A Deep Learning Approach for Seamless Integration of Cognitive Skills for Humanoid Robots

Jungsik Hwang, Minju Jung, Jinhyung Kim and Jun Tani*

Korea Advanced Institute of Science and Technology, Daejeon, South Korea
{jungsik.hwang, minju5436, kkjh0723, tani1216jp}@gmail.com

***Abstract*—This study investigates the seamless integration of cognitive skills, such as visual recognition, attention switching, action preparation and generation for a humanoid robot. In our preliminary study [1], the deep dynamic neural network model was introduced to process spatio-temporal visuomotor patterns. In the current study, we extended the previous model further to enhance its capability of handling sequential visuomotor information as well as forming visuomotor representation. We conducted synthetic robotic experiments in which a robot learned goal-directed actions of reaching to grasp objects under two different experimental settings. In the first experiment, a task of reaching to grasp objects was conducted under parameterized visual occlusion condition for the purpose of examining the memory capability in the model. In the second experiment, the action of reaching to grasp objects was incorporated with visual recognition of human gesture patterns with using the working memory. The experimental results revealed that the proposed model was able to generalize its reaching and grasping skills in the novel situations. Furthermore, the analysis using the dimensionality reduction technique on neuron activation verified that the proposed model was capable of manipulating high dimensional spatio-temporal visuomotor patterns by forming their dynamic link to the actional intention developed in the higher level of the model via iterative learning.**

***Index Terms*—Deep learning, developmental robotics, humanoids, visuomotor coordination.**

I. Introduction

Deep learning [2] is a fast-growing field in machine learning and artificial intelligence. It has attracted widespread attention by showing outstanding performance in the several tasks, such as visual recognition, speech recognition, text pattern recognition and many others. (See [2] and [3] for a recent review on deep learning.) One of the most important characteristics of deep learning is that it can autonomously extract task-related features in the data without the necessity of hand-engineered feature extraction methods [2-4]. Consequently, deep learning also seems promising in the robotics context. By properly utilizing deep learning techniques, a robot can learn directly from its raw sensorimotor data acquired through dynamic interaction with its environment [4].

Recently, Hwang and colleagues [1] have proposed a deep neural network model called Visuo-Motor Deep Dynamic Neural Network (VMDNN) which was designed to process and integrate raw visuomotor patterns. The VMDNN model consisted of three different types of subnetworks: Multiple Spatio-Temporal scales Neural Network (MSTNN) [5], Multiple Timescales Recurrent Neural Network (MTRNN) [6] and PFC (Prefrontal Cortex) subnetworks. MSTNN and MTRNN were used to process dynamic visual images and to control robot's action and attention respectively. Those two subnetworks were tightly integrated through the PFC subnetwork so that the whole system could process dynamic visuomotor patterns in a seamless manner. However, there are several limitations in this study. The PFC layer was limited, such that it was not equipped with the recurrent loops and backward connection from the proprioception. Our preliminary study indicated that this prevented the generalization capability of the model especially in the task of object grasping associated with human gesture recognition.

In the current study, we extended the previous VMDNN model further by introducing recurrent loops in the PFC subnetwork as well as the backward connection from MTRNN to PFC. We conducted synthetic robotics experiments to evaluate the model and to understand possible biological mechanisms of learning goal-directed actions. We particularly focused on the developmental learning aspect of visuomotor coordination for reaching and grasping behavior of a humanoid robot. It requires a robot to coordinate a set of cognitive skills such as visual recognition, attention switching, action preparation and generation. In the first experiment, we examined the model in a visual occlusion experiment. During the training phase, a robot was learned to grasp a target object located on a task space. During the occlusion testing, the visual input to the network was unexpectedly occluded during

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2014R1A2A2A01005491) and by the Industrial Strategic Technology Development Program (10044009) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea).

*The correspondence should be sent to tani1216jp@gmail.com.

reaching. This was to verify whether the network is equipped with a sort of internal memory so that it can be generalized to the case when the visual information is completely occluded. In the second experiment, a robot was learned to recognize gestures demonstrated by several human subjects and to grasp the target object specified by the gestures. In both experiments, we examined the model's generalization capability to the unlearned situations. In addition, we clarified the internal representations by analyzing the neuron activation using t-Distributed Stochastic Neighbor Embedding (t-SNE) dimensionality reduction algorithm [7]

II. The Deep Neural Network Model

In this section, we describe the dynamic deep neural network model called Visuo-Motor Deep Dynamic Neural Network (VMDNN) [1] in detail. It was designed to process and integrate raw visuomotor information through a hierarchical structure which is considered as the essential characteristic of cortical computation [8, 9]. Also, the VMDNN model is multimodal such that both perception and action are not separated but tightly intertwined within the system. The model is composed of three subnetworks: (1) vision subnetworks that process dynamic visual images (MSTNN), (2) action subnetworks that control robot's action and attention (MTRNN) and (3) prefrontal cortex (PFC) subnetwork which is located on the top of those two subnetworks and dynamically integrates them.

