ed. edition, 1983.

[17] E. H. Gombrich, *Art and Illusion: A Study in the Psychology of Pictorial Representation*. Phaidon Press, 1960.

[18] J. Piaget and B. Inhelder, *The child's conception of space*. Routledge and Kegan Paul, Rondon, 1956.

[19] D. J. Cohen and S. Bennett, "Why can't most people draw what they see?," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 23, no. 3, pp. 609–621, 1997.

[20] S. Roncato, G. Sartori, J. Masterson, and R. Rumiati, "Constructional apraxia: An information processing analysis," *Cognitive Neuropsychology*, vol. 4, no. 2, pp. 113–129, 1987.

[21] F. Goodenough, *Measurement of Intelligence by Drawings*. New York: Word Books, 1926.

[22] D. F. Benson and M. I. Barton, "Disturbances in constructional ability," *Cortex*, vol. 6, no. 1, pp. 19–46, 1970.

[23] L. Trojano, D. Grossi, and T. Flash, "Cognitive neuroscience of drawing: Contributions of neuropsychological, experimental and neurofunctional studies," *Cortex*, vol. 45, pp. 269–277, Mar. 2009.

[24] V. S. Ramachandran, *The Emerging Mind: The BBC Reith Lectures 2003*. Profile Books(GB), 2003.

[25] C. Di Dio and G. Vittorio, "Neuroaesthetics: a review," *Current Opinion in Neurobiology*, vol. 19, no. 6, pp. 682–687, 2009.

[26] D. Marr, *Vision*. W.H. Freeman and Campany, 1982.

[27] M. J. Farah, "The neurological basis of mental imagery: A componential analysis," *Cognition*, vol. 18, no. 1, pp. 245–272, 1984.

[28] H. Hicheur, S. Vieilledent, M. J. E. Richardson, T. Flash, and A. Berthoz, "Velocity and curvature in human locomotion along complex curved paths: A comparison with hand movements," *Experimental Brain Research*, vol. 162, no. 2, pp. 145–154, 2005.

[29] I. E. Sutherland, "Sketch pad a man-machine graphical communication system," in *Proceedings of the SHARE Design Automation Workshop*, DAC '64, (New York, NY, USA), pp. 6.329–6.346, ACM, 1964.

[30] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 1624–1636, Nov. 2011.

[31] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–10, 2012.

[32] S. Saga, "A freehand interface for computer aided drawing systems based on the fuzzy spline curve identifier," in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 3, pp. 2754–2759, Oct. 1995.

[33] Y. Wang, Y. Chen, J. Liu, and X. Tang, "3d reconstruction of curved objects from single 2d line drawings," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1834–1841, 2009.

[34] J. Chu, M. T. Gao, G. M. Zhang, and R. N. Feng, "Line-based optimization for 3d object reconstruction from single line drawings," *Applied Mechanics and Materials*, vol. 20-23, pp. 910–915, Jan. 2010.

[35] S.-Z. L. S.-Z. Liao, X.-J. W. X.-J. Wang, and J.-L. L. J.-L. Lu, "An incremental bayesian approch to sketch recognition," in *2005 International Conference on Machine Learning and Cybernetics*, vol. 7, pp. 18–21, Aug. 2005.

[36] C. Calhoun, T. F. Stahovich, T. Kurtoglu, and L. B. Kara, "Recognizing multi-stroke symbols," in *AAAI Spring Symposium on Sketch Understanding*, pp. 15–23, 2002.

[37] H. Li, H. Shao, J. Cai, and X. Wang, "Hierarchical primitive shape classification based on cascade feature point detection for early processing of on-line sketch recognition," in *2010 International Conference on Computer Engineering and Technology, Proceedings*, vol. 2, pp. 397–400, 2010.

[38] G. Orbay and L. B. Kara, "Beautification of design sketches using trainable stroke clustering and curve fitting," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 5, pp. 694–708, 2011.

[39] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, Nov. 1986.

[40] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *9th European Conference on Computer Vision* (A. Leonardis, H. Bischof, and A. Pinz, eds.), (Berlin, Heidelberg), pp. 428–441, Springer Berlin Heidelberg, 2006.

[41] H. Cohen, *The First Artificial Intelligence Coloring Book*. Addison-Wesley Pub, 1983.

[42] B. Gooch and A. Gooch, *Non-photorealistic rendering*. A K Peters/CRC Press, May 2001.

[43] A. Hertzmann, "A survey of stroke-based rendering," *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 70–81, 2003.

[44] A. Wanner, "Building the plotter: an aesthetic exploration with drawing robots," in *International Symposium on Computational Aesthetics*, vol. 1, pp. 39–46, 2011.

[45] O. Deussen, T. Lindemeier, S. Pirk, and M. Tautzenberger, "Feedback-guided stroke placement for a painting machine," *Computational Aesthetics in Graphics, Visualization, and Imaging*, pp. 25–33, 2012.

[46] S. Mueller, N. Huebel, M. Waibel, and R. D'Andrea, "Robotic calligraphy - learning how to write single strokes of chinese and japanese characters," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1734–1739, IEEE, Nov. 2013.

[47] S. Kudoh, K. Ogawara, M. Ruchanurucks, and K. Ikeuchi, "Painting robot with multi-fingered hands and stereo vision," *Robotics and Autonomous Systems*, vol. 57, pp. 279–288, Mar. 2009.

