Object Recognition using Neural Networks with Bottom-up and Top-down Pathways

Yuhua Zheng, *Student Member*, Yan Meng, *Member*, *IEEE*, and Yaochu Jin, *Senior Member, IEEE*

***Abstract*—In this paper, a new artificial neural network model is proposed for visual object recognition, in which the bottom-up, sensory-driven pathway and top-down, expectation-driven pathway are fused in information processing and their corresponding weights are learned based on the fused neuron activities. During the supervised learning process, the target labels are applied to update the bottom-up synaptic weights of the neural network. Meanwhile, the hypotheses generated by the bottom-up pathway produce expectations on sensory inputs through the top-down pathway. The expectations are constrained by the real data from the sensory inputs which can be used to update the top-down synaptic weights accordingly. To further improve the visual object recognition performance, the multi-scale histograms of oriented gradients (MS-HOG) method is proposed to extract local features of visual objects from images. Extensive experiments on different image datasets demonstrate the efficiency and robustness of the proposed neural network model with features extracted using the MS-HOG method on visual object recognition compared with other state-of-the-art methods.**

***Index Terms*—neural networks, bottom-up and top-down pathways, visual object recognition, multi-scale histograms of oriented gradients.**

I. Introduction

Visual object learning and recognition is a challenging problem in computer vision and machine learning areas. Although extensive algorithms have been proposed during past decades, it is still very hard to recognize and learn various objects under different environments with significant variant appearances. Generally speaking, object recognition is to learn invariance features or so-called latent variables of the objects from various training data and to recognize the learned object from unseen data. This procedure usually consists of two steps: object feature extraction and classification. In classification, one main challenge is to correctly represent feature distributions due to significant data variances. Many generative models have been proposed to describe such distributions directly, whereas the parameters of the presumed models are learned through probability-based methods like Bayesian networks [1]. Objects are modeled as flexible constellations of parts and the parameters were learned through an expectation-maximization process in [2]. The approaches based on bag-of-words analogously take image patches as words in texts and learn the patch distributions over the categories based on probabilities [3]. Generative models usually have interpretable meanings and are able to draw samples or synthetic data. However, it is difficult to build optimal generative models with little prior knowledge of the object, especially with a small number of data sets.

On the other hand, many discriminative approaches focused on finding separation boundaries between different categories in object recognition, such as nearest neighbors [4], support vector machines (SVM) [5], multiple classifiers [6], etc.. Discriminative algorithms usually have a better recognition accuracy compared to generative models. However, discriminative algorithms heavily rely on the training data which may lead to the over-fitting problems and poor generalization.

Among these algorithms, artificial neural networks (ANN) have been studied and applied in different ways. For example, a wavelet neural network is applied to recognize object boundary representations with efficient computational cost due to the learning of the optimal scale-translation parameters [7]. A neural network based intelligent machine vision system for cork tiles classification is described in[8], which consists of image acquisition, feature generation and processing. Xin et al. [9] propose a neural network model, where the model is built on individual stable spaces to recognize people faces under uncontrolled conditions. The graph neural network model extends conventional neural network models by representing the data in graph domains to explore their underlying relationships [10]. When applied on object recognition, most ANN models adopt the feed-forward (FF) structures and the supervised learning with error back-propagation from data space to latent space.

However, evidence found in cognitive brain research and neuroscience suggests that the nervous system responsible for object recognition has distributed cortical structures containing both bottom-up and top-down pathways [11, 12]. Grossberg started to explore this area since 1970's and proposed the Adaptive Resonance Theory (ART), which is a general framework for representing interactions between bottom-up and top-down pathways [13]. The Hopfield network was studied as associative memories with symmetric connections

Yuhua Zheng and Yan Meng are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA. (Email: yzheng1@stevens.edu, yan.meng@ stevens.edu).

Yaochu Jin is with the Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, UK. (Email: yaochu.jin@surrey.ac.uk).

.

between neurons back to 1980s[14]. The biased competition theory has been proposed in [15] to explain the top-down attention of spatial stimulus and different feature dimensions. A neural cluster model is proposed in [16] to develop spatiotemporal features by adapting both Hebbian rules and lateral inhibition from natural videos. A neural network of Wilson-Cowan oscillators is proposed in [17] for recognition of abstract object by investigating the interactions among topological maps, auto-associative memory and gamma-bank synchronization

In this paper, we aim to explore the potentials of combining discriminative and generative data flows in ANN architectures. More specifically, we propose a novel ANN model, called FBTP-NN, by fusing the information from both bottom-up (stimulus) and top-down (expectation) pathways, and apply this model to object recognition. A learning algorithm for the proposed FBTP-NN model is also suggested which focuses on the following two procedures iteratively: the impact of top-down expectations on the modulations of neuron activity in the lower-layer of the ANN, and the consequential updates of neuron activities in the higher-layer of the ANN through the bottom-up propagation.

