Computer Science and Information Systems 12(2):787–799 DOI: 10.2298/CSIS141114026Z

Human-level Moving Object Recognition from Traffic Video

Fei Zhu[1,2], Quan Liu[1,2], Shan Zhong[1], and Yang Yang[3]

[1] School of Computer Science and Technology, Soochow University
Shizi Street No.1 Box 158, Suzhou, China, 215006
{zhufei, liuquan, 20134027005}@suda.edu.cn
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China, 130012
[3] Lund University, Box 117, 221 00 Lund, Sweden
yang.yang@med.lu.se

Abstract. Video preserves valuable raw information. Understanding these data and then recognizing objects and tagging them are crucial to intelligent planning and decision making. Deep learning provides us an effective way to understand big data with a human-level. As traffic video is characterized by crowded scene and low definition, it will be non-effective to deal with the whole image once. An alternative way is to separate image and determine a small window for each moving object. A Q-learning based moving object recognition approach, which firstly finds out moving object region and then uses a Q-learning based optimization method to determine the most compact region that contain the moving object, is proposed. The algorithms enable to detect the most compact rectangle around the moving object at near real-time speed. After that, a deep neural network is used to semantic tag the recognized objects. The experiment results show the algorithms work effectively.

Keywords: Q-learning, deep learning, moving object recognition, traffic video, big data.

1. Introduction

Big data [1] era calls for novel data processing patterns and models with more power of decision-making, insight and business process optimization ability so that they are competent to deal with mass, high rate growing and diverse data. In recent years, traffic video data have rapidly accumulated due to the application of traffic control surveillance equipment. The traditional manual video analysis cannot satisfy the need brought by such rapidly increasing data. Thereby, how to manage and make good use of these video data have attracted more and more attention as intelligent control applications based on video data rely on analysis of video content by extracting key information from the video for successive processing. Although some algorithms have been researched more than a decade, it remains confronted with many severe challenges, one of which is the foreground detection in crowded scenes. In such scenes, most foreground objects keep still, which results in inefficiency detection.

Reinforcement learning [2] provides a framework to learn directly from the interaction and achieve goals. Reinforcement learning framework is abstract, flexible, and can be applied in many different applications. The temporal difference (TD) [3] learning is capable of learning directly from raw experience without determining dynamic model of environment in advance. Moreover, the model learned by temporal difference is updated by estimation which is based on part of learning rather than final results of the learning. TD is particularly suitable for solving the prediction problems and control problems in real-time control applications. Q-learning [4] is an off-policy version of TD control. Therefore by taking advantage of Q-learning, we can accurately identify objects from traffic video which is generally of low resolution and with much noise. At the same time, as Q-learning is capable of reducing the computational complex, the identification speed reaches real-time level.

Deep Learning [5-9] is a new area of machine learning research, which has been introduced with the objective of moving machine learning closer to artificial intelligence. The motivation of deep Learning is to build a neural network which is capable of simulating the mechanism of analyzing and learning of human brain. Deep learning essentially treats learning hierarchy as a learning network; the unsupervised learning is used for pre-training and training output of a lower layer is used as the input of a higher layer; the supervised learning is used to fine tune the constructed network. Through feature transform in each layer, the feature representation of sample will be mapped to a new features space, which makes it easy to achieve classification and prediction. Compared with manual methods, the deep learning methods can fully take advantage of big data to learn the features, thus being able to better depict inside information of data.

In this work, we propose a Q-learning based object recognition approach which is able to deal with the hazy scenes of traffic video as well as crowd scenes of traffic video. The recognition algorithm also meets real time speed requirement. Then we utilize deep neural network to label semantic tag on the recognized objects.

2. Principles of Human Vision and Deep Learning

2.1. Principles of Human Vision

After the light signal is perceived by the human retina, the detected signal will be transmitted backward. The signal will be separated in optic chiasma and be transmitted backward. After separation, the left neural glial cells only transmit right visual signal, and the right neural glial cells transmit left visual signal, respectively through lateral geniculate nucleus. Signals are transmitted to the primary visual cortex and higher-level visual cortex. The objects perceived by human retina pass through the pathway and will be mapped to visual cortex on spatial relations [10]. The procedure is called topological maps of the retina, also known as retinotopy [11-12].