[48] H. M. L. Josh and Y. Yam, "Stroke trajectory generation experiment for a robotic chinese calligrapher using a geometric brush footprint model," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (St. Louis, USA), pp. 2315–2320, 2009.

[49] V. Mohan, P. Morasso, J. Zenzeri, G. Metta, V. S. Chakravarthy, and G. Sandini, "Teaching a humanoid robot to draw ' shapes'," *Autonomous Robots*, vol. 31, pp. 21–53, July 2011.

[50] V. S. Chakravarthy and B. Kompella, "The shape of handwritten characters," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1901–1913, 2003.

[51] N. Xie, H. Hachiya, and M. Sugiyama, "Artist agent: A reinforcement learning approach to automatic stroke generation in oriental ink painting," *IEICE Transactions on Information and Systems*, vol. E96.D, pp. 1134–1144, June 2013.

[52] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, Dec. 1943.

[53] M. Minsky and S. Papert, *Perceptron*. MIT Press, 1969.

[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988.

[55] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[56] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, pp. 1–9, 2012, 1102.0183.

[57] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504–507, July 2006, 20.

[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ArXiv e-prints*, 2015, arXiv:1502.03167.

[59] G. Hinton, "Dropout : A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 1929–1958, 2014.

[60] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudk, eds.),

vol. 15 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 315–323, PMLR, 11–13 Apr. 2011.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *ArXiv e-prints*, Dec. 2015, arXiv:1512.03385.

[62] Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, and T. Hospedales, "Sketch-a-net that beats humans," *ArXiv e-prints*, pp. 1–11, Jan. 2015, arXiv:1501.07873.

[63] R. K. Sarvadevabhatla, J. Kundu, and V. B. R, "Enabling my robot to play pictionary: Recurrent neural networks for sketch recognition," in *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, (New York, NY, USA), pp. 247–251, ACM, 2016.

[64] O. Seddati, S. Dupont, and S. Mahmoudi, "Deepsketch: Deep convolutional neural networks for sketch recognition and similarity search," in *13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 1–6, IEEE, June 2015.

[65] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database," *ACM Transactions on Graphics*, vol. 35, pp. 1–12, July 2016.

[66] V. Turchenko, E. Chalmers, and A. Luczak, "A deep convolutional auto-encoder with pooling - unpooling layers in caffe," *ArXiv e-prints*, 2017, arXiv:1701.04949.

[67] L. Zhang, L. Lin, X. Wu, S. Ding, and L. Zhang, "End-to-end photo-sketch generation via fully convolutional representation learning," in *5th ACM on International Conference on Multimedia Retrieval*, pp. 627–634, ACM Press, Oct. 2015.

[68] E. Simo-serra, "Learning to simplify : Fully convolutional networks for rough sketch cleanup," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–11, 2016.

[69] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *ArXiv e-prints*, pp. 1–14, Nov. 2015, arXiv:1511.05440.

[70] L. a. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *ArXiv e-prints*, pp. 3–7, Aug. 2015, arXiv:1508.06576.

[71] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *ArXiv e-prints*, pp. 1–9, 2014, arXiv:1406.2661.

[72] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *ArXiv e-prints*, pp. 1–16, 2015, arXiv:1511.06434.

[73] E. Simo-Serra, S. Iizuka, and H. Ishikawa, "Mastering sketching: Adversarial augmentation for structured prediction," *ACM Transactions on Graphics*, vol. 37, pp. 11:1–11:13, Jan. 2018.

[74] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel distributed processing: explorations in the microstructure of cognition*, pp. 318–362, MIT Press Cambridge, MA, USA, 1986.

[75] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[76] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[77] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," *ArXiv e-prints*, 2014, arXiv:1411.4555.

[78] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *ArXiv e-prints*, 2016, arXiv:1602.01783.

[79] D. Ha and D. Eck, "A neural representation of sketch drawings," *ArXiv e-prints*, pp. 1–20, 2017, arXiv:1704.03477.

[80] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, arXiv:1412.6980.

[81] K. Mochizuki, S. Nishide, H. G. Okuno, and T. Ogata, "Developmentalhuman-robot imitation learning of drawing with a neuro dynamical system," in *2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013*, pp. 2336–2341, 2013.

[82] S. Nishide, K. Mochizuki, H. G. Okuno, and T. Ogata, "Insertion of pause in drawing from babbling for robot's developmental imitation learning," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Hong Kong, China), pp. 4785–4791, 2014.

[83] M. Amenomori, A. Kono, J. S. Fournier, and G. A. Winer, "A cross-cultural developmental study of directional asymmetries in circle drawing," *Journal of Cross-Cultural Psychology*, vol. 28, no. 6, pp. 730–742, 1997.

[84] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, 2009.

[85] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy, "Open-images: A public dataset for large-scale multi-label and multi-class image classification," 2017.

[86] B. Wilson, "The artistic tower of babel: Inextricable links between culture and graphic development," *Visual Arts Research*, vol. 11, no. 1, pp. 90–104, 1985.

[87] M. V. Cox, J. Perara, M. Koyasu, and H. Hiranuma, "Children's human figure drawings in the uk and japan: The effects of age, sex and culture," *British Journal of Developmental Psychology*, vol. 19, no. 2, pp. 275–292, 2001.

[88] J. J. Freyd, "Representing the dynamics of a static form," *Memory & Cognition*, vol. 11, no. 4, pp. 342–346, 1983.

