Published in Proc. of the Fourth Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-Epirob2014), Genoa, Italy, pp. 227-233, October 2014.

Multiple Spatio-Temporal Scales Neural Network for Contextual Visual Recognition of Human Actions

Minju Jung
KAIST
Daejeon, South Korea
minju.jung@kaist.ac.kr

Jungsik Hwang
KAIST
Daejeon, South Korea
jungsik.hwang@kaist.ac.kr

Jun Tani
KAIST
Daejeon, South Korea
tani1216jp@gmail.com

Abstract

This paper introduces a novel dynamic neural network model which can recognize dynamic visual image patterns of human actions based on learning. The proposed model is characterized by its capability of extracting the spatio-temporal feature hierarchy latent in the training visual image streams. The model achieves this property by integrating two essential ideas: (1) multiple spatial-scales processing and (2) multiple timescales processing, which have been introduced in the convolutional neural network (CNN) and the multiple timescale recurrent neural network (MTRNN), respectively. The evaluation of the model performance conducted by utilizing the Weizmann dataset showed that the proposed model outperforms other neural network models in recognition of a set of prototypical human movement patterns. Furthermore, additional evaluation testing for recognition of concatenated sequences of these prototypical movement patterns indicates that the model is endowed with a remarkable capability for contextual recognition of long-range dynamic visual patterns.

Index Terms—Convolutional neural network, deep learning, delay response manner, dynamic vision, multiple timescale recurrent neural network, self-organization, spatio-temporal hierarchy

I. INTRODUCTION

Human's cognitive competency to visually recognize others' actions should involve implementation of sophisticated mechanisms that utilize both compositional and contextual information processing in order to adequately manage massively high-dimensional spatio-temporal patterns [1-3]. For the purpose of exploring the possible underlying mechanisms for such cognitive competency, brain-inspired synthetic modeling has been considered to be one feasible approach to pursue. In the machine vision research community, deep learning schemes by utilizing particular neural network models [4-7] have attracted large attentions recently. Among such models, the convolutional neural network (CNN) [4], developed as inspired by the spatial hierarchical processing of visual features known in mammal visual cortical areas, has demonstrated remarkably superior recognition performance for static natural visual images as compared to other existing methods [8].

However, the original form of CNN cannot cope with dynamic visual image patterns efficiently because the model is merely a static input-output mapping system that does not comprise any temporal processing elements. In order to overcome this limitation, several studies have introduced some extensions of the CNN. For example, the 3D CNN proposed by Ji et al. [9]. As its name implies, 3D CNN can deal with spatio-temporal dimensions by replacing 2D convolution with 3D convolution, which convolves 3D kernels to the cube formed by stacking multiple contiguous frames together. Indeed, 3D CNN performed well on standard human action recognition datasets, specifically the TRECVID and KTH datasets [10], which are recognized well by simply extracting short-range temporal correlations [11]. However, the model cannot be applied to certain classes of visual recognition tasks that require extraction of contextual information or long-range temporal correlations in the visual image streams because the 3D CNN model can only maintain temporal information within the restricted temporal dimension of the cube. Baccouche et al. [12] has proposed a two-stage model to maintain temporal information in the entire sequence by adding the long short-term memory (LSTM) [13] as a second stage of the 3D CNN. But, spatial and temporal information processing are still not fully combined into one model.

In order to successfully recognize dynamic visual image patterns that are characterized by multiple scales properties both in spatial and temporal dimensions, the model should possess the capability of self-organizing adequate spatio-temporal hierarchy via iterative learning of the observed visual images. To satisfy this requirement, we propose a novel model, formed by integrating two essential ideas presented in different models: self-organization of spatial hierarchy by the CNN and temporal hierarchy implementation via the multiple timescale recurrent neural network (MTRNN), a dynamic neural network model, proposed by Yamashita and Tani [14]. More specifically, this new model, termed the multiple spatio-temporal scales neural network (MSTNN), assumes that the lower level network consists of the faster dynamics processing units permitting only shorter distance local connectivity among them, while the higher level network comprises the slower dynamics processing units that constitute the longer distance global connectivity. Also, the size of the receptive field at each level changes from relatively small in the lower level to large in the higher level, which is analogous to observed organization in the mammal visual cortices [15]. The assumed temporal hierarchy has also been evidenced in the human visual cortices [16]. The premise for this assumption is that functional hierarchy could be self-organized by utilizing both spatial and temporal constraints incorporated in the learning processes of massively high-dimensional spatio-temporal patterns present in dynamic visual images.