David Hubel [13] found the visual cortex is actually hierarchical. From primary cortical to beginner high-level visual cortex, visual information is transmitted step by step. The brain's primary visual cortex looks for simple features such as edges while the advanced level of visual cortexes understand more and more complicated and abstract contents, from model to specific objects, to features of objects and relationships of objects. The flow of hierarchical transmission of information from primary cortical to beginner high-level visual cortex in the human brain is showed as Figure 1 [14-17].

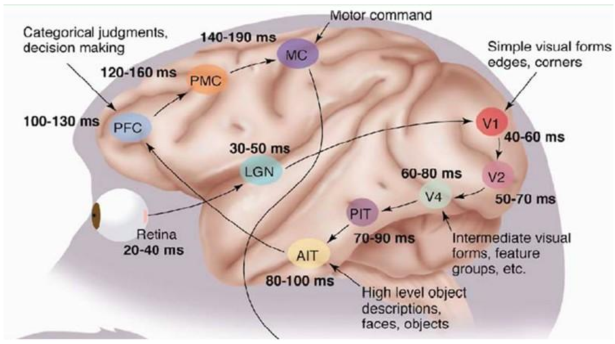

Fig. 1. Hierarchical transmission of information from primary cortical to beginner to high-level visual cortex in human brain

2.2. Deep Learning

Machine learning is a discipline that focuses on studying how to simulate or implement human behavior with computers in order to acquire new knowledge or skills and reorganize existing knowledge structure to continuously improve its performance. The basic workflow and subtasks of machine learning are as Figure 2.

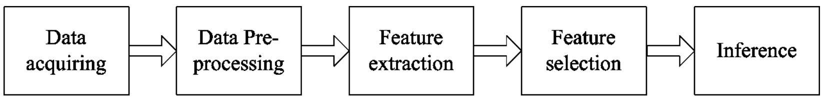

Fig. 2. Basic workflow and subtasks of machine learning

Initially the machine learning system acquires raw data which usually contain noise, and then generally carries on some processing so that the data are fit for successive subtasks. After that, the system will extract and select features for prediction. As for most machine learning systems, features are curial to algorithm performance, features of learning phase are manually selected, which is a very long and costly costive work and thus is unable to deal with big data.

In 2006, Hinton [5] proposed an effective approach to establishing a multilayer neural network with unsupervised data. The method was divided into two steps: training a network layer and optimizing parameters of network so that the generated representation of information is as the same as the original data as possible. Initially, a single layer of the neural network is trained in turn until the network is constructed. A wake-sleep algorithm, including wake phase and sleep phase, is then used to optimize the network. The wake phase is a cognitive process, which generates an abstract representation of each layer through external features and cognitive weight, and tunes generation weights by using the gradient descent. The sleep phase is a generation phase, which generates a top layer representation and downward, and adjust upward weights of the layer at the same time. The upward weights are used to reorganization and the downward weights are used to generation. The requirement of consensus of reorganization and generation guaranteed the top layer to present as many nodes as possible which are able to correctly restore the underlying layer. For example, a top node is used to denote an image of the cat, then all the image of the cat should activate the node, and downward generated image is supposed to able to be a rough cat image.

The fundamental idea of deep learning is to stack multiple layers, which means that the output of the lower layer is treated as the input of the upper layer. In this way, it is able to achieve hierarchical representation of information. Given a system S which has n layers (S_1, ... S_n) as well as input I and output O, the processing flow is represented as I->S_1->S_2->…->S_n->O. If output O is equal to input I, which means there is no information loss after being processed by the system. That is, for each layer, the output O is another representation of the input I as the information keeps unchanged. In deep learning, we need learn features automatically. By adjusting parameters of the system S such that the output is the same as the input, we can automatically obtain features, S_1,..., S_n, of input of each layer.

The work of Hinton showed that artificial neural network with multiple hidden layers has excellent feature learning capacity, by which the obtained features are characterized by more natural representation of data, thus conducive to visualization and classification; the difficulty of training of deep neural network can be effectively through layer-wise pre-training.

3. Reinforcement Learning

3.1. Introduction to Reinforcement Learning

Reinforcement learning is based on the idea that the system learns directly from the interaction during the process of approaching the goals. The reinforcement learning framework has five fundamental elements: a controller, environment, state, reward, and action [18], showed as Figure 3. The controller, which learns knowledge by interacting with outside environment and then chooses an action in accordance with the decision made by established controlling model, is the agent of the system; accordingly, the state will then be changed; the environment will return a reward to evaluate the action taken.