Instead of using the predefined spatial attention to distribute attention strengths to different regions in the images, like most bottom-up and top-down approaches in neural networks [15, 18], the top-down expectation in the proposed model is generated from training samples, and focuses on interpreting the object appearances to recognize objects instead of searching and localizing them. In other words, our FBTP-NN model focuses on solving "what is the object?" problem instead of "where is the object?" problem. We believe that the best interpretation of an object should contain not only the input data but also a priori knowledge of the object that has been learned before, which can be realized through the top-down expectations. Compared to other classifiers, such as SVM, where a specific SVM classifier has to be constructed for each class, the proposed FBTP-NN model is a single model which can be applied to multi-class recognition directly.

Some preliminary work has been presented in our previous work [19]. Several major extensions are reported in this paper. (1) A probability-based framework is introduced to describe the iterative fusion process with the constraint of minimizing the joint distribution of the synaptic weights of both bottom-up and top-down pathways. Then a cost function containing both pathways is constructed and the corresponding learning rules are conducted. (2) To improve the overall object recognition performance, a new feature extraction method, called the multi-scale histograms of oriented gradients (MS-HOG) method, has been proposed. (3) Several new experiments on large datasets such as the MIT pedestrian dataset and Caltech objects datasets have been adopted to evaluate the efficiency of the proposed model.

The rest of this paper is organized as follows. Related work is discussed in Section II. The proposed FBTP-NN model is presented in Section III. The learning process of the FBTP-NN model is described in Section IV. Section V presents a experiment without feature extraction to demonstrate the integration process of two pathways of the proposed model. Section VI describes experimental results with an advanced feature extraction method (i.e., MS-HOG) on different object recognition databases. Conclusions and future work are given in Section VII.

II. Related Work

Some work has been proposed to combine generative and discriminative approaches in a two-stage way: using generative algorithms to extract features and using discriminative models to learn features. The 'discriminatively-trained' generative model is proposed in [20] to blend both discriminative and generative priors via a specific parameter, which can be treated as a general way to interpolate discriminative and generative extremes. A boosting algorithm is applied in [21] to select features considering both discrimination and reconstruction to achieve better robustness.

Some neural-network-like approaches tried to integrate bottom-up and top-down information for object recognition and interpretation. The auto encoder/decoder algorithms [22] and well-known Restricted Boltzmann Machine[23] focus on learning generative models from unlabeled data. A feedback model is proposed in [24] to bias the perceptual stimuli and facilitate the learning of sub-ordinate level representations suitable for object identification and perceptual expertise. Salinas and Sejnowski [25] propose a gain modulation theory to explain how the modulating neurons affect the gain or sensitivity of others as a widespread mechanism. A neural network model is proposed in [26] to restore partially-occluded patterns using feedback signals. The Helmholtz machine [27] contains one generative network and one discriminative network independently, and a sleep-wake learning algorithm is applied to search for the latent variables from the data in an unsupervised way. However, these approaches either learn the top-down pathway or bottom-up pathway separately without fusing the neural dynamics of both pathways, or tangle both pathways together by applying symmetric weights. Some methods mainly focus on building up biologically plausible models, where only very simple images have been considered in experiments.

In this paper, we aim to propose a new ANN model, where the information in the bottom-up and top-down pathways is fused in a natural way under the joint probability distribution of synaptic weights of both pathways. Constrained by the true labels and sensory data, the bottom-up stimuli and the top-down modulation propagate iteratively to change neuron activities and adjust related synaptic weights of the neural networks.

III. Fusing Bottom-up and Top-down Pathways in Neural Networks (FBTP-NN)

A. The System Framework

Although the working mechanisms of human cortex have not been fully understood in neuroscience and cognitive science, increasing evidence has revealed that the neural system associated with learning and object recognition is a distributed cortical structure containing both bottom-up and top-down pathways. When an object is presented, the sensory input may generate ambiguous hypotheses, which could get similar scores (from neuron activities) in the conventional feed forward neural networks (FF-NN). However, the top-down signals that contain a priori knowledge or the memory of the related objects can help to modulate the bottom-up pathway so that the ambiguousness in the stimulus can be reduced and more confident hypothesis can be generated and selected.