To evaluate performance of the MSTNN, we conducted two classes of recognition-by-learning tasks. The first experiment was conducted to evaluate performance of the MSTNN for recognition of the set of prototypical human movement patterns found in the Weizmann dataset [17]. With this dataset, the MSTNN showed performance comparable to that found in currently published reports. Even so, the MSTNN performed better than two baseline models: CNN and 3D CNN. The second experiment was performed to evaluate the capability of the MSTNN to recognize long-range visual images of combinatorial action sequences. The prototypical actions in the Weizmann dataset were concatenated to generate the learning and the testing sequences for this experiment. The experimental result indicated that the MSTNN also performed well in this recognition task. The analysis on this experiment indicated that the slow timescale dynamics in the MSTNN plays an important role in contextual information processing for the recognition of compositional visual image sequences. The next section will describe our proposed model of the MSTNN.

II. MODEL

The multiple spatio-temporal scales neural network (MSTNN) has a capability of self-organizing spatio-temporal feature hierarchy by implementing spatial constraints on the CNN that shares weights to reduce the number of learnable parameters, and temporal constraints on the MTRNN, which assigns faster timescale dynamics at lower levels and slower timescale dynamics at higher levels to yield dynamic neuronal properties, *i.e.* fast to slow timescale dynamics toward the higher level. The timescale at each level is determined according to the setting of the time constant parameter (τ) of leaky integrator type neurons allocated in the level. A smaller τ and a larger τ result in faster and slower timescale dynamics, respectively.

The top layer consists of a set of winner-take-all neural units, which represents the categorical outputs determining the recognition results for dynamic visual image patterns input contained in the bottom row image layer. The categorical outputs were trained with the targets in a delay response manner. This means that the target categorical outputs are presented immediately after each visual stream input is terminated. The learning processes employ a version of the back-propagation through time algorithm [18], which was adapted for the MTRNN model [14]. The core hypothesis assumed for the model is that the spatio-temporal hierarchy required for contextual recognition of dynamic visual image patterns could be self-organized by utilizing the multiple scales of spatial-temporal constraints imposed on the neural activity in the course of supervised training on a set of exemplars. The details of the model are described in the following sections.

A. Model Architecture

The MSTNN consists of multilayers of retinotopically organized neural units that represent multiple features at each retinotopic position at each level as like the CNN. Standard CNN has a subsampling layer between convolutional layers that extracts features invariant to small distortions and shifts, while significantly reducing computational complexity. However, the subsampling operation generates discontinuity in the value of the neuron in the subsampling layer through time. Therefore, the model followed another approach proposed by Simard et al. [19] which eliminates a subsampling layer by combining convolution and subsampling operations into one operation performed in the convolutional layer.

The network utilized in the later described experiments consists of 5 layers: one input layer (layer 1), three convolutional layers (layer 2-4), and one fully-connected layer (layer 5) (see Fig.1). Layer 1 is the input layer, which has only one feature map size of 48x54, and contains the raw input image. Layer 2 is a convolutional layer that has 6 feature maps of size 22x22 with step size (or "stride") of 2. Each feature map in layer 2 connects with the feature map in layer 1 through a kernel size of 6x12. These feature maps are encoded via dynamic activities of (22x22x6) leaky integrator neurons with their time constant τ set to 2.0. Layer 3 is a convolutional layer encompassing 50 feature maps of size 8x8 with a step size of 2 and time constant τ set to 5.0. Each feature map in layer 3 connects with each feature map in layer 2 through a kernel size of 8x8. Layer 4 is a convolutional layer that has 100 feature maps of size 1x1 with a step size of 1 and time constant set to 100.0. Each feature map in layer 4 connects with each feature map in layer 3 through a kernel size of 8x8. Layer 5 generates the categorical outputs encoded by a set of static neural units using the softmax activation function. The number of neurons in layer 5 is the same as the number of classes in the dataset. Each neuron in layer 5 is fully-connected with all neurons of the 100 feature maps in layer 4. Most activated neurons in layer 5 represent the categorization result. Details of the forward dynamics are explained in the following section.

B. Forward Dynamics

We used a leaky integrator model, by which each neuron's activity at each layer is calculated not only by convoluting its kernels with the corresponding feature maps in the previous layer, but also by adding its decayed internal state from the previous time step. The decay rate depends on the time constant τ. More specifically, the internal state and activation value