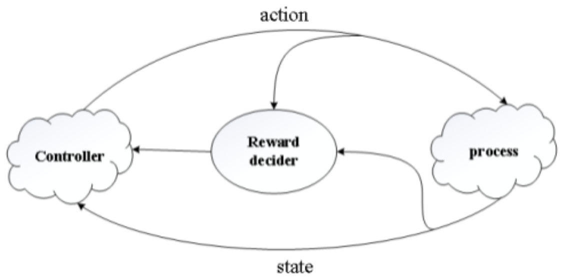

Fig. 3. Framework of reinforcement learning. Controller selects an action; the environment responds to the action, generates new scenes to the agent, and then returns a reward

The reinforcement learning provides a straight forward framework of the problem of learning from interaction to achieve a goal. It introduces an approach to learning in an unknown environment. During each episode of reinforcement learning model, agent chooses an action from the available action set and receives a reward. At each time step k, the controller selects an action $u_k\in U$ from action space U. As a result, the state changes to $x_{k+1}\in X$ from $x_k\in X$ in accordance with a transition function probability f: $X\times U\rightarrow[0,\infty)$:

$$x_{k+1}= f(x_k ,u_k) \tag{1}$$

The controller attains a reward r_{k+1} according to rewarding function $\tilde{\rho}$:

$$r_{k+1}=\tilde{\rho}\left(x_k,u_k,x_{k+1}\right) \tag{2}$$

The state-action value function Q^{π}: $X\times U\rightarrow$R of some policy π yields the return, a long-term reward, from a starting state:

$$Q^{\pi}\left(x,u\right)=\sum_{k=0}^{\infty}\gamma_k\rho\left(x_k,u_k\right) \tag{3}$$

where $\gamma\in[0,1]$ is a discount rate which shows how far sighted the controller is in considering the rewards and is also a factor for increasing uncertainty on future rewards.

Q-learning is an off-policy version of TD control, which is defined by

$$Q\left(x_k,u_k\right)\leftarrow Q\left(x_k,u_k\right)+\alpha\left[r_{k+1}+\gamma\max_u Q\left(x_{k+1},u\right)-Q\left(x_k,u_k\right)\right] \tag{4}$$

3.2. Markov Decision Process

A state signal which retains all relevant information is Markov, or has the Markov property. A Markov decision process (Markov decision process, MDP) is often used to model the reinforcement learning problem [19]. Usually a MDP model can be represented by a tuple M = <X, U, f, ρ, γ>, where X is the state space, U is the action space, f: $X\times U\to[0,1]$ is the state transfer function, ρ: $X\times U\to\mathbf{R}$ is the reword function and γ is the discount factor.

In the reinforcement learning framework, the controller interacts with the environment, gets the representation of environment denoted by state $x_k\in X$, and then chooses an action $u_k\in U$ according to its policy h: $X\to U$ such that $u_k = h(x_k)$, where u is all available actions. By taking the action, the controller will attain an immediate reward $r_{k+1} = \rho(x_k, u_k)$ and gets to a new state x_{k+1}. In the long term of trying, learning and optimizing, the controller will try to get the maximal sum of the rewards suggesting the optimal action sequence.

In a stochastic case, the state transition function is $\tilde{f}$: $X\times U\times X\to[0,\infty)$, and the reward function is $\tilde{\rho}$ $X\times U\times X\to\mathbf{R}$. At any time step k, the controller takes action u_k according to the control policy h: $X\to U$ in the state x_k, and then the system transfers to the next state x_{k+1} with the probability $P(x_{k+1}\in X_{k+1}\,|\,x_k,u_k)=\int_{X_{k+1}}\tilde{f}(x_k,u_k,x')dx'$ and obtains an immediate reward $r_{k+1}=\tilde{\rho}(x_k,u_k,x_{k+1})$. If the state space is countable, the state transition function is $\bar{f}$: $X\times U\times X\to[0,1]$, the transfer probability to the next state is $P(x_{k+1}=x'\,|\,x_k,u_k)=\bar{f}(x_k,u_k,x')$. Given f and ρ, as well as the current state x_k and the current action u_k, which are used to determine the next state x_{k+1} and the immediate reward r_{k+1}.