[89] J. Parkinson and B. Khurana, "Temporal order of strokes primes letter recognition," *Quarterly journal of experimental psychology (2006)*, vol. 60, no. 9, pp. 1265–1274, 2007.

[90] A. H. Waterman, J. Havelka, P. R. Culmer, L. J. B. Hill, and M. Mon-Williams, "The ontogeny of visual-motor memory and its importance in handwriting and reading: a developing construct," *Proceedings of the Royal Society B: Biological Sciences*, vol. 282, pp. 20140896–20140896, Nov. 2014.

[91] L. Bottou, *Stochastic Gradient Descent Tricks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[92] Y. Bengio and P. Frasconi, "Diffusion of context and credit information in markovian models," *Journal of Artificial Intelligence Research*, vol. 3, pp. 249–270, 1995.

[93] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *ArXiv e-prints*, 2014, arXiv:1408.5093.

[94] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262–282, 2007.

[95] Y. Cao, H. Wang, C. Wang, Z. Li, L. Zhang, and L. Zhang, "Mindfinder: interactive sketch-based image search on millions of images," in *ACM Multimedia Conference*, p. 1605, 2010.

[96] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.*, vol. 40, pp. 5:1–5:60, May 2008.

[97] A. Krizhevsky and G. Hinton, "Using very deep autoencoders for content-based image retrieval," in *European Symposium on Artificial Neural Networks*, pp. 1–7, 2011.

[98] T. q. Peng and F. Li, "Image retrieval based on deep convolutional neural networks and binary hashing learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1742–1746, Mar. 2017.

[99] D. Varga and T. Szirnyi, "Fast content-based image retrieval using convolutional neural network and hash function," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2636–2640, Oct. 2016.

[100] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, (Washington, DC, USA), pp. 1386–1393, IEEE Computer Society, 2014.

[101] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive Science*, vol. 16, no. 3, pp. 307–354, 1992.

[102] R. Nishimoto and J. Tani, "Learning to generate combinatorial action sequences utilizing the initial sensitivity of deterministic dynamical systems," *Neural Networks*, vol. 17, no. 7, pp. 925–933, 2004.

[103] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proceedings of the Annual Conference of the International Speech Communication Association*, pp. 338–342, Jan. 2014.

[104] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Studies in Computational Intelligence and Complexity, Springer, 2012.

[105] O. Ronneberger, P. Fischer, and T. Brox, "U-net:convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.0, pp. 1–8, 2015, arXiv:1505.04597.

[106] A. Graves, "Generating sequences with recurrent neural networks," *ArXiv e-prints*, pp. 1–43, Aug. 2013, arXiv:1308.0850.

[107] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *ArXiv e-prints*, 2012, arXiv:1211.5063.

[108] "The quick, draw! dataset." github.com/googlecreativelab/quickdraw-dataset, (Date last accessed 2nd-December-2017).

[109] "Magenta." magenta.tensorflow.org, (Date last accessed 2nd-December-2017).

[110] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *ArXiv e-prints*, Nov. 2015, arXiv:1511.06349.

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

[112] J. Martens, "Deep learning via hessian-free optimization," in *27th International Conference on Machine Learning*, vol. 951, (Haifa, Israel), pp. 735–742, 2010.

[113] R. D. Beer, "On the dynamics of small continuous-time recurrent neural networks," *Adaptive Behavior*, vol. 3, no. 4, pp. 469–510, 1995.

[114] A. Robotics, "Nao humanoid," July 2015. http://doc.aldebaran.com/2-1/home_nao.html, (Date last accessed 12th-July-2015).

[115] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[116] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia Tools and Applications*, pp. 1–17, 2017.

[117] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[118] D. G. Lowe, "Object recognition from local scale-invariant features," in *Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999.

[119] Y. T. Zhou and R. Chellappa, "Computation of optical flow using a neural network," in *IEEE 1988 International Conference on Neural Networks*, vol. 2, pp. 71–78, July 1988.

[120] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[121] S. Harnard and S. Harnad, "The symbol grounding problem," *Physica D*, vol. 42, pp. 335–346, 1990.

[122] A. Chatterjee, "Neuroaesthetics: A coming of age story," *Journal of Cognitive Neuroscience*, vol. 23, no. 1, pp. 53–62, 2011.

Appendix A

Neural Networks

This chapter is written for readers who are not familiar with neural networks. First, this chapter introduces the basic idea of deep neural networks (DNNs). Then, several variations of neural network models are explained.

The idea of the neural networks originated from a model of the neuron made by McCulloch and Pitts in 1943 [52]. This idea led to more nonlinear models having many neurons that increased the ability to approximate a probability density function. The ability of the approximation depended on the number of parameters that could be changed by the optimization process. The number of parameters could be increased by stacking the calculation unit having learnable parameters and a nonlinear function. The model with these stacked units is called Feedforward Neural Network (FNN), which has "layers" as its unit.