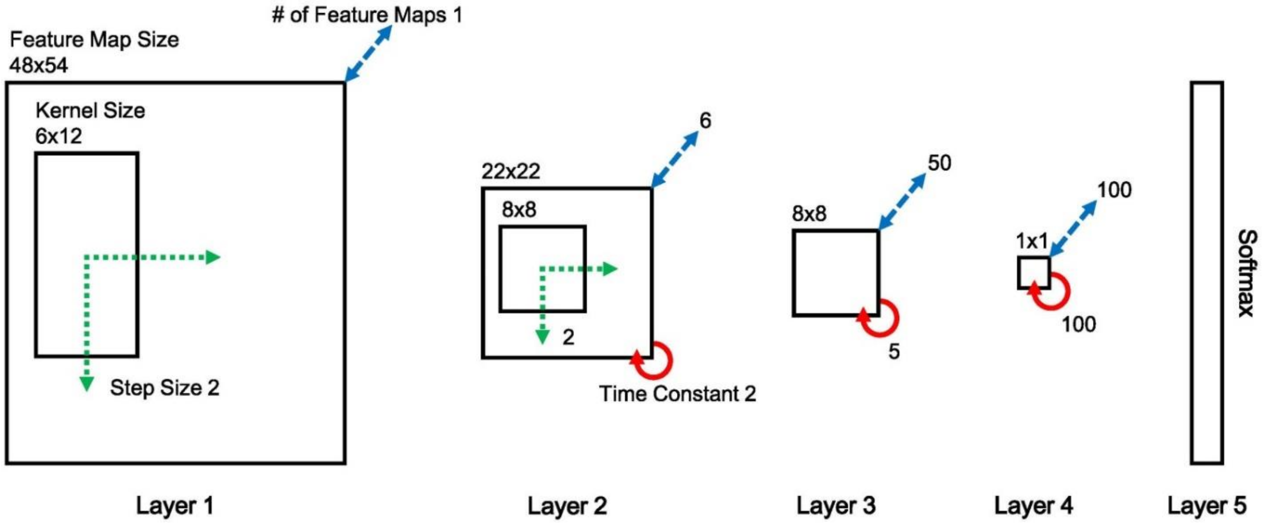


Fig. 1. Architecture of the MSTNN. The architecture consists of one input layer, three convolutional layers and one fully-connected layer. Each layer has a set of parameters: feature map size, kernel size, number of feature maps (blue dashed arrow), and step size (green dotted arrow). Only the convolutional layer has an additional time constant parameter (red solid arrow), which plays a key role in this model. The higher convolutional layer has a larger time constant than the lower convolutional layer. Layer 5: the softmax activation function used for classification.

of the neuron at position (x, y) in the mth feature map in the lth layer at time step t, denoted as u_{lm}^{txy} and v_{lm}^{txy}, respectively, are calculated by

$$u_{lm}^{txy} = \left(1 - \frac{1}{\tau_l}\right) u_{lm}^{(t-1)xy} + \frac{1}{\tau_l}\left(\sum_{n=1}^{N_{(l-1)}} \left(\mathbf{k}_{lmn} * \mathbf{v}_{(l-1)n}^{t}\right)_{xy} + b_{lm}\right) \tag{1}$$

$$\left(\mathbf{k}_{lmn} * \mathbf{v}_{(l-1)n}^{t}\right)_{xy} = \sum_{p=1}^{P_l}\sum_{q=1}^{Q_l} k_{lmn}^{pq} v_{(l-1)n}^{t(S_l(x-1)+p)(S_l(y-1)+q)} \tag{2}$$

$$v_{lm}^{txy} = \begin{cases} \dfrac{exp(u_{lm}^{txy})}{\sum_{n=1}^{N_l} exp(u_{ln}^{txy})} & l = L \\ 1.7159 \tanh(\frac{2}{3} u_{lm}^{txy}) & l \neq L \end{cases} \tag{3}$$

where L is the number of layers of the model, τ_l is the time constant of the lth layer, N_l is the number of feature maps in the lth layer, S_l is the step size of the lth layer, k_{lmn}^{pq} is the value at the position (p, q) of the kernel connected from the nth feature map in the previous layer to the current feature map, b_{lm} is the bias for the current feature map, and P_l and Q_l are the height and width of the kernel in the lth layer respectively.

By defining neuron and weight of the fully-connected layer as a feature map size of 1x1 and a kernel size of 1x1 respectively, and setting τ_l and S_l to 1, both equations for the convolutional and fully-connected layers are expressed as shown in Eq. (1), Eq. (2), and Eq. (8)-(10). Eq. (2) shows that the kernel is shifted by S_l pixels both in horizontal and vertical directions before convolution, in accordance with the Simard et al. [19] approach. In Eq. (3), the softmax activation function is applied to neurons only in the output layer, and neurons in the other layers are calculated using a scaled version of the hyperbolic tangent activation function, as recommended by LeCun et al. [20].

C. Training Method

The error function E is determined using Kullback-Leibler divergence defined as follows:

$$E = \sum_{t=1}^{T} E_t \tag{4}$$

$$E_t = \begin{cases} \sum_{m=1}^{N_L} \hat{y}_m ln\left(\dfrac{\hat{y}_m}{y_m^t}\right) & \text{if } T\text{ - }t < d \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where T is the length of the visual sequence, d is the length of the label sequence given immediately after each visual stream input is terminated, y_m^t is redefined from v_{Lm}^{t11}, which represents confidence of class m when provided with the visual sequence at time step t, in order to simplify notation, and $\hat{y}_m$ is the true label. If input visual sequence belongs to class c, $\hat{y}_{m=c}$ is set to 1 while the rest of the entries $\hat{y}_{m\neq c}$ are set to 0. Because the model is trained in a delay response manner, error is only generated in the final d time steps of the visual sequence.