A terminal task is regarded as absorbing, which means the agent cannot move out of the state once it transfers to it. The task that has a terminal state is called an episodic task; a task which doesn't have any terminal state is called a continuous task. In particular, the problem solved by an MDP model is ergodic if it doesn't have any unreachable state or non-accessible state. Reinforcement learning is intended to get a solution solving a stable policy h: $X\times U\to[0,1]$ that will not change over time. The value of $h(x, u)$ refers to the probability of taking the action u under state x. If h is a deterministic strategy, in any state x, agent can get a deterministic action $u = h(x)$. We can use value function V_h and action value function Q_h to evaluate the merits of the policy, where $V_h(x)$ is the expected accumulative reword under state x by policy h, and Q_h is the expected accumulative reword under current state-action (x, u) by policy h.

$$V^h(x)=\sum_{u\in U}h(x,u)[\rho(x,u)+\gamma\sum_{x'\in X}f(x,u,x')V^h(x')] \quad (5)$$

$$Q^h(x,u)=\rho(x,u)+\gamma\sum_{x'\in X}f(x,u,x')\sum_{u'\in U}h(x',u')Q^h(x',u') \quad (6)$$

3.3. Q-learning

The temporal difference (TD) learning is capable of learning directly from raw experience without determining dynamic model of environment in advance. Moreover, the model learned by temporal difference is updated by estimation which is based on part of learning rather than final results of the learning. These two characteristics of temporal difference make it particularly suitable for solving the prediction problems and control problems in real-time control applications. Given some experience with policy *h*, temporal difference learning updates estimated V of V_h [19], as

$$V(x_k) \leftarrow V(x_k) + \alpha\left[R_k - V(x_t)\right] \quad \textbf{(7)}$$

where R_k is actual return after time step k, α is a step size parameter. Temporal difference learning updates V in step k+1 using the observed reward r_{k+1} and estimated $V(x_{k+1})$.

Let $Q^h(x, u)$ be the value of taking action u, in U under a policy. $Q^h(x, u)$ [20] is defined as

$$Q^h(x,u) = \rho(x,u) + \sum_{k=1}^{\infty} \gamma^k \rho(x_k, u_k) \quad \textbf{(8)}$$

Q-learning is an off-policy version of TD control, which is defined by

$$Q(x_k,u_k) \leftarrow Q(x_k,u_k) + \alpha\left[r_{k+1} + \gamma \max_u Q(x_{k+1},u) - Q(x_k,u_k)\right] \quad \textbf{(9)}$$

Q-learning is an off-policy version of TD control. The ultimate goal of reinforcement learning is to get an optimum strategy h^*. The corresponding value function $V_h(x)$ and action-value function $Q_h(x,u)$can be represented as

$$V^*(x) = \max_{u \in U}\{\rho(x,u) + \gamma \sum_{x' \in X} f(x,u,x') V^*(x')\} \quad \textbf{(10)}$$

$$Q^*(x,u) = \rho(x,u) + \gamma \sum_{x' \in X} f(x,u,x')\{\max_{u' \in U} Q^*(x',u')\} \quad \textbf{(11)}$$

4. Moving Objects Recognition and Semantic Tagging

In this work, we propose a Q-learning based moving objects recognition algorithm, which be divided into two steps, finding out object rectangle and detecting it, with frame difference approach.

4.1. Moving Object Region Recognition

The basic idea of moving object detection with frame difference [20] that the static objects will be removed by subtraction of image sequences and then moving object will be captured. In this work, we introduce a two stepwise approach to detect the fittest moving object region. The first step is to find out moving object region and then a Q-learning based optimization method is used to determine the fittest moving object which in fact is the most compact region that contain the moving object. The Moving object region detection with frame difference is as algorithm 1.