Figure A.1 shows an FNN model with three layers including the input as the first layer. The output of this model h^2 is obtained by calculating the activation value of each layer as follows:

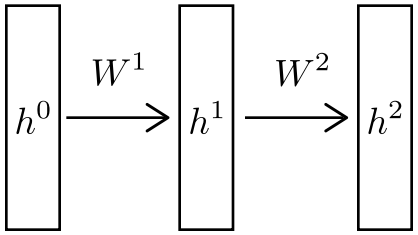


Figure A.1: A feedforward neural network model

$$h^1 = \sigma^1(W^1 \cdot h^0 + b^1) \tag{A.1}$$
$$h^2 = \sigma^2(W^2 \cdot h^1 + b^2), \tag{A.2}$$

where W^1 and W^2 are the matrices of the first and second layers, respectively; b^1 and b^2 are the vectors of the learnable parameters for the first and second layers, respectively. The activation value of the second layer h^1 is given by the nonlinear function σ^1 that accepts the linear combination of the input h^0. The output of the model corresponds to the activation value of the third layer whose activation process is that of the second layer. The learnable parameters $W and b$ are obtained by the optimization process that minimizes an *objective function* to measure how much the model can give the desired data. For example, the objective function is cross-entropy between the probability estimated by the model and the corresponding target data whose format is often given by the one-hot vector $(0, \cdots, 0, 1, 0, \cdots, 0)$. The optimization process is implemented by using the backpropagation method [74]. This method provides a way of calculating the derivation of the objective function with respect to each trainable parameters by applying the chain rule. For example, the derivation of W^2 is given as follows:

$$\frac{\partial L}{\partial W^2} = \frac{\partial L}{\partial h^2}\frac{\partial h^2}{\partial u^2}\frac{\partial u^2}{\partial W^2}, \tag{A.3}$$

where $u^2 = W^2 \cdot h^1 + b^2$ is the linear combination at the third layer, and L refers to the objective function. By using these derivations, each parameter will be updated by the following gradient descent:

$$\theta_{i+1} = \theta_i - \alpha\frac{\partial L}{\partial \theta_i}, \tag{A.4}$$

where θ_i indicates a parameter at the i th iteration step. The parameter is updated to decrease the objective function. The size of a step is controlled by a hyperparameter called the learning rate.

The backpropagation method allows us to stack more than three layers to increase the complexity of the approximated function. Honrik showed that FNNs could become universal approximators when there were enough layers [115]. An

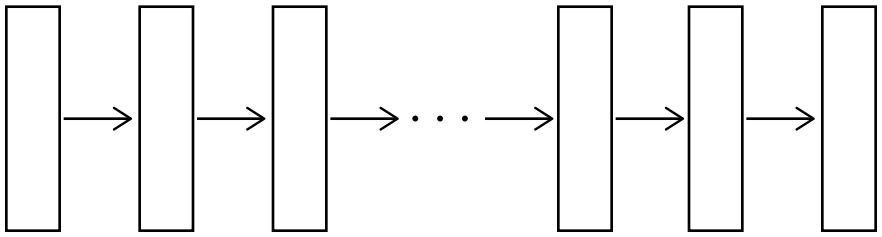


Figure A.2: A deep neural network model

FNN with many layers is known to have difficulty in optimization because of the instability of the gradient values during the backpropagation process [92]. Many methods have been proposed to improve the optimization efficiency of FNNs with many layers [58, 80, 116]. These methods could be investigated by using the power of the processor and the computational memory.

FNNs with many layers have recently been called DNNs [57, 117]. The appearance of DNNs enables developers to input large-dimensional data without any feature-extraction process. For example, conventional image classifiers acquired pre-designed algorithms to extract invariant features, such as edge information [118]. Instead of the extracted features, the DNN accepts normalized pixel values, and the feature extraction process is acquired through the optimization process [55, 56]. Also, a DNN can regress large-dimensional data so that it can behave as a generative model [72].

DNN for image generation or recognition attempts to employ sparse connectivity to process large-dimensional data by using few parameters. Convolution operations are used to calculate a linear combination of the input image data. A layer using the convolution operation is called "convolutional layer." The DNN using the convolutional layer is often called convolutional neural network (CNN). The calculation process of the convolutional layer is given as follows:

$$y = \sigma(W * x + b), \tag{A.5}$$

where $Wandb$ are trainable parameters used for the acquisition of a linear map of the input x; σ refers to the activation function to obtain the output y; and $*$ indicates the convolution operation. For example, $u = W * x$ for the two-dimensional image x is also the two-dimensional data whose value at (i, j) is

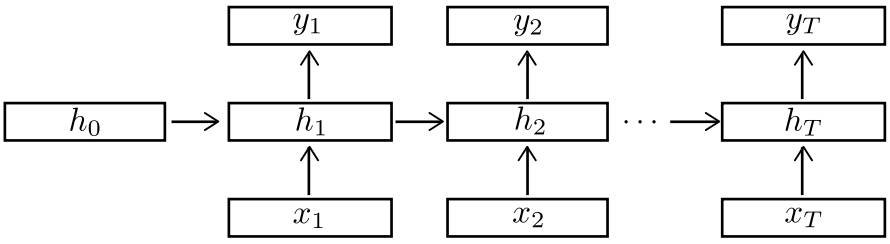


Figure A.3: A recurrent neural network model

given by

$$u(i,j) = \sum_m \sum_n W(m,n)x(i-m, j-n). \tag{A.6}$$

When the convolution operation is applied, it is possible to skip some indices of the spatial dimension of x. To apply every two pixels in each dimension of x means u becomes downsampled x. Downsampling is required when the output dimension is much smaller than the input. Another popular method to downsample the spatial dimensionality is "pooling," which attempts to choose the specific pixel value within the desired space [119]. Similar to defining downsampling by skipping indices, upsampling can also be defined. In this case, the order of the convolution operation is backward, thus $u = x * W$.