Here, the learnable parameters of the network are denoted by $\boldsymbol{\theta}$. We implemented a stochastic gradient descent method during the training phase, where the learnable parameters of the network $\boldsymbol{\theta}$ are updated to minimize the error function defined in Eq. (4) after each visual sequence is given. Additionally, the kernel weights updating applied weight decay of 0.0005 in order to prevent overfitting [8].

$$k_n = k_{n-1} - \alpha \left\{ \left\langle \frac{\delta E}{\delta k} \right\rangle_{S_i} + 0.0005 k_{n-1} \right\} \tag{6}$$

$$b_n = b_{n-1} - \alpha \left\langle \frac{\delta E}{\delta b} \right\rangle_{S_i} \tag{7}$$

where α is the learning rate and $\langle \frac{\delta E}{\delta \boldsymbol{\theta}} \rangle_{S_i}$ is the average over the ith visual sequence S_i of the partial differential equation $\frac{\delta E}{\delta \boldsymbol{\theta}}$ for each learnable parameter, which can be solved with a conventional back-propagation through time method (BPTT) [18] by additionally considering their decaying effects determined by the time constant [14]. The BPTT is performed as follows.

$$\frac{\partial E}{\partial u_{lm}^{txy}} = \begin{cases} y_m^t - \hat{y}_m & l = L \\ \left(1 - \frac{1}{\tau_l}\right) \frac{\partial E}{\partial u_{lm}^{(t+1)xy}} + \frac{1}{\tau_{(l+1)}} \frac{\partial v_{lm}^{txy}}{\partial u_{lm}^{txy}} \sum_{n=1}^{N_{(l+1)}} \pi_{(l+1)nm}^{txy} & l \neq L \end{cases} \tag{8}$$

$$\text{where } \frac{\partial v_{lm}^{txy}}{\partial u_{lm}^{txy}} = \frac{2 \cdot 1.7159}{3} \left(1 - \tanh^2 \left(\frac{2}{3} u_{lm}^{txy}\right) \right)$$

$$\text{and } \pi_{lnm}^{txy} = \sum_{(\tilde{x},\tilde{y}) \in R_{lnm}^{xy}} k_{lnm}^{(x - S_l(\tilde{x}-1))(y - S_l(\tilde{y}-1))} \frac{\partial E}{\partial u_{ln}^{t\tilde{x}\tilde{y}}}$$

$$\frac{\partial E}{\partial k_{lmn}^{pq}} = \frac{1}{\tau_l} \sum_{t=1}^{T} \sum_{x=1}^{X_l} \sum_{y=1}^{Y_l} v_{(l-1)n}^{t(S_l(x-1)+p)(S_l(y-1)+q)} \frac{\partial E}{\partial u_{lm}^{txy}} \tag{9}$$

$$\frac{\partial E}{\partial b_{lm}} = \frac{1}{\tau_l} \sum_{t=1}^{T} \sum_{x=1}^{X_l} \sum_{y=1}^{Y_l} \frac{\partial E}{\partial u_{lm}^{txy}} \tag{10}$$

where R_{lnm}^{xy} indicates a set of the position $(\tilde{x}, \tilde{y})$ of the neurons in the nth feature map in the lth layer, that are affected by the neuron at position (x, y) in the mth feature map in the previous layer during forward dynamics, and X_l and Y_l are the height and width of the feature map in the lth layer, respectively.

To accelerate training, we used an adaptive learning rate method. If the mean square error (MSE) calculated subsequent to one epoch is smaller than the previous one then α is multiplied by 1.05, otherwise α is divided by 2.

All kernel weights and biases were initialized with randomly selected values from a zero-mean Gaussian distribution with standard deviation of 0.05. The initial state of each neuron in the convolutional layer u_{lm}^{0xy} was set to 0. The remaining parameters, including the number of layers, the number of feature maps, the size of feature map, the size of kernel, the step size, and the time constant, were set as described in Section II-A.

III. EXPERIMENT AND RESULTS

In order to examine characteristics of the MSTNN, we performed two classes of recognition-by-learning tasks. The first experiment was conducted to evaluate simple prototypical action recognition performance of the model using the Weizmann dataset. The second experiment was conducted to evaluate the recognition performance for long-range action sequences by preparing a set of concatenated sequences of the prototypical action patterns in the Weizmann dataset. When the MSTNN was trained in each experiment, 15 black frames were added at the end of the all visual sequences from the dataset to give label sequence in a delay response manner. In testing, the video sequences were recognized by majority voting of the recognition results generated when black frames were introduced to the MSTNN.