Algorithm 1:Moving Object Region Detection with Frame Difference
Input: image sequences *Is* with moving object
Output: set of rectanges *rt* for each moving object

1: initialize testing set *T* from *Is*
2: initialize moving object region *mor*
3: randomly select a unprocessed region *rt*
4: **do**
5:   get check frame *cf* for frame subtraction
6:   compare the region with *cf* to determine whether *rt* is part of the moving object region
7:   **if** *rt* is part of the moving object region **then**
8:     add *rt* to *mor*
9:   **else**
10:     go to step 4 and select another unprocessed region again
11:   **end if**
12:   compare the similarity of rt with each region *reg* **in** *mor*
13:   *r* ←highest similarity of *rt* with *reg* in mor
14:   **if** *r* is greater than the predefined threshold **then**
15:     merge *rt* and *reg*
16:     go to step 4
17:   **end if**
18: **until** *Is* are all processed
19: **return** set of rectanges *rt*

Then we use Q-learning to obtain a most compact rectangular outline of the object to be detected. A successful identification of eligible outline can be viewed as the process of finding an optimal strategy by Q-learning. The procedure is divided into two states. In the first stage, we can use a triple <*A*, *S*, *R*> to represent the Q-learning system, where *A* denotes the action that determines which part of region will be included, *S* denotes how much of current image has finished detection, and *R* is the feedback to the detector: if the included part is really part of the object, the agent will give out a positive reword; if the included part is really part of the object, the agent will give out a negative reword. Determining the fittest rectangle which contains the moving object is as algorithm 2.

Algorithm 2: Moving Object Rectangle Determining by with Q-learning
Input: image rectangle *rt* of moving object
Output: fittest rectange *frt* for moving object

```
initialize Q(x,u) arbitrarily
repeat
 for each episode
   initialize x arbitrarily
   initialize u arbitrarily
   select u from x' using ε-greedy policy with most compact
   take action u, and observe r, x'
   Q(x_k,u_k) ← Q(x_k,u_k) + α[r_{k+1} + γ max_u Q(x_{k+1},u) − Q(x_k,u_k)]
   x ← x'
  end for
 until terminated
return frt
```

4.2. Semantic Tagging with Deep Learning

In video, the reward of reinforcement learning takes form of external reward. However, as the environment cannot directly figure out the value of external reward, the external reward can only adaptively be constructed according to video data and corresponding semantic information. Since video data are characterized by high dimension of feature vector, it tends to lead to gradient diffusion, low network learning efficiency, and falling to local optimal. Deep neural network is a multiple neural network learning model based on deep learning, which learns some more useful features by constructing a machine learning model with many hidden layers and huge number of training data, in order to improve the accuracy of classification and prediction. One of the benefits of multiple layers of is able to represent a sophisticated function with fewer parameters.

The network model learning phase includes pre-training phase and fine-tuning phase. In the pre-training phase sample data without tags will be utilized. After setting the objective, the system will carry on approximating between input and out of each layer by unsupervised learning, thus obtaining initial distribution of network parameter. During the fine-tuning phase, the system optimizes parameters of the network obtained in the pre-training phase by supervised learning, so that the probability distribution of data for the unsupervised learning keeps consistent with that of data for the supervised learning.

Agent obtains new video data through constantly perceiving environment. We need set up a semantic deep learning neural network to map the image to the tag so that the images with the same semantic tag can be clustered to a class. We also need a map mechanism to map the semantic tag to external reward such that different video data have corresponding external reward. Because auto-encoder has completed automatically unsupervised learning during the video data feature automatic extraction phase, the trained auto-encoder can be used as the first part of the network. And on the base of it, we add a hidden layer and output layer, thus constituting a combined three-layered neural network, as Figure 4.

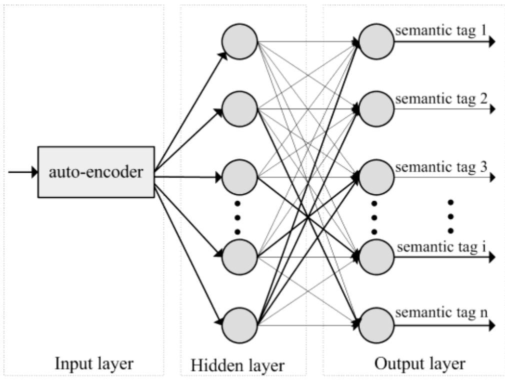

Fig. 4. Architecture of semantic deep learning neural network

In Figure 4, the number of input element of the hidden layer is the same as the number of neurons in the output layer of the automatic encoder. Images with sematic tags are used as sample data to get the sematic tags of the output layer. Then the resulting semantic tag is compared with the true semantic tag, therefore obtaining the total error of the network, as

$$E_{sum} = \sum_{i=1}^{n_s} (y_i - y_{ilabel})^2 \qquad \textbf{(12)}$$

where y is the resulting semantic tag, y_{label} is the true semantic tag, and n_s is the total number of samples.