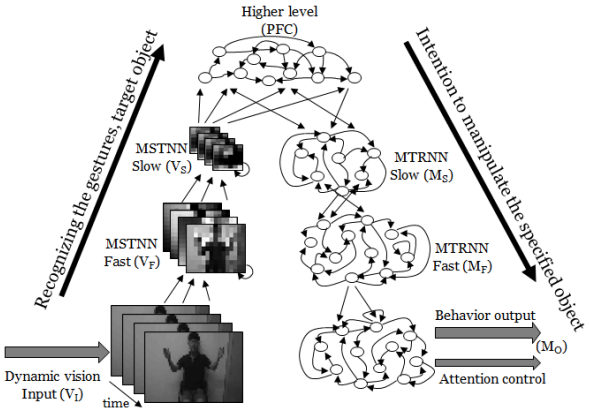

Fig. 1. The VMDNN model consisting of three distinctive subnetworks: MSTNN for dynamic vision processing (left), MTRNN for action generation (right) and PFC for integration (top)

A. MSTNN (Multiple Spatio-Temporal Scales Neural Network)

MSTNN (Multiple Spatio-Temporal Scales Neural Network) was employed in the model to process dynamic visual images perceived while the robot is conducting a task. MSTNN is an extension of Convolutional Neural Network (CNN) with leaky integrator neural units with a different time constants [5]. Unlike CNN [10] which was designed to handle static patterns, the MSTNN was designed to process dynamic patterns, making it an adequate candidate for robotics applications. In the model, the MSTNN subnetwork consists of vision input (V_I) layer, MSTNN-fast (V_F) layer which has short distant connectivity with smaller time constants and MSTNN-slow (V_S) layer which has longer distant connectivity with bigger time constants. Each layer has a set of feature maps consisting of neural units and has forward connections from V_I to PFC.

B. MTRNN (Multiple Timescales Recurrent Neural Network)

In this model, MTRNN (Multiple Timescales Recurrent Neural Network) was adopted for behavior generation and attention control. MTRNN [6] has a hierarchical structure consisting of a multiple CTRNN with leaky integrator neural units. The lower level has a smaller time constant showing fast dynamics whereas the higher level has a bigger time constant exhibiting slow dynamics. Due to this temporal hierarchy, MTRNN can learn compositional action sequences and the meaningful functional hierarchy can emerge within the system [6, 9]. In our model, the MTRNN subnetwork has a hierarchy consisting of MTRNN-slow (M_S) showing slow dynamics with the bigger time constant, the MTRNN-fast (M_F) showing fast dynamics with smaller one and the MTRNN-output (M_O) with the smallest one. M_S and M_F are asymmetrically connected to every neuron in the neighboring layers including itself. M_O is a softmax layer and it receives inputs from M_F and generates behavior outputs and attention control signals.

C. PFC (Prefrontal Cortex)

On the top of those two dynamic networks, we allocated the PFC (Prefrontal Cortex) layer consisting of a set of leaky-integrator neurons. In this study, we extended the previous VMDNN model by introducing the recurrent loops in the PFC layer as well as the backward connection from M_S to PFC. Consequently, the PFC layer receives inputs from V_S, M_S and itself and it sends output to M_S. This layer is characterized by the following aspects. First, the neurons in the PFC layer are assigned with the largest time constant so that it can show the slowest-scale dynamics. Therefore, it is able to maintain the higher-level task-related information at the PFC layer throughout the task phases. Second, the PFC layer has recurrent connections which are essential to handle dynamic sequential sensorimotor flows by keeping track of time concept [4, 11-13]. Third, the PFC layer is designed to integrate two monomodal subnetworks (MSTNN and MTRNN) and form multimodal representations abstracted from raw visuomotor data.

D. Action Generation Mode

The internal states of all neural units were initialized with neutral values at the onset of action generation mode. Then, the grayscale pixel image obtained from robot's camera was given to the vision input layer (V_I) and each neural unit's internal state and activation in every subnetwork were computed from V_I to M_O successively. The outputs at the M_O layer consisting of arm and hand movements as well as attention control signals were transformed to control the robot's actual joints and attention.

At each time step t, the internal state u_i^{txy} and the dynamic activation v_i^{txy} of the neural unit located on the (x, y) position in the *i*th feature map of each MSTNN layer ($i \in V_F \vee V_S$) is determined by the following formula:

$$u_i^{txy} = \left(1 - \frac{1}{\tau_i}\right) u_i^{(t-1)xy} + \frac{1}{\tau_i}\left[\sum_{j \in V_j} \left(k_{ij} * v_j^t\right)_{xy} + b_i\right] \quad (1)$$

$$v_i^{txy} = 1.7159 \times \tanh\left(\frac{2}{3}u_i^{txy}\right) \quad (2)$$

τ is the time constant, V_j is the feature maps in the previous layer (if $i \in V_F$, then $V_j = V_I$ and if $i \in V_S$, then $V_j = V_F$), k_{ij} is the value of the kernel, b is the bias, and * is the convolution operator.