For the evaluation protocol, we used leave-one-subject-out cross-validation for both experiments. If we assume the number of the subject is N, $N-1$ subjects are selected for training, the remaining subject is selected for testing. The

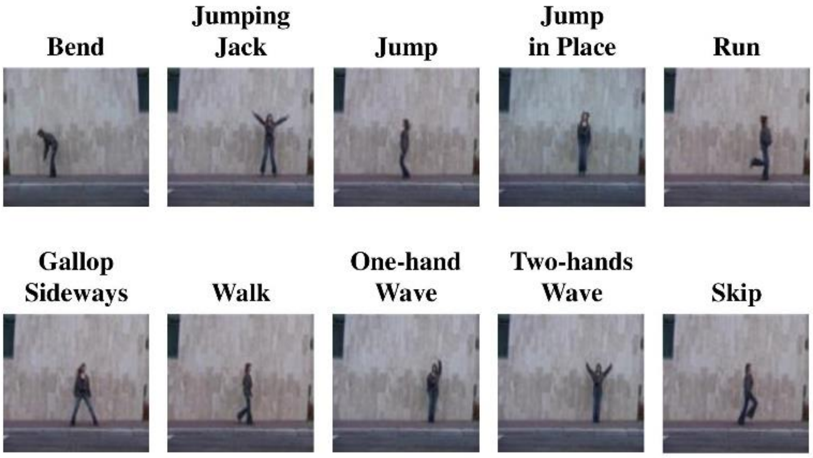


Fig. 2. A sample of actions from the Weizmann dataset.

recognition performance corresponds to the average of N trials, each selecting a different subject for testing.

A. Prototypical action recognition on the Weizmann Dataset

The first experiment evaluates the basic performance of the MSTNN to perform recognition-by-learning on a set of relatively simple prototypical human actions. The experiment utilizes the Weizmann dataset [17], which contains 10 human actions (walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in the place, jumping jack and skip), each being performed once by 9 subjects (see Fig. 2). The view point and background are static. In the experiment, we used foreground silhouettes by background subtraction using background sequences. We normalized all silhouette frames into the same size of 48x54, and converted those into a 2592 dimensional vector in a raster scan manner. The performance of the MSTNN and two baseline models (CNN, 3D CNN) were evaluated using exactly the same conditions, with the following exclusions: (1) the recognition of the two baseline models was done using general majority voting, which encompasses recognition results over the entire time span of the input, and (2) the size of the temporal dimension of the cube was set to 7 for the 3D CNN, as used by Ji et al. [9]. The recognition accuracy of the MSTNN and two baseline models are shown in Table I along with published accuracies of the leave-one-subject-out cross-validation for the evaluation protocol.

The MSTNN achieves an accuracy comparable with published accuracies (97.78%), although we cannot directly compare to published accuracies because they performed experiments under different conditions in terms of data preprocessing, parameter settings, and so on. The accuracy of the MSTNN is superior to those of the two baseline models: 3D CNN (96.67%) and CNN (92.22%). Accuracy difference between MSTNN and CNN is quite large compared to that of the 3D CNN. It is consistent with [9] that considering temporal information improves the recognition performance since all capabilities of spatial feature extraction of the three models are the same.