One solution to build learners of sequential data is to employ RNNs that can retain memory beyond the specific time steps [120]. To retain memory, RNN has feedback between its layers. A simple RNN model is depicted in Figure A.3. This model accepts the sequential input $x = (x_1, x_2, \cdots, x_t, \cdots x_T)$ to obtain the output y_t through the hidden layer and retains the memory as h_t. The output is given as follows:

$$h_t = \sigma^H(W^{HX} \cdot x_t + W^{HH} \cdot h_{t-1} + b^H) \tag{A.7}$$

$$y_t = \sigma^O(W^{YH} \cdot h_t + b^Y), \tag{A.8}$$

where W are the weight matrices, and σ refers to the activation functions. The state of h_t is decided not only by the input x but also by the value of h at the previous step. At the first step, h_0 can take any vector expression, but it is

usually given as a zero vector or as learnable parameters; h_0 is sometimes called the *initial state*. The behavior of RNN is sensitive to the initial states so that we can implement a variety of behaviors into a single model [102]. The learnable parameters of RNN are optimized by gradient descent and FNN and CNN.

Continuous time series data, such as a robot's motion, can be efficiently trained in the RNN model whose output value changes continuously. A continuous time-scale recurrent neural network (CTRNN) is a RNN whose hidden layer's state changes its internal state [113]. Internal state means the state before applying the activation function. The internal state of the CTRNN's hidden layer is given as follows:

$$\tau \dot{u} = -u + W^{HX} \cdot x + W^{HH} \cdot h + b^H. \tag{A.9}$$

By replacing $\dot{u}$ by $\frac{u_{t+1}-u_t}{\Delta t}$ and considering $\tau \leftarrow \frac{\tau}{\Delta t}, t \leftarrow t+1$, we obtain the following:

$$u_t = (1 - \frac{1}{\tau})u_{t-1} + \frac{1}{\tau}(W^{HX} \cdot x_t + W^{HH} \cdot h_{t-1} + b^H). \tag{A.10}$$

Then, the state of hidden layer h_t is calculated as follows:

$$h_t = \sigma^H(u_t). \tag{A.11}$$

The above-mentioned simple RNN can be considered as a specific case of CTRNN whose τ is one; τ functions as a time constant value that determines the response characteristics against the input's change. Figure A.4 shows the output of a CTRNN model whose weight matrix is one. If the value of the input sequence x_t changes suddenly at $t = 10$ from zero to one, the model's output gradually reaches one. The reaching speed is determined by *tau*. A small *tau* value provides a faster CTRNN response to the input.

Discontinuous sequences, such as the probability transition of words in sentences, attempt to be processed by RNN with the gating functions. The gating function was introduced to solve the vanishing gradient problem because of the feedback connection in its hidden layer [107]. The challenges of this problem led to the creation of many variations of RNNs. LSTM is a RNN that has the gating