From the PFC layer to the M_O layer, the internal state u_i^t and the dynamic activation y_i^t of the *i*th neuron in the PFC and MTRNN layers ($i \in PFC \vee M_S \vee M_F \vee M_O$) can be computed by the following equations:

$$u_i^t = \left(1 - \frac{1}{\tau_i}\right)u_i^{t-1} + \begin{cases} \frac{1}{\tau_i}\left[\sum_{j\in V_S} k_{ij} * v_j^t + \sum_{k\in M_S \vee PFC} w_{ik} y_k^{t-1} + b_i\right] & if\ i \in PFC \\ \frac{1}{\tau_i}\left[\sum_{j\in PFC} w_{ij} y_j^t + \sum_{k\in M_F \vee M_S} w_{ik} y_k^{t-1} + b_i\right] & if\ i \in M_S \\ \frac{1}{\tau_i}\left[\sum_{j\in M_S \vee M_F} w_{ij} y_j^{t-1} + b_i\right] & if\ i \in M_F \\ \frac{1}{\tau_i}\left[\sum_{j\in M_F} w_{ij} y_j^t + b_i\right] & if\ i \in M_O \end{cases} \quad (3)$$

$$y_i^t = \begin{cases} 1.7159 \times \tanh\left(\frac{2}{3}u_i^t\right) & if\ i \in PFC \vee M_S \vee M_F \\ \frac{\exp(u_i^t)}{\sum_{j\in M_O}\exp(u_j^t)} & if\ i \in M_O \end{cases} \quad (4)$$

τ is the time constant, k_{ij} and w_{ij} is the value of the kernel and weight from the *j*th unit to the *i*th unit and b is the bias.

E. Training Mode

The model was trained with visuo-proprioceptive sequences under a supervised learning framework. The training data is raw visuomotor data obtained from a repeated tutoring in which a robot was manually operated without the neural network. Backpropagation through time (BPTT) [14] was used to learn the parameters of the network. At the beginning of the training, the learnable parameters in the MSTNN subnetworks were initialized by means of the pre-training. Previous studies [3, 15] have shown that pre-training can provide efficient initialization of the network parameters. Pre-training of MSTNN was conducted by allocating the softmax output layer on the top of PFC and removing the connections from the MTRNN subnetworks including recurrent connections at PFC layer. In this configuration, the system was equivalent to the typical MSTNN model and it was trained to classify the visual images. Once the pre-training is completed, the parameters in the visual pathway were used in the proposed model as the initial values.

After the pre-training, the end-to-end training was conducted in which the network's entire learnable parameters (w_{ij}, k_{ij}, b_i) were updated to minimize the error E represented by Kullback-Leibler divergence between the teaching signal $\bar{y}_i^t$ and the network's output y_i^t.

$$E = \sum_t \sum_{i\in M_O} \bar{y}_i^t log \frac{\bar{y}_i^t}{y_i^t} \quad (5)$$

A stochastic gradient descent method was applied during the training and the learnable parameters were updated when each visuo-proprioceptive sequence was presented.

$$w_{ij}(n+1) = w_{ij}(n) - \eta\left(\frac{\partial E}{\partial w_{ij}} + 0.0005 w_{ij}(n)\right) \quad (6)$$

$$k_{ij}(n+1) = k_{ij}(n) - \eta\left(\frac{\partial E}{\partial k_{ij}} + 0.0005 k_{ij}(n)\right) \quad (7)$$

$$b_i(n+1) = b_i(n) - \eta\left(\frac{\partial E}{\partial b_i}\right) \quad (8)$$

where η is the learning rate and n is an index of the learning step. Also the weight decay method was used to prevent overfitting [10] and the weight decay rate was set to 0.0005.

III. Experiment Settings

We conducted two experiments to evaluate the proposed model. In the first experiment (Experiment I), we examined the model in reaching and grasping an object task. In the second experiment (Experiment II), we further extended the first experiment by incorporating human gesture recognition. In the following sections, we describe the experiment settings that were used throughout the two experiments.