B. Recognition of sequentially combined prototypical actions

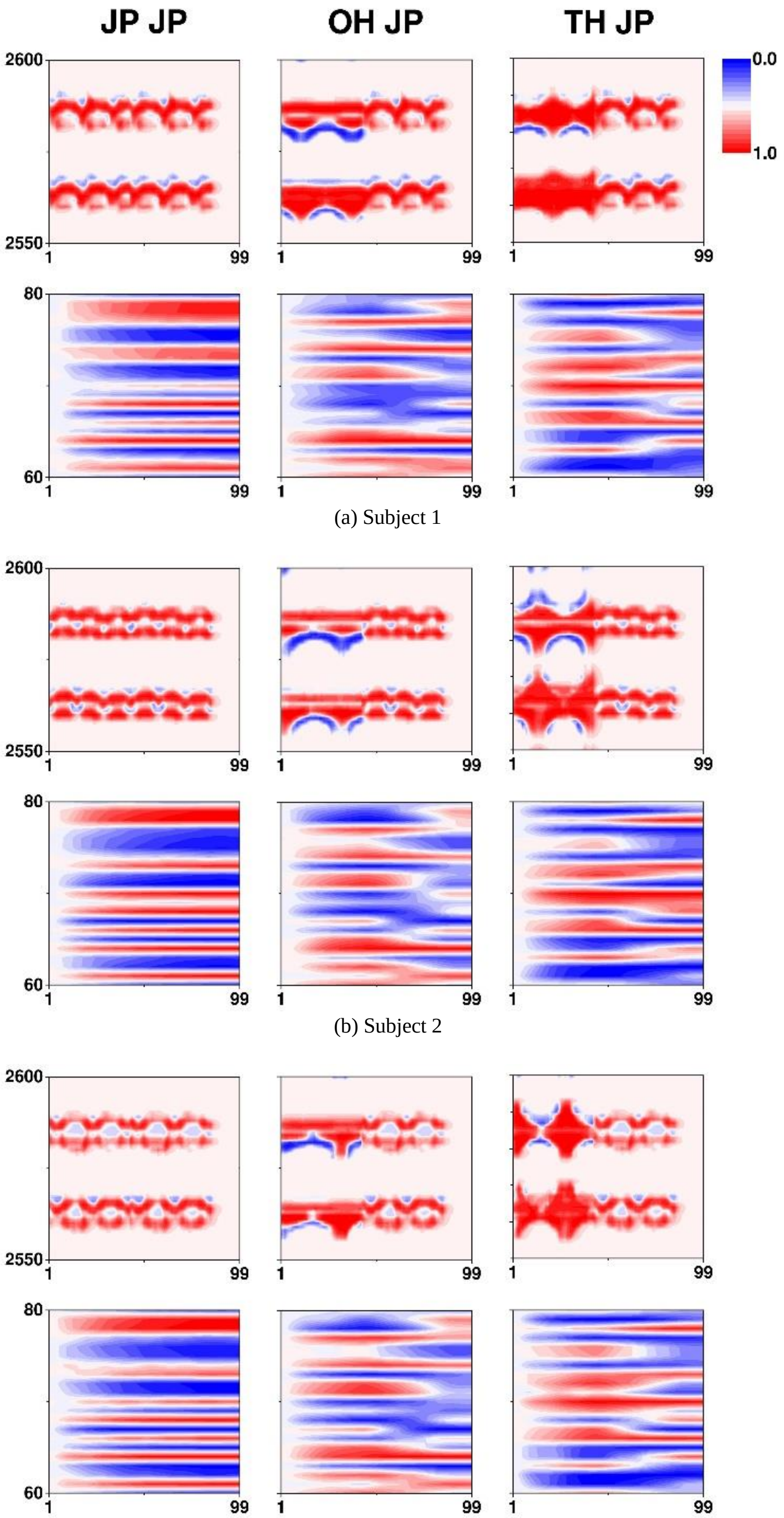
In the second experiment, the MSTNN is tested with visual images containing sequential combinations of prototypical human action patterns for the purpose of examining its capability for contextual recognition of long-range dynamic visual image patterns. For this purpose, visual image patterns of 9 action sequences were synthesized by concatenating two prototypical actions out of three (jump in the place (JP), one-hand wave (OH), and two-hands wave (TH)) in the Weizmann dataset for all possible combinations for each subject. The number of frames per concatenated prototypical action is 42 frames.

TABLE I
COMPARISON OF ACCURACIES ON THE WEIZMANN DATASET

Method	**Accuracy**
MSTNN	97.78%
3D CNN	96.67%
CNN	92.22%
Bregonzio et al. [21]	96.66%
Sun et al. [22]	97.8%
Weinland and Boyer [23]	100%
Zhang et al. [24]	92.89%

The MSTNN was then trained and tested for recognition of 9 categories of the combinatorial action patterns with the aforementioned leave-one-subject-out cross-validation for the evaluation protocol. The experiment results indicate that the recognition accuracy was 85.19%. Comparative studies with CNN and 3D CNN were not made in this experiment because it is obvious that CNN, without any temporal processing mechanisms, cannot cope with the current recognition task, and 3D CNN is limited to a small number of contiguous video frames as reported in [9].

The task of recognizing the combinatorial action sequences is not trivial because the network has to keep the memory of the first actions perceived when it generates the categorical outputs for the whole combinatorial action sequences at the end in the delay response manner. In order to clarify the underlying mechanism, we conducted analysis of the internal dynamics. Fig. 3 shows time developments of 50 representative neural units' activity in layer 2 with a smaller time constant ($\tau_2 = 2.0$), and 20 representative neural units' activity in the layer 4 with a larger time constant ($\tau_4 = 100.0$) for the cases of three different combinatorial actions (JP → JP, OH → JP, and TH → JP) done by three different subjects.



(a) Subject 1

(b) Subject 2

(c) Subject 3

Fig. 3. Time developments of the internal dynamics during recognition of the concatenated action patterns. The neural activities are shown for those in layer 2 $(\tau_2 = 2.0)$ and layer 4 $(\tau_4 = 100.0)$. In each plot, the vertical axis represents the indices of the neurons and horizontal axis represents time steps. The plots show 50 over 2904 encoded neuron's activities in the layer 2 (first row) and 20 over 100 neuron's activities in the layer 4 (second row) during JP → JP (first column), OH → JP (second column), TH → JP (third column) demonstrated by three different subjects for (a), (b) and (c). Activities are mapped to the range from 0 to 1.