The work flow of semantic tagging is as Figure 5.

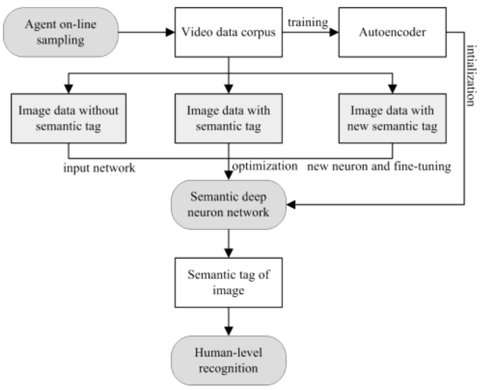

Fig. 5. The construction process of semantic tagging

The video data can be categorized three classes: image data without semantic tag, image data with semantic tag image and image data without new semantic tag. The processing flow is as:

Case I: image data without semantic tag

The image data will be inputted as testing data to semantic deep neural network. The data will be processed and transmitted forward along the deep neural network to the output layer of the network. The semantic label corresponding to the activated neurons of the output layer will be determined as semantic tag of the image.

Case II: image data with semantic tag

The image data will be inputted as training data to fine tune the parameters of the hidden and output layer online until the network is optimized by supervised learning. After that the image data will be inputted again to deep semantic neural network and the original semantic tag will be replaced by the resulting new semantic tag.

Case III: image data without new semantic tag

As new image semantic tag is not included in semantic deep neural network, it is necessary to add a new neuron to corresponding to the new semantic tag. We will re-train the hidden layer and output layer of the semantic deep neural network until getting a stable network by supervised learning.

5. Experiment

Traffic video data can be used for many intelligent control, planning and decision making. However, the huge amount makes it impossible for individuals to analyze them manually. Hence manipulating them automatically with utilizing computers attracted more and more attentions. As most intelligent control systems depend on extracting key information from the video for successive processing, human-level object recognition from traffic video is crucial.

We used the Q-learning based objects recognition algorithm to detect moving objects from video. We can see from the Figure 6 that our recognition approach is able to find out moving objects effectively.

Fig. 6. Objects recognition from video with Q-learning.

We then used deep learning neural network to semantic tagging the recognized objects in 20 different videos. In the test, the semantic tags pedestrian and vehicle are labeled to the recognized objects. The average precision rates are given in Table 1.

Table 1. Average recognition precision of deep learning neural network

Pedestrian tagging precision (%)	Vehicle tagging precision (%)
98.67	99.72

6. Conclusion

As a class of preservation format, video is rich of valuable raw information that can be used for intelligent planning and decision making. Generally, for most intelligent planning and decision making algorithms, recognizing objects from the video and tagging them is a crucial step. However, it is unable to deal with them manually due to huge amount. Deep learning provides us an effective way to understand big data in human-level. Deep learning is about learning to data modeling potential distribution of multi-layer expression of algorithms. Learning high-level features using deep learning architectures has become a big wave of new learning paradigms.

As traffic videos are usually with crowded scenes and low definition, most algorithms are not effective. In this work, we separate image and determine a small rectangle window that contains moving object for each moving object. We propose a Q-learning based moving object recognition algorithm, which is capable of learning directly from raw experience. By taking advantage of Q-learning, we can accurately identify objects from traffic video which is generally of low resolution and with much noise. At the same time, as Q-learning is capable of reducing the computational complex, the identification speed reaches real-time level.

Acknowledgments. This paper is supported by National Natural Science Foundation of China (61272005, 61303108, 61373094, 61472262), Natural Science Foundation of Jiangsu (BK2012616), High School Natural Foundation of Jiangsu (13KJB520020), Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172014K04), Suzhou Industrial application of basic research program part (SYG201422).