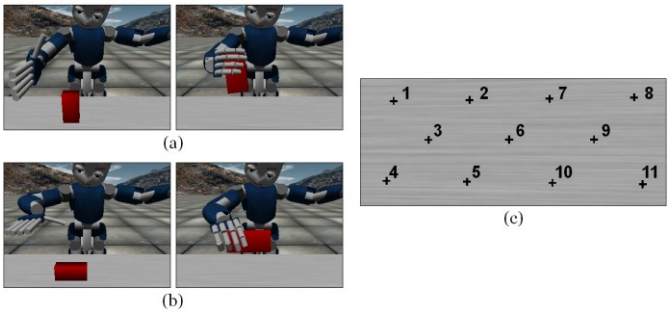

Fig. 2. iCub simulator used in our experiments. The figures are showing two different ways of reaching and grasping: (a) reaching and grasping a tall object from side and (b) reaching and grasping a long object from above. (c) 11 object locations on a task space denoted by a number.

A. Robotic Platform

Simulation of the iCub humanoid robot (Fig. 2) was used for the experiments. iCub [16] is a humanoid robot in a child-like shape and it has 53 degrees of freedom (DoF) distributed in the entire body. Its simulator provides an adequate platform for studying developmental robotics since the robot's physical interaction with the environment can be well reconstructed with a good accuracy [8, 17, 18].

Regarding the visual input to the network, we used the dynamic visual images obtained from iCub simulator's camera embedded in its left eye. We first resized those images to 64 (w) × 48 (h) and then, converted to grayscale and normalized to -1 to 1. Regarding the joints of the robot, we used the robot's right arm consisting of 7 DoF (shoulder's pitch, roll, yaw, elbow, wrist's pronosupination, pitch, and yaw). In addition, the network also outputted the level of extension or flexion of the fingers' joints for grasping as similar to [19]. Furthermore, we used two attention control mechanisms. First, two joints in the neck (pitch and yaw) were used to orient its head so that the attended object can be located at the center of its view. Another attention control mechanism is foveation. That is, the resolution of the visual frames to the neural network varied depending on the level of focus. Throughout the experiments, two different types of objects were used: a tall object with a size of 2.8cm × 5cm × 10cm and a long object with a size of

2.8cm × 10cm × 5cm. The object was placed on 11 positions distributed on XY-plane of the task space with 5 different orientations (-45°, -22.5°, 0°, 22.5°, 45°).

B. Network Configurations

The VMDNN model used in our study consists of 7 layers and each layer consists of a set of feature maps. The model parameters, such as a number and the size of feature maps, kernel size vary between the layers. Table I illustrates the settings of the parameters used throughout the experiments.

TABLE I
PARAMETER SETTINGS

Layer	Number of Feature Maps	Size of Feature Map	Kernel Size	Sampling Factor
V_I	1	64×48	-	-
V_F	4	15×11	8×8	4
V_S	8	5×3	7×7	2
PFC	20	1×1	5×3	1
M_S	30	1×1	-	-
M_F	50	1×1	-	-
M_O	110	1×1	-	-

Regarding the time scale properties, we mainly compared 4 different temporal scale configurations in the visual pathway and PFC of the model: CNN with fast-scale PFC, CNN with slow-scale PFC, MSTNN with fast-scale PFC and MSTNN with slow-scale PFC. For each condition, the time constants were set differently (Table II). Throughout the experiments, the time constants of M_S, M_F and M_O were set to 70, 2 and 1 respectively.

TABLE II
FOUR TYPES OF NETWORK CONDITIONS AND TIME CONSTANT SETTINGS

Type of Vision Layer	PFC Dynamics	Time Constant Settings		
		V_F	V_S	PFC
CNN	*Fast*	1	1	1
CNN	*Slow*	1	1	150
MSTNN	*Fast*	1	15	1
MSTNN	*Slow*	1	15	150

IV. EXPERIMENT 1: REACHING AND GRASPING AN OBJECT

In the first experiment, we conducted an experiment in which a robot reaches and grasps the target object on the task space. The overall task flow is as follows: First, at the home position, the robot faces the black screen located in front of it. Then, the robot shifts attention to the task space on which the target object (either long or tall) is located. Then, the robot attends to the object by locating the target at the center of the view through orienting its head. Once the target object is attended, the robot starts reaching and grasping the target object. Please note that the way of reaching and grasping is different depending on the type of object (Fig. 2). When the hand reaches close to the object, the robot controls the level of foveation so that the visual image containing the object and robot's hand can be given to the network with a higher resolution (please see the supplementary video).

A. Experiment Settings

We conducted supervised end-to-end training on the proposed model. The training data consisted of vision-motor pairs and it was acquired from repeated tutoring in which a robot was manually operated by the experimenter. During the training, the network learned 110 cases consisting of 11 different object positions, 2 different object types (tall and long) and 5 different object orientations. During training, the network parameters in the MSTNN layers were first initialized by means of the pretraining as described in Section III. In the pretraining, the MSTNN layers were trained to classify the location, type and orientation of the object. The other learnable parameters in the network were randomly initialized between -0.025 to 0.025. The network was trained for 13,000 epochs and the learning rate was set to 0.01.