The neural activity in layer 2 showed detailed rhythmic patterns that are assumed to be correlated with cyclic movements in the actions, such as jumping or waving hands repeatedly. Conversely, the neural activity in layer 4 showed steady patterns during the period of movement repetition same movements repeated but changed drastically to other steady patterns when the actions were altered. More importantly, among-subjects-variances of layer 4 activity were shown to be smaller than those of layer 2 activity for the same combinatorial actions. Furthermore, it was interesting to see that layer 4 activity was significantly different for even the same action of "jump in place" each time it was preceded by a different

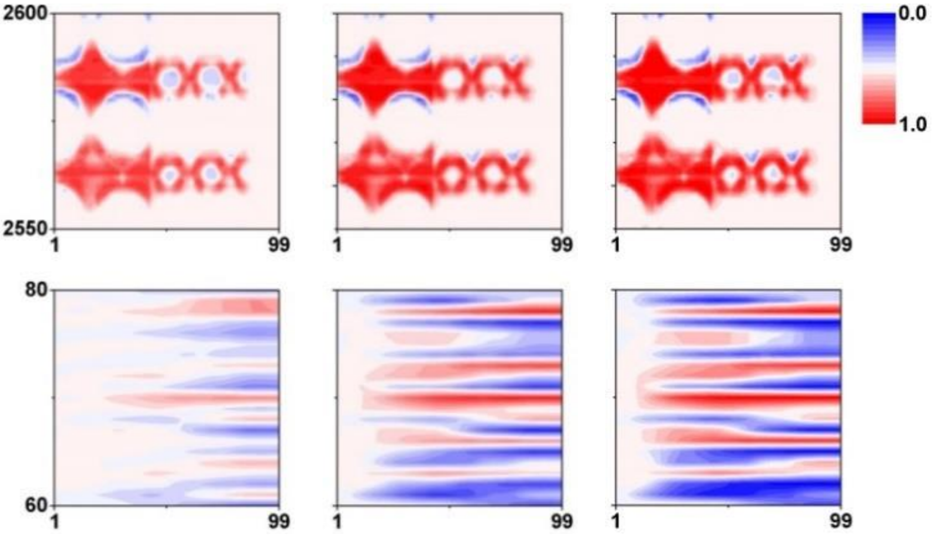


Fig. 4. Developmental processes of the different timescale dynamics through learning during combinatorial action (TH → JP) demonstrated by one subject. From left to right column indicates 5, 10, 50 epoch respectively. The rest of the formats are the same as Fig. 3.

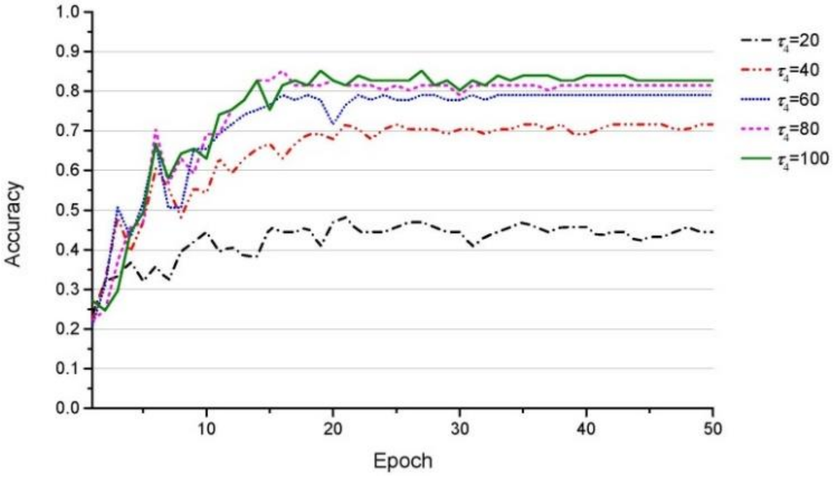


Fig. 5. Development of recognition accuracy with different time constants assigned for the layer 4. The vertical axis represents the accuracy obtained from leave-one-subject-out cross-validation and horizontal axis represents epochs during training phase. By changing the time constant from small $(\tau_4 = 20.0)$ to large $(\tau_4 = 100.0)$ stepping by 20, the accuracy is largely increased.

action. It can be summarized that layer 4 activity is self-organized such that the activation patterns in layer 4 at the end of perceiving the same combinatorial actions become quite similar among different subjects while at the same time can be differentiated even when perceiving the same action using the context of perceiving different actions in the past. This explains how the contextual recognition with the inter-subjects generalization could be achieved in the current task.