In supervised learning of FF-NN, usually the objective function is to minimize the error between the predicted labels and the real ones. In training the FBTP-NN model, both sensory input data and output labels are treated equally as the environmental constraints. And the FBTP-NN model tries to learn both hypotheses from bottom-up pathway and expectations from top-down pathway at the same time, which can be achieved by updating the network weights based on the fusions of the bi-directional data flows.

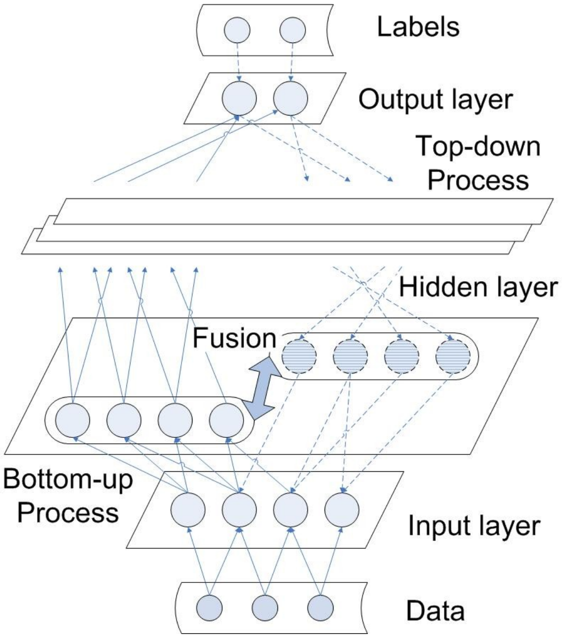


Fig. 1. The framework of the FBTP-NN model. The bottom-up process is represented in solid lines and the top-down process is described in dashed lines. Data and labels put constrains on the input and output layers of the network. At every hidden layer, the bottom-up stimuli (in solid circles) are fused with top-down expectations (in dashed circles) and vice versa.

The general framework of the FBTP-NN model is developed based on the above ideas and is shown in Fig.1. The model may have multiple layers but contains only one input layer and one output layer, which are the interfaces of the network to the environment (i.e., input data and output labels). A number of hidden layers exist in between. The input layer receives the sensory input and generates a few hypotheses at the output layer through the bottom-up pathway, the output layer then produces expectations on the sensory stimulus via the top-down pathway, and this information processing procedure is conducted layer by layer. For example, the expectation information will be fused with the sensory stimuli to update neuron activities of the hidden layers. The updated neuron activities will then generate new hypotheses and send them to the output layer accordingly. Such procedures repeat until certain stop conditions are met. During the learning, the fusion of the neuron activities in both pathways is conducted only at hidden layers.

To train the FBTP-NN model, it is essential to define a cost function that considers requirements of both pathways. To this end, a cost function that considers both the labeling error at the output layer and the discrepancy at the input layer has been developed. The weights in both pathways of the neural network are updated iteratively by minimizing this cost function. Details about the cost function and the fusion technique will be discussed in later sections.

B. The Basic Two-layer FBTP-NN Sub-network

The proposed FBTP-NN may contain multiple hidden layers. Since the update of neuron activities and synaptic weights depend only on its adjacent layers, for the sake of simplicity, we will first discuss the basic two-layer FBTP-NN sub-network structure, as shown in Fig. 2. In this two-layer structure, the bottom layer $\{x_1, x_2, ..x_N\}$ is called the data layer with N neurons. The top layer $\{y_1, y_2, ..y_M\}$ is called the feature layer with M neurons, which is considered as the features of the data layer. Each layer contains a number of neurons. Neurons of different layers are fully connected. For example, w_{iu} is the synaptic weight of the bottom-up pathway from neuron x_i to y_u; and q_{ui} is the synaptic weights of the top-down pathways from neuron y_u to x_i. Note that the assumption of a fully-connected structure may not be plausible in real visual cortical systems, where neurons of different layers are connected sparsely according to the receptive fields with various sizes. Since we mainly focus on exploring vertical data flows here, it is assumed that the network has fully-connected inter-layer connections.

The sub-network (inside the dotted square) is stimulated by the environment, which may contain both data vector $\mathbf{D} = \{d_1, d_2, ..d_N\}$ and the feature vector $\mathbf{L} = \{l_1, l_2, ..l_M\}$.

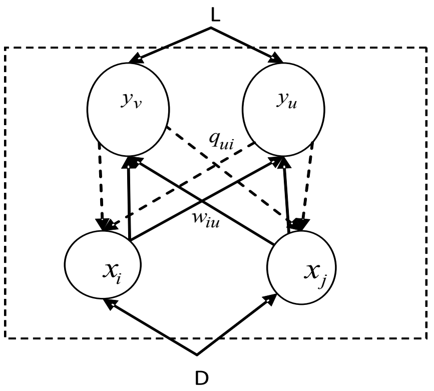


Fig. 2. The basic two-layer FBTP-NN sub-network.