During the testing, we examined the proposed model in a visual occlusion experiment (please see the supplementary video). Vision input to the network was blocked at the 30th (onset of reaching), 50th and 70th (onset of grasping) step. The main focus of this experiment is to verify whether the network is equipped with a sort of internal memory so that it can be generalized to the case when the visual information is completely occluded. From the training patterns, we randomly selected 2 orientations in each type of object for 11 orientations (a total number of 44). We also examined the network's generalization capability under the various object variations by testing 44 trials in which the object was randomly placed with varying orientations from -45° to 45° on the task space.

B. Results

The performance of the model was evaluated in terms of a success rate across the training and testing trials. Each trial was evaluated as success if the robot grasped the target object and failure otherwise. Fig. 3 illustrates the success rate of (a) training and (b) testing trials with respect to the different visual occlusion timing with the aforementioned 4 different timescale configurations. As we expected, the performance generally degraded when the vision input was blocked at the earlier phases. Especially, when the visual input was occluded at 30th step (onset of reaching), the performance degraded drastically except for the MSTNN with slow-scale PFC condition. We assumed that the model's memory capability played important role especially when the vision input was blocked at the earlier stage. That is, the robot was able to maintain information about the object's position and orientation throughout the task phases by means of the memory capability. When the vision input was not blocked, the all conditions showed the high success rate in both training and testing patterns, meaning that the model successfully learned the training data and it was able to generalize reaching and grasping skill to the unlearned object positions and orientations.

See the supplementary videos at http://neurorobot.kaist.ac.kr/video/jungsik/icdl2016.html

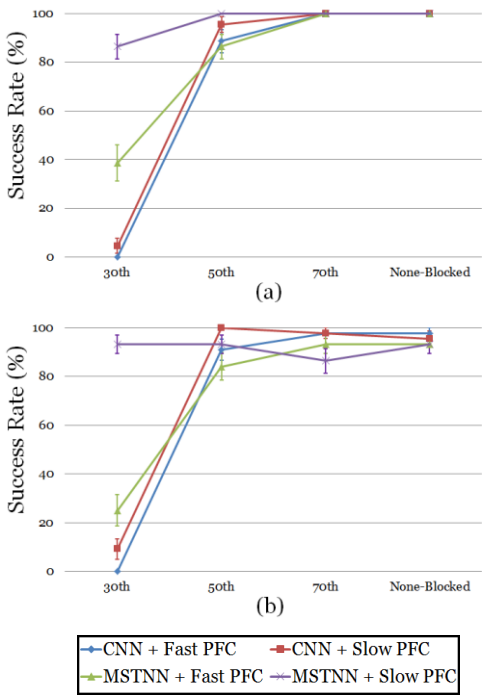

Fig. 3. The success rate of the four different network configurations in the visual occlusion experiment. (a) testing on the learned object positions and orientations (training trials) and (b) testing on the unlearned object positions and orientations (testing trials).

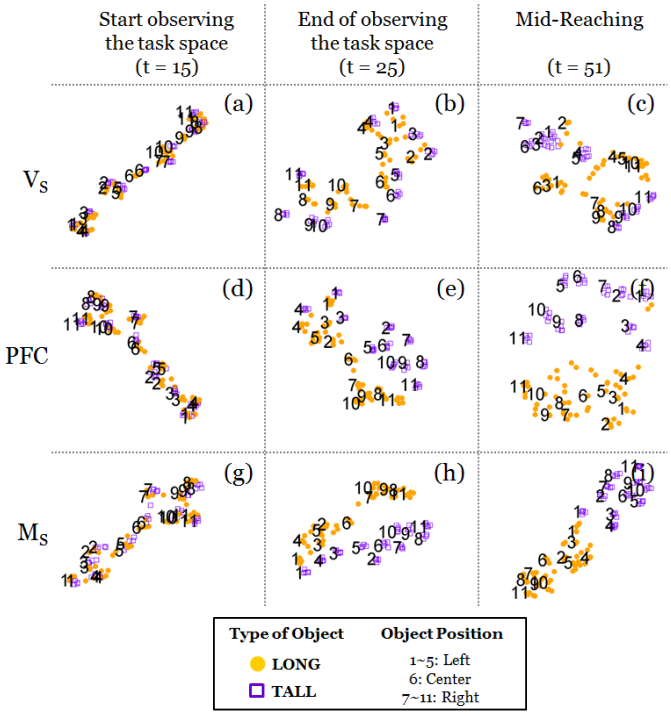

Fig. 4. The development of internal representation at V_S, PFC and M_S in three different task phases. Each point represents a single trial and the distance between those points represents the relative similarity between the trials. The shape and the color of each point denote the type of object and the type of human gesture respectively. The number next to each point denotes the object's position. Please note that the way of grasping is different depending on the type of the object.