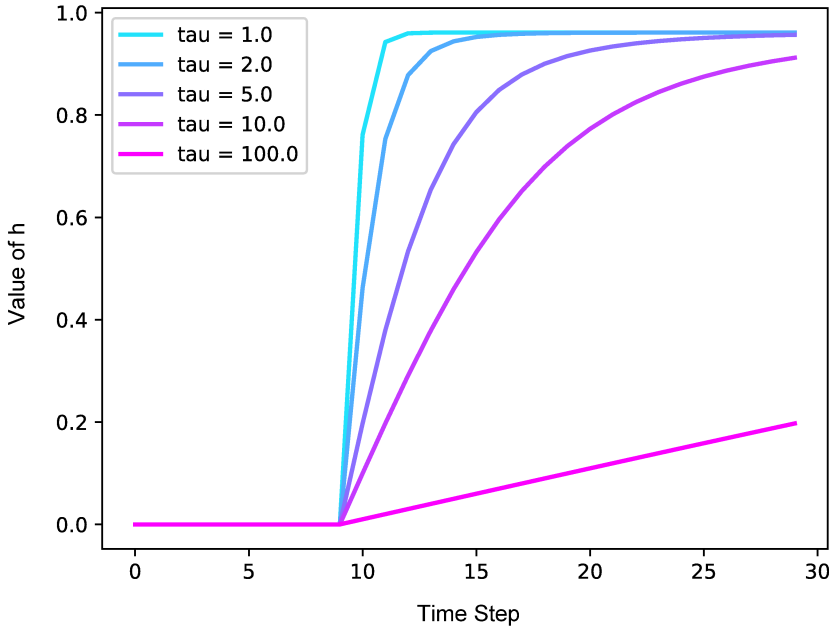


Figure A.4: The state of CTRNN by using various time constant values

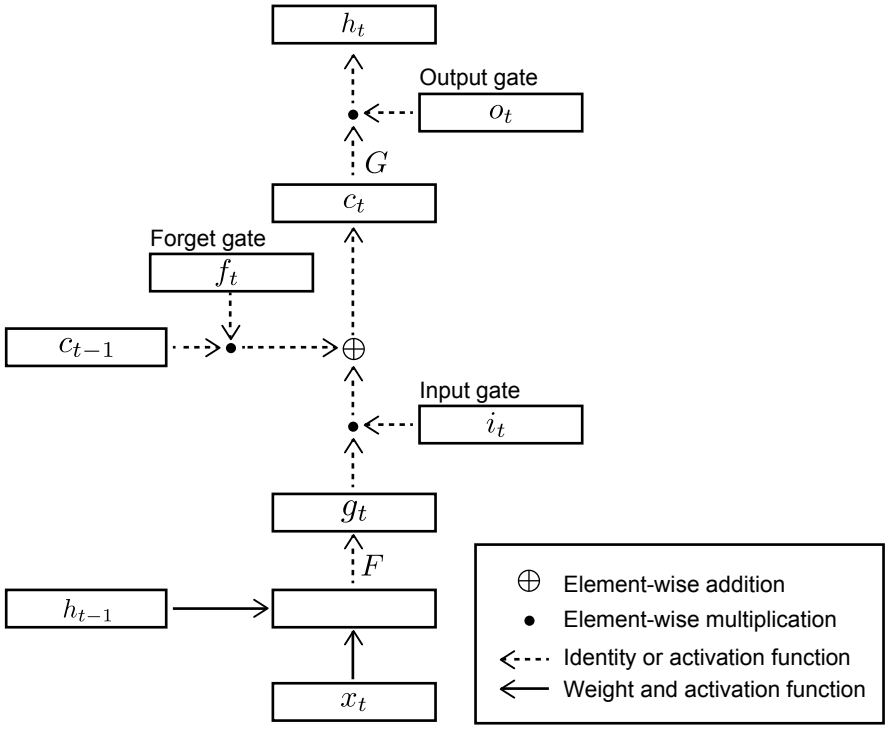


Figure A.5: Hidden layer of LSTM

function of controlling the gradient flow in the hidden layer [75].

Figure A.5 describes the forwarding process of LSTM's hidden layer. This layer gives h_t by applying the gating functions with trainable variables for the input x_t as follows:

$$
\begin{aligned}
i_t &= \sigma(W^{IX} \cdot x_t + W^{IH} \cdot h_{t-1} + b^I) && \text{(A.12)} \\
f_t &= \sigma(W^{FX} \cdot x_t + W^{FH} \cdot h_{t-1} + b^F) && \text{(A.13)} \\
o_t &= \sigma(W^{OX} \cdot x_t + W^{OH} \cdot h_{t-1} + b^O) && \text{(A.14)} \\
g_t &= F(W^{GX} \cdot x_t + W^{GH} \cdot h_{t-1} + b^G) && \text{(A.15)} \\
c_t &= f_t c_{t-1} + i_t g_t && \text{(A.16)} \\
h_t &= o_t F(c_t), && \text{(A.17)}
\end{aligned}
$$

Here, σ is a sigmoid function whose range is $[0, 1]$, and F and G are other activation functions, such as $tanh$. The gating functionality corresponds to the element-wise multiplication of Equations A.17 and A.17. These gating values are determined by the input state and the previous state.

Appendix B

Robot Experiment Hardware Setup

This chapter provides the details of the robot experiments given in Chapter 5. For a robot platform, the humanoid robot NAO was used. Figure B.1 shows the experimental setup. The robot was positioned on the Wacom pen tablet holding a stylus pen. To avoid capturing errors, the robot was fixed to the pen tablet through a metal plate whose design is described in Figure B.3. The pen was also attached to the robot's hand to avoid any unexpected movement by using an adapter, as shown in Figure B.2. This adapter allowed the pen to move vertically when it was pushed against the tablet.