We also analyzed the developmental processes of the different timescale dynamics at different levels through learning. Fig. 4 shows the internal dynamics in layer 2 and 4 at epoch 5, 10, 50, respectively, for the case of recognizing combinatorial action (TH → JP) demonstrated by one subject. The fast timescale dynamics in layer 2 at epoch 5 is very similar to the one at epoch 50. However, the slow timescale dynamics in layer 4 at epoch 5 is much less organized when compared to the one at epoch 50. These results imply that the slow timescale dynamics in the higher level develops after the fast timescale dynamics in the lower level develops. These results are in agreement with previous results on MTRNN learning [25].

The observation of dynamic neural activity in layer 2 and layer 4 suggests that the slow timescale dynamics in layer 4 may play an essential role in the contextual recognition of the visual image. Next, we examined the contribution of the slow timescale dynamics to the success of the contextual recognition with quantitative measure. The recognition performance of the model was tested by changing the time constant in the fourth layer from 20 to 100, while maintain the original time constants for the lower layers. The experiment result is shown in Fig. 5, where it can be seen that the accuracy deteriorates as the time constant value is reduced from 100 to 20.

IV. Conclusion

The current paper proposed a novel dynamic neural network model that can recognize complex dynamic visual image patterns by means of self-organizing adequate spatio-temporal hierarchy via utilization of both spatial and temporal constraints imposed on the learning processes of the exemplar visual patterns.

For evaluating the performance of the model, two types of human action recognition experiments were conducted.

The first experiment evaluated the capability for recognizing a set of prototypical actions by utilizing the Weizmann dataset. The experimental results showed that the MSTNN outperforms other baseline models in the recognition accuracy. The second experiment tested the capability for recognizing long-range visual images containing sequential combinations of the prototypical human actions in the Weizmann dataset. The experiment results showed that the MSTNN possesses remarkable capability for contextual recognition of such long-range visual image patterns. The analysis of the internal neural activity revealed that the development of the slow timescale dynamic neural activity in the higher level contributes to the success of the contextual recognition.

Future research will investigate the scaling property of the model in recognition of more complex dynamic visual images of human actions such as object-directed actions and human-to-human interactions.

ACKNOWLEDGMENT

This work (2014R1A2A2A01005491) was supported by Mid-career Researcher Program through NRF grant funded by the MSIP.

REFERENCES

[1] S. Schaal, "Is Imitation Learning the Route to Humanoid Robots?," *Trends Cognit. Sci.*, 1999.

[2] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by Watching: Extracting Reusable Task Knowledge from Visual Observation of Human Performance," *Robot. Autom.*, 1994.

[3] J. Triesch and C. von der Malsburg, "A System for Person-Independent Hand Posture Recognition against Complex Backgrounds," *PAMI*, 2001.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE*, 1998.

[5] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput*., 2006.

[6] M. Ranzato, H. Fu Jie, Y. L. Boureau, and Y. LeCun, "Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition," In *CVPR,* 2007.

[7] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations," In *ICML*, 2009.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," In *NIPS*, 2012.

[9] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," *PAMI*, 2013.

[10] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: A Local SVM Approach," In *ICPR*, 2004.

[11] K. Schindler and L. Van Gool, "Action Snippets: How many frames does human action recognition require?," In *CVPR*, 2008.

[12] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential Deep Learning for Human Action Recognition," In *Human Behavior Understanding*, 2011.

[13] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks," *J. Mach. Learn. Res.*, 2003.

[14] Y. Yamashita and J. Tani, "Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment," *PLoS Comput. Biol.*, 2008.

[15] D. H. Hubel, *Eye, Brain, and Vision*. New York, 1988.

[16] U. Hasson, E. Yang, I. Vallines, D. J. Heeger, and N. Rubin, "A Hierarchy of Temporal Receptive Windows in Human Cortex," *J Neurosci*, 2008.

[17] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," In *ICCV*, 2005.

[18] D. E. Rumelhart and J. L. McLelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cogniton*. MIT Press, 1986.

[19] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," In *ICDAR*, 2003.

[20] Y. LeCun, L. Bottou, G. Orr, and K.-R. Müller, "Efficient BackProp," *Lecture Notes Comput. Sci.*, 1998

[21] M. Bregonzio, G. Shaogang, and X. Tao, "Recognising Action as Clouds of Space-Time Interest Points," In *CVPR*, 2009

[22] X. Sun, M. Chen, and A. Hauptmann, "Action Recognition via Local Descriptors and Holistic Features," In *CVPR*, 2009.

[23] D. Weinland and E. Boyer, "Action Recognition using Exemplar-based Embedding," In *CVPR*, 2008.

[24] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, "Motion Context: A New Representation for Human Action Recognition," In *ECCV*, 2008.

[25] R. Nishimoto and J. Tani, "Development of hierarchical structures for actions and motor imagery: a constructivist view from synthetic neurorobotics study," *Psychol. Res.*, 2009.