C. *Development of Internal Representation*

In order to clarify the internal representation emerged in the proposed model, we analyzed the neuron activation using t-Distributed Stochastic Neighbor Embedding (t-SNE) dimensionality reduction algorithm [7] (Fig. 4). In the tSNE analysis, the neuron activation data were first transformed in the 10-dimensional PCA (Principal Component Analysis) space with setting the perplexity as 30. Similar to [20], we focused on the relationship between each patterns so we did not plot the axes which are varying between plots.

When the robot first started observing the task space (a, d and g), the object representation regarding the location appeared in three layers but there was no clear distinction between the two types of object. This indicated that the model was able to recognize the location of the object at a glance but it was not sufficient to identify the type of the object. After the robot sufficiently observed the task space (b, e and h), the object representations began to be differentiated by both location and the type. Particularly, it is interesting that the M_S layer already encoded the location and the type of the object even before the robot started orienting its head and reaching (h). This implied that an appropriate action for a given object was pre-planned or calibrated at a higher-level proprioception (M_S) before the robot actually moved its head or arm. It is assumed that this preplanning capability might also have significant influences in the visual occlusion experiment, enabling the robot to reach and grasp the target object even without monitoring the target and the hand during reaching. Several studies in children development [21, 22] also argued that after the proprioceptive information is calibrated based on the visual information, vision is no longer necessary afterwards. This finding is similar to neuroscientific evidences in which F5 neurons have been shown to encode show grip-specific information even no movement is intended [23]. The analysis also clearly illustrated the different representations developed in each layer. In the mid-reaching phase, M_S layer mainly encoded the way of grasping which was equivalent to the type of object (i). In the PFC layer, the representation about the target objet was encoded with respect to both location of the object (horizontally) and the way of grasping (vertically).

V. Experiment II: Reaching and Grasping an Object with Gesture Recognition

In this experiment, we incorporated gesture recognition into the previous experiment. The robot first observed the human gesture and grasped the target object indicated by the gesture. Therefore, this task inherently requires the model to have working memory to maintain the gesture information throughout the task phases. The main task flow was as follows. The robot first observed a human gesture displayed on the screen (Fig. 5). There were four different types of human gestures indicating: the left, right, long and tall object. After observing the gesture, the robot shifted attention to the task space by orienting its head. Two objects consisting of one tall object and one long object were placed on the task space and the robot reached and grasped the target object which was indicated by the human gesture at the beginning of the task

(please see the supplementary video). For example, when the human gesture was indicating the left object, the robot had to figure out the type and orientation of the object placed on the left side of the task space. Similarly, the robot had to figure out where the long object was when the gesture was indicating a long object. To successfully achieve this task, the robot had to maintain the information about human gesture displayed at the beginning and combine it with the object properties perceived while observing the task space and objects.

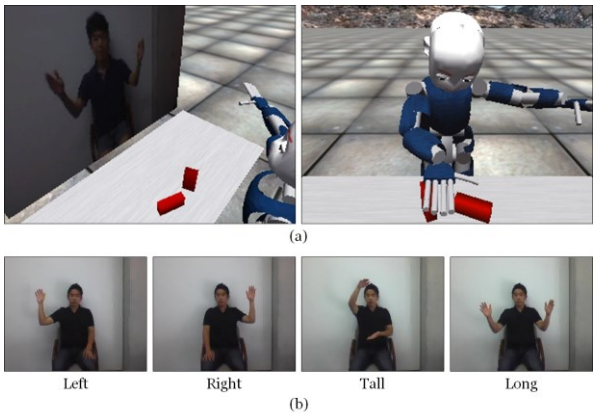

Fig. 5. The experiment setting in Experiment II. (a) Human gesture is displayed on the screen in front of the robot and two objects consisting of one tall and one long object are located on the task space. (b) Four different types of gesture

A. Experiment Settings

During the training, the robot learned 200 cases consisting of varying human gestures and object configurations. There were four different gestures indicating left, right, long and tall object. The 40-frame video clips of human gesture were displayed on the screen located in front of the robot. 7 different subjects' gestures with several trials for each gesture were collected and they were randomly selected in the training dataset. Regarding the object's configuration such as type, position and orientation, we used the same configuration as in Experiment I, except the one on the center of the task space. Two objects consisting of one tall and one long object were presented on the task space systematically so that the way of presenting objects did not bias the robot's behavior. The parameters in MSTNN layers were initialized by the pre-training in which MSTNN was trained to classify the human gestures. The other parameters including those in the MTRNN layers were initialized by the values learned from Experiment I to enhance the convergence speed of the learning. The network was trained for 5,000 epochs with the learning rate of 0.01.

During the testing, we examined the model's generalization capability with respect to the novel situations. We examined the model by randomly locating the objects with a random orientation presented with gestures of a novel subject. For the training and testing trials, we examined 200 and 80 cases respectively.