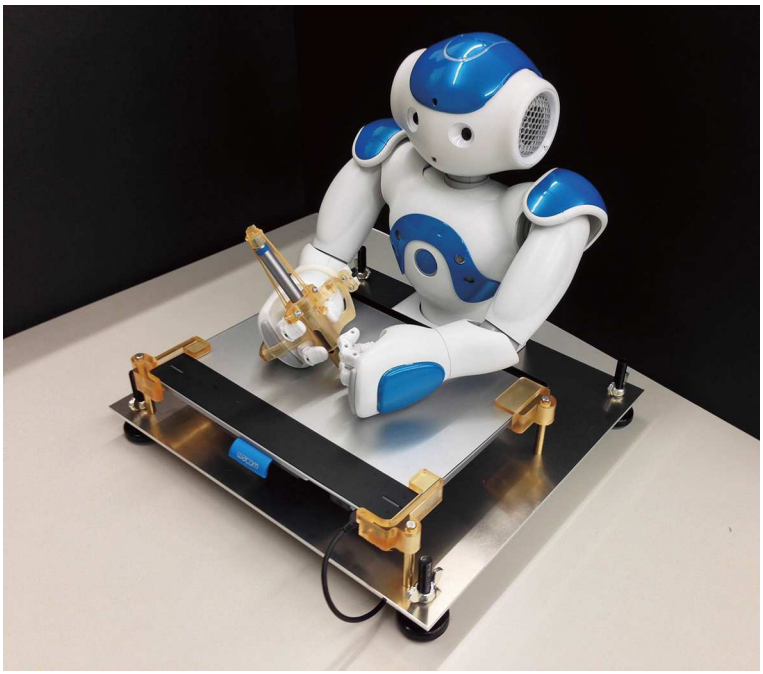


Figure B.1: The setup for robot experiments

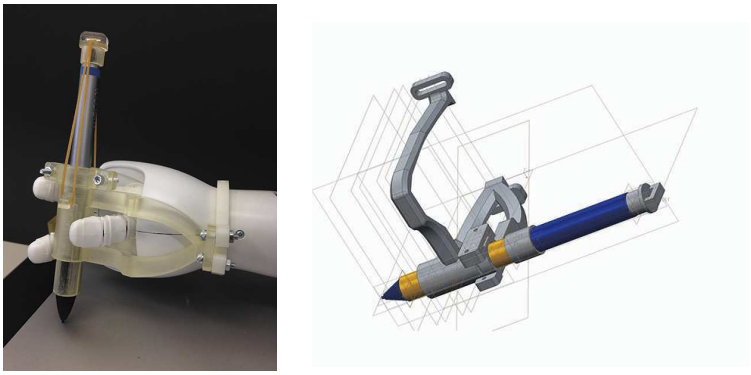


Figure B.2: The robot hand with the pen adapter

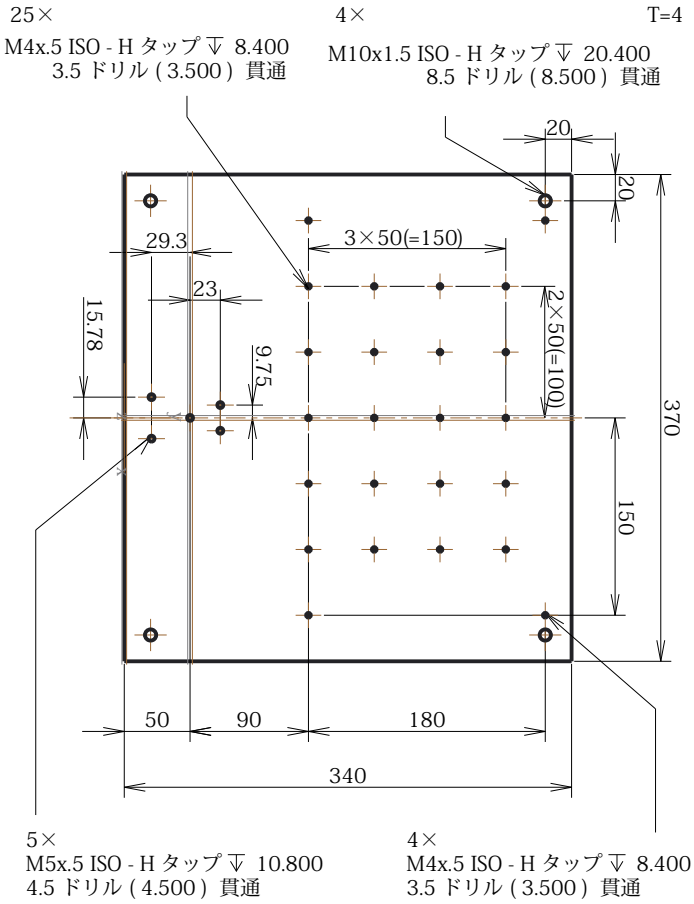


Figure B.3: Design of the plate to fix the robot and tablet

Appendix C

Embodiment Informatics

The author has been supported by a scholarship program called the Graduate Program for Embodiment Informatics by the Ministry of Education, Culture, Sports, Science, and Technology (MEXT). This program proposes embodiment informatics as an interdisciplinary field of mechanical engineering and informatics. This combination of subjects is expected to bring forth new studies to create computational systems that have an embodiment driven by the cutting edge of intelligent systems. This chapter explains the methodology of this thesis as a study of embodiment informatics.

Mechanical engineering and informatics attempt to formulate phenomena differently. Mechanical engineering typically represents a phenomenon as a physical system in continuous space, such as the kinematics of robots. However, informatics attempts to use a discrete system of symbols. For example, a controlling system for a moving robot can be implemented by a system that can solve kinematics problems to move the robot to the desired direction. When we want to control this robot according to the commands given by a user, the system needs to include a sub-system that converts the given route into a series of commands for the controller. In this case, a behavior of the robot is represented as a sequential data of position, velocity, or acceleration of the mass points. However, the behavior can also be written as an oriented graph.

One of the keys of letting mechanical engineering and informatics exist together in an intelligent system for humans is to determine how to design the interactions of the representations given by each field. This interaction limits the system's

intelligence. To control a moving robot, it is important to determine how the sub-system interprets the user's commands and converts them into other types of representations to be used in the controller. The design methodology may depend on what we want to do using the system, but it is challenging in general because of the diversity of symbols in the world [121]. The command "Go there" can lead to many possible moves depending on the interactions between the system and user.

The main problem resolved in this thesis is the diversity associated with pictures depicted by humans. The diversity lies in three types of representations: symbol (name or category), image, and the drawing process. A category "bear" can have any variations of drawn bears and all of them would be regarded as bears. Even the same depiction target shown can be drawn by many different processes. The methodology adopted in this thesis is to consider the drawing process as a visuomotor adaptation process. In this sense, the drawn picture corresponds to the goal of the process. The process to solve the diversity of the symbol and the image was implemented as two computational processes to convert from one of these representations to another. The process from the image to a symbol was replicated by the functionality acquired by training DNNs. Another process from the symbol to an image is implemented by exploring the drawing process by minimizing the prediction error of the drawn picture. The diversity ways of drawing an image is taken as visuomotor sequences learned by a recurrent neural network (RNN). The computational theory a combination of discrete and continuous representations of drawing is achieved by using the flexibility of neural networks (NNs). In general, NNs can be optimized to approximate the variations of probability functions that are not limited to classification or regression. An NN can be seen as a converter between different representations.