References

1. Wu, X., Zhu, X., Wu, G. Q., et al.: Data Mining with Big Data. IEEE Transactions on Knowledge & Data Engineering, Vol. 26, No. 1, 97-107. (2014)
2. Wiering M, van Otterlo M.: Reinforcement Learning: State-of-the-Art (Adaptation, Learning, and Optimization). Springer Berlin Heidelberg. Berlin, German. (2012)
3. Sutton. R., Barto, A.: Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning). The MIT Press, USA. (2006)
4. Watkins, H., Dayan, P.: Q -learning. Machine Learning, Vol. 8, No. 3, 279-292. (1992)
5. Hinton, G. E., Osindero, S.: A Fast Learning Algorithm for Deep Belief Nets. Neural Computation, Vol. 18, No. 7, 1527-1554. (2006)
6. Erhan, D., Courville, A., Bengio, Y., et al.: Why Does Unsupervised Pre-training Help Deep Learning? Journal of Machine Learning Research, Vol. 11, No. 3, 625-660. (2010)
7. Vincent, P., Larochelle, H., Lajoie, I., et al.: Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. Journal of Machine Learning Research, Vol. 11, No. 6, 3371-3408. (2010)
8. Zhou, S., Chen, Q., Wang, X.: Active Deep Learning Method for Semi-Supervised Sentiment Classification. Neurocomputing, Vol. 120, No. 10, 536-546. (2013)
9. Yu, D., Deng, L.: Deep Learning and Its Applications to Signal and Information Processing. IEEE Signal Processing Magazine, Vol. 28, No. 1, 145 – 154. (2011)
10. Tootell, R. H., Hadjikhani, N. K., Vanduffel, W., et al.: Functional Analysis of Primary Visual Cortex (V1) in Humans. Proceedings of the National Academy of Sciences, Vol. 95, No. 3, 811-817. (1998)

11. Engel, S. A., Glover, G. H., Wandell, B. A.: Retinotopic Organization in Human Visual Cortex and the Spatial Precision of Functional MRI. Cerebral Cortex, Vol. 7, No. 2, 181-192. (1997)
12. Gray, H., Standring, S., Borley, N. R.: Gray's Anatomy: the Anatomical Basis of Clinical Practice. Journal of the American Medical Association, Vol. 301, No. 17, 1829-1830. (2009)
13. Van, E., David, C., Maunsell, H. J.: Hierarchical Organization and Functional Streams in the Visual Cortex. Trends in Neurosciences, Vol. 6, No. 1, 370-375. (1983)
14. Miyawaki, Y., Uchida, H., Yamashita, O., et al.: Visual Image Reconstruction from Human Brain Activity Using a Combination of Multiscale Local Image Decoders. Neuron, Vol. 60, No. 5, 915-929. (2008)
15. Grossman, E. D., Randolph B.: Brain Areas Active During Visual Perception of Biological Motion. Neuron, Vol. 35, No. 6, 1167-1175. (2002)
16. Rokszin, A., Márkus, Z., Braunitzer, G., et al.: Visual Pathways Serving Motion Detection in the Mammalian Brain. Sensors, Vol. 10, No. 4, 3218-3242. (2010)
17. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol. 34, No. 7, 1409-1422. (2012)
18. Kearns, M., Singh, S.: Near-optimal Reinforcement Learning in Polynomial Time. Machine Learning, Vol. 49, No. 2, 209-232. (2002)
19. Sutton, R. S., Precup, D., Singh, S.: Between MDPs and Semi- MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. Artificial Intelligence, Vol. 112, No. 1, 181-211. (1999)
20. Connor, D. J., Limb, J. O.: Properties of Frame-Difference Signals Generated by Moving Images. IEEE Transactions on Communications, Vol. 22, No. 10, 1564-1575. (1974)

Fei Zhu is a PhD and is an associate professor of computer science and technology. He is a member of China Computer Federation. His main research interests include reinforcement learning, image processing, and pattern recognition. E-mail: zhufei@suda.edu.cn.

Quan Liu is a member of China Computer Federation. He is a PhD, post-doctor, professor and PhD supervisor. His main research interests include reinforcement learning, intelligence information processing and automated reasoning. He is the corresponding author of this paper. E-mail: quanliu@suda.edu.cn.

Shan Zhong is a PhD candidate. She is a student member of China Computer Federation. Her main research interests include reinforcement learning. E-mail: 20134027005@ stu.suda.edu.cn.

Yang Yang’s interests include bioinformatics, systems biology, and human mutation. E-mail: yang.yang@med.lu.se.

Received: November 14, 2014; Accepted: May 3, 2015.