B. Generalization Performance of the Model

Table III shows the success rate of each network condition in Experiment II. The success rate was computed the same way for the training and testing trials as in Experiment I. The MSTNN with slow-scale PFC condition showed the higher success rate than the other network conditions. The model successfully learned the training sequences and it was able to generalize to the different conditions. For the learned gesture trials and the learned object configurations (Training trials), the model showed the highest success rate (93.50%). In the testing trials, it was shown that the model was able to conduct the task even the object was randomly located on the task space and indicated by the novel subject that was not experienced during training (78.75%). It is worth noting that the model showed the relatively poor performances when it was set to have fast-scale PFC. This is due to the characteristics of the task which requires the robot to remember the gesture displayed at the beginning throughout the task phases.

TABLE III
THE SUCCESS RATE OF FOUR NETWORK CONDITIONS IN EXPERIMENT II

Network Condition	Success Rate	
	Training Trials	***Testing Trials***
CNN + Fast PFC	51.50%	40.00%
CNN + Slow PFC	86.50%	76.25%
MSTNN + Fast PFC	84.50%	57.50%
MSTNN + Slow PFC	93.50%	78.75%

C. Development of Internal Representation

We plotted the neural activation (Fig. 6) of V_S, PFC and M_S in three different task phases using t-SNE algorithm [7]. In V_S, the visual information about the target object was mainly encoded. For example, when the robot started observing the target object (b), V_S encoded the pair of two objects presented on the task space simultaneously and there was no clear distinction between the target object and 'another' object next to the target object. When the robot was reaching for the target object (c), V_S encoded the target object's location and type but the distinction was less clear than that in PFC (f). In PFC, there were four clusters representing the type of gesture at the end of observing the gesture (d). Then, those clusters started to develop progressively to represent the specific target object regardless of the human gesture (f). This can be understood that the dynamic visual images of a human gesture were abstracted via hierarchical processing of the model and the PFC layer encoded one of the four "intentions" underlying the human gesture. While the robot was observing the target object and reaching for it, the human's intention displayed by the gesture was translated into robot's own intention for reaching and grasping the specific target object. This development of neuron activation also implied that such robot's intentions were dynamically computed rather than directly mapped from visual perception. Interestingly, the activation of neurons in M_S was clearly different even the robot was remained in the same position and did not start reaching (g and h). The four clusters representing the each type of gesture were formed at the end of observing the human gesture (g) and the onset of observing the target object (h). This implied that higher level proprioception (M_S) was calibrated based on the perceived visual information

(gesture) before reaching.

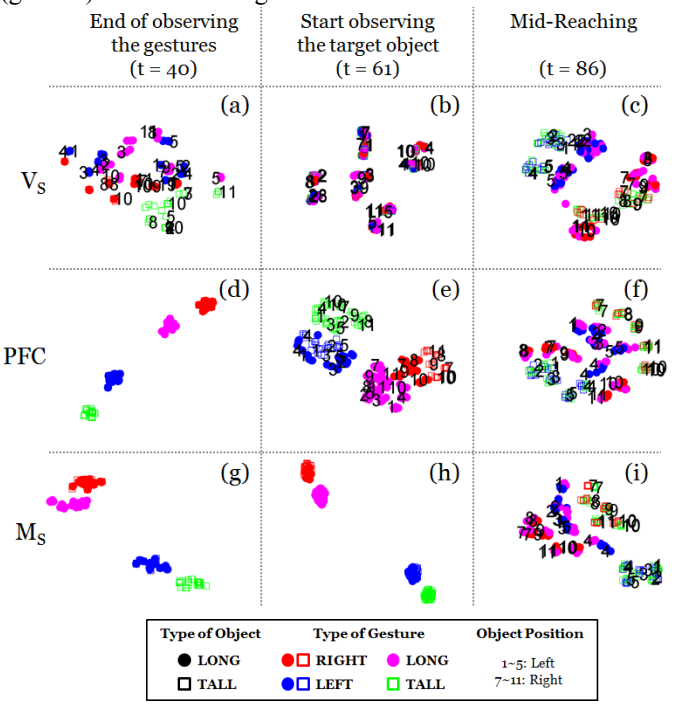

Fig. 6. The development of internal representation at V_S, PFC and M_S in three different task phases. Each point represents a single trial and the distance between those points represents the relative similarity between the trials. The shape and the color of each point denote the type of the target object and the type of human gesture respectively. The number next to each point denotes the object's position. In (d), (g) and (h), the numbers were omitted due to the large overlap between them.