One problem that can be understood by embodiment informatics for human system interactions is the social behavior of the system. Social behavior means the way the system affects communications among agents who have own representation systems. For example, the idea of beauty plays a role in interactions between many styles and persons who judge whether the work is good or not. This does not deny the studies for understanding the sense of beauty defined by

cognitive sciences [122, 25]. Besides the cognitive aspects, we need to consider the cultural backgrounds and the consent among different viewers. Other human belief systems should also be discussed with multiple agents who share each system to solve his or her problems in the real world.

Relevant Publications

Journal Papers

1. Kazuma Sasaki, Kuniaki Noda, and Tetsuya Ogata. "Visual Motor Integration of Robot's Drawing Behavior using Recurrent Neural Network", Robotics and Autonomous Systems, Vol. 86, pp. 184-195, 2016.

International Conferences

1. Kazuma Sasaki and Tetsuya Ogata. "End-to-end Visuomotor Learning of Drawing Sequences using Recurrent Neural Networks", International Joint Conference on Neural Networks, Rio, Brazil, July, 2018.

2. Kazuma Sasaki, Madoka Yamakawa, Kana Sekiguchi, and Tetsuya Ogata. "Classification of Photo and Sketch Images using Convolutional Neural Networks", 25th International Conference on Artificial Neural Networks, Barcelona, September, 2016, Lecture Notes in Computer Science, Vol. 9887.

3. Kazuma Sasaki, Hadi Tjandra, Kuniaki Noda, Kuniyuki Takahashi, and Tetsuya Ogata. "Neural Network based Model for Visual-motor Integration Learning of Robot's Drawing Behavior: Association of a Drawing Motion from a Drawn Image", IEEE/RAS International Conference on Intelligent Robots and Systems, Hamburg, September, 2015.

Domestic Commentary

1. 佐々木一磨，尾形哲也. “手描きスケッチを扱う深層学習モデル”, 日本画像学会誌 Vol. 56, No. 2 pp. 177-186, 2017.

Domestic Conferences

1. 山川まどか，関口香菜，佐々木一磨，尾形哲也. “Convolutional Neural Network による写真と手描きスケッチの認識”， 第 30 回人工知能学会全国大会，福岡, 2016 年 6 月.

2. 佐々木一磨，Hadi Tjandra，野田邦昭，高橋城志，尾形哲也. “再帰結合型神経回路モデルによる描画像からの描画運動連想”，計測自動制御学会 システムインテグレーション部門講演会 SI2014, 東京, 2014 年 12 月.

Other Publications

Journal Papers

1. Ping-Chu Yang, Kazuma Sasaki, Katana Suzuki, Kei Kase, Shigeki Sugano, and Tetsuya Ogata. “Representable Folding Task by Humanoid Robot Worker using Deep Learning”, IEEE Robotics and Automation Letters(RA-L), Vol. 2, No.2, pp. 397-403, 2017.

Domestic Conferences

1. 本吉俊之，佐々木一磨，大西直，曽田尚宏，尾形哲也. “時系列情報を入力とするCNNを用いた自動運転ステアリング・アクセル学習”，第35回 日本ロボット学会 学術講演会，埼玉, 2017年9月.

2. 大西直、佐々木一磨、本吉俊之、菅佑樹、尾形哲也. “ロボットシミュレーション環境構築フレームワーク「RTM-Unity Sim」の開発”，日本機械学会ロボティクスメカトロニクス講演会，福島, 2017年5月.

3. 陽品駒，佐々木一磨，鈴木彼方，加瀬敬唯，高橋城志，菅野重樹，尾形哲也. “Wizard of Ozと深層学習によるロボットの柔軟物折り畳み作業”，第34回日本ロボット学会学術講演会, 山形, 2016年9月.

4. 橋本直矢，佐々木一磨，中臺一博，尾形哲也. “時系列を考慮したConvolutional Neural Networkによる視覚音声認識のための音素識別”， 第34回 日本ロボット学会学術講演会, 山形, 2016年9月

5. 松永寛之，橋本直矢，佐々木一磨，中臺一博，尾形哲也. “音素バランスを

考慮した読み上げ用フリー文章 データベースの構築手法”，第 30 回人工知能学会全国大会，福岡, 2016 年 6 月.

6. 佐々木一磨，Hadi Tjandra，野田邦昭，高橋城志，尾形哲也. “再帰結合型神経回路モデルによる描画像からの描画運動連想”，計測自動制御学会 システムインテグレーション部門講演会 SI2014, 東京, 2014 年 12 月.

7. 寺田翔太，佐々木一磨，有江浩明，野田邦昭，菅佑樹，尾形哲也. “レコードスケッチ”，計測自動制御学会 システムインテグレーション部門講演会 SI2013, 神戸, 2013 年 12 月.

8. 佐々木一磨，寺田翔太，有江浩明，野田邦昭，菅佑樹，尾形哲也. “マルチメディア向けグラフィカル統合開発環境「Max」と RTC を繋ぐブリッジプラグインの開発”，計測自動制御学会 システムインテグレーション部門講演会 SI2013, 神戸, 2013 年 12 月.