VI. Summary and Conclusion

In this study, we clarified how deep learning schemes can be utilized to integrate a set of cognitive skills of a humanoid robot in a seamless manner. We extended the dynamic neural network model called VMDNN (Visuo-Motor Deep Dynamic Neural Network) and evaluated it thoroughly by conducting synthetic robotic experiments. Throughout the experiments, we verified that the proposed model was capable of learning goal-directed actions which require seamless integration among visual recognition, attention switching, action preparation and generation. The robot was able to link the lower level perception of large dimensionality into the higher level "conceptual" actional intention for generating precise motor program to be executed. There are several key aspects of the model. First, there is no explicit split between the perception, action and decision making in the proposed model. By means of the tightly coupled structures and spatio-temporal constraints imposed on the model, the robot was able to learn goal-directed action by developing multimodal representation in multiple levels in a coordinated dynamic process. Second, the experimental results showed that the proposed model can develop and facilitate a sort of working memory required for the tasks. In both experiments, the robot was able to maintain higher-level task-related information throughout the task phases. This is due to the recurrent connections at PFC and slow dynamics of PFC and M_S. It remains for the future studies to examine the model in more complex robotic tasks requiring other cognitive skills and modalities.

References

[1] J. Hwang, M. Jung, N. Madapana, J. Kim, M. Choi, and J. Tani, "Achieving "synergy" in cognitive behavior of humanoids via deep learning of dynamic visuo-motor-attentional coordination," in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids),* pp. 817-824, 2015.

[2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks,* vol. 61, pp. 85-117, 2015.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, pp. 1798-1828, 2013.

[4] O. Sigaud and A. Droniou, "Towards Deep Developmental Learning,", *IEEE Transactions on Autonomous Mental Development,* vol. PP, pp. 1-1, 2015.

[5] M. Jung, J. Hwang, and J. Tani, "Self-Organization of Spatio-Temporal Hierarchy via Learning of Dynamic Visual Image Patterns on Action Sequences," *PLoS ONE,* 2015.

[6] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment," *PLoS Computational Biology,* vol. 4, p. e1000220, 2008.

[7] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research,* vol. 9, p. 85, 2008.

[8] A. Di Nuovo, V. M. De La Cruz, and A. Cangelosi, "A Deep Learning Neural Network for Number Cognition: A bi-cultural study with the iCub," in *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob),* pp. 320-325, 2015.

[9] R. Nishimoto and J. Tani, "Development of hierarchical structures for actions and motor imagery: a constructivist view from synthetic neuro-robotics study," *Psychol Res,* vol. 73, pp. 545-58, Jul 2009.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[11] P. F. Dominey, M. Hoen, J. M. Blanc, and T. Lelekov-Boissard, "Neurological basis of language and sequential cognition: evidence from simulation, aphasia, and ERP studies," *Brain Lang,* vol. 86, pp. 207-25, Aug 2003.

[12] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes," *Neural Networks,* vol. 16, pp. 11-23, 2003.

[13] J. A. Starzyk and H. He, "Anticipation-based temporal sequences learning in hierarchical structure," *IEEE Trans Neural Netw,* vol. 18, pp. 344-58, Mar 2007.

[14] D. E. Rumelhart, J. L. McClelland, and P. R. Group, *Parallel distributed processing* vol. 1: MIT press, 1986.

[15] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems,* vol. 19, p. 153, 2007.

[16] N. G. Tsagarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, *et al.*, "iCub: the design and realization of an open humanoid platform for cognitive and neuroscience research," *Advanced Robotics,* vol. 21, pp. 1151-1175, 2007.

[17] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori, "An open-source simulator for cognitive robotics research: the prototype of the iCub humanoid robot simulator," presented at the Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, Gaithersburg, Maryland, 2008.

[18] D. Marocco, A. Cangelosi, K. Fischer, and T. Belpaeme, "Grounding Action Words in the Sensorimotor Interaction with the World: Experiments with a Simulated iCub Humanoid Robot," *Frontiers in Neurorobotics,* vol. 4, p. 7, 2010.

[19] P. Savastano and S. Nolfi, "A Robotic Model of Reaching and Grasping Development," *Autonomous Mental Development, IEEE Transactions on,* vol. 5, pp. 326-336, 2013.

[20] C. E. Vargas-Irwin, L. Franquemont, M. J. Black, and J. P. Donoghue, "Linking Objects to Actions: Encoding of Target Object and Grasping Strategy in Primate Ventral Premotor Cortex," *The Journal of Neuroscience,* vol. 35, pp. 10888-10897, 2015.

[21] R. E. Lasky, "The effect of visual feedback of the hand on the reaching and retrieval behavior of young infants," *Child Dev,* vol. 48, pp. 112-7, 1977.

[22] M. E. McCarty, R. K. Clifton, D. H. Ashmead, P. Lee, and N. Goubet, "How infants use vision for grasping objects," *Child Dev,* vol. 72, pp. 973-87, 2001.

[23] J. Carpaneto, M. A. Umilta, L. Fogassi, A. Murata, V. Gallese, S. Micera, *et al.*, "Decoding the activity of grasping neurons recorded from the ventral premotor area F5 of the macaque monkey," *Neuroscience,* vol. 188, pp. 80-94, 2011.