In such a sub-network, the neuron activity of a neuron x on the data layer is defined as:

$$x_i(t+1)=\omega_1\cdot x_i(t)+(1-\omega_1)\cdot g\left(\sum_{u=1}^{M}q_{ui}\cdot y_u(t+1)\right) \tag{1}$$

where $x_i(t+1)$ is the neuron activity of i-th data neuron at time step t+1, which depends on its neuron activity from the last time step $x_i(t)$ and the stimuli from the current time step $g\left(\sum_{u=1}^{M}q_{ui}\cdot y_u(t+1)\right)$. $\sum_{u=1}^{M}q_{ui}\cdot y_u(t+1)$ actually represents the sum of top-down expectations from all its related feature neurons y_u and the corresponding top-down weights q_{ui}. The expectation is then fed into the activation function g, which is a sigmoid function defined as $g(x)=1/(1+e^x)$ to represent the activation characteristic of neurons. $\omega_1\in(0,1)$ is a factor.

Similarly the neuron activity of a feature neuron y is defined as:

$$y_u(t+1)=\omega_2\cdot y_u(t)+(1-\omega_2)\cdot g\left(\sum_{i=1}^{N}w_{iu}\cdot x_i(t+1)\right) \tag{2}$$

where $y_u(t+1)$ is the neuron activity of u-th feature neuron at time step t+1, which depends on its previous neuron activity $y_u(t)$ and the current stimuli of $g\left(\sum_{i=1}^{N}w_{iu}\cdot x_i(t+1)\right)$ from all its connected data neurons $x_i(t+1)$ with the corresponding bottom-up weights w_{iu}. $\omega 2\in(0,1)$ is a factor.

Now we will provide a simple convergence proof of the neuron dynamics in Eqns. (1) and (2). We will use Eqn. (2) as an example here. Eqn. (2) can also be written as:

$$y_u(t+1)=\omega_2\cdot y_u(t)+\Delta y_u(t+1) \tag{3}$$

where $\Delta y_u(t+1)=(1-\omega_2)\cdot g\left(\sum_{i=1}^{N}w_{iu}\cdot x_i(t+1)\right)$. We can further expend Eqn. (3) to the following format:

$$\begin{aligned}y_u(t+1)&=\omega_2\cdot y_u(t)+\Delta y_u(t+1)\\&=\omega_2(\omega_2\cdot y_u(t-1)+\Delta y_u(t))+\Delta y_u(t+1)\\&=\omega_2^2 y_u(t-1)+\omega_2^1 y_u(t)+\omega_2^0\Delta y_u(t+1)\\&=\sum_{k=0}^{t+1}\omega_2^{t+1-k}\cdot\Delta y_u(k)\end{aligned} \tag{4}$$

Where the new stimuli of moment k is $\Delta y_u(k)=(1-\omega_2)\cdot g\left(\sum_{i=1}^{N}w_{iu}\cdot x_i(k)\right)$. Since the sigmoid function is within the range of (0,1), we have $0\le\Delta y_u(k)\le(1-\omega_2)$. Suppose that from time step k', the bottom-up weights w_{iu} are fixed, if the neuron activities of the input layer $x_i(t)=x_i(k'),\ \text{where}\ t\ge k'$ are also fixed, the stimuli of the feature layer can be written as $\Delta y_u(t)=\Delta y_u(k')=(1-\omega_2)\cdot C,\ \ \text{where}\ t\ge k'$ and the constant C is the result of the activation function. Therefore, from Eqn. (4), we have

$$\begin{aligned}y_u(t+1)&=\sum_{k=0}^{k'}\omega_2^{t+1-k}\cdot\Delta y_u(k)+\sum_{k=k'}^{t+1}\omega_2^{t+1-k}\cdot(1-\omega_2)\cdot C\\&\cong\sum_{k=k'}^{t+1}\omega_2^{t+1-k}\cdot(1-\omega_2)\cdot C\\\Rightarrow&\lim_{t\to\infty}y_u(t+1)=C\end{aligned} \tag{5}$$

Eqn. (5) shows that given the fixed bottom-up weights and neuron activities of the lower layer, the activities of feature neurons can be approximated by the sum of a geometric series which will converge over time. Eqn. (5) means that the neuron activity will converge if the related weights are learned. Similar proof can be applied to Eqn. (1).

Ideally the converged neuron activity will be applied to the cost function (defined in later section) and be used to update the related connection weights. Practically, the neuron activities after several iterations are adopted to learn the synaptic weights of the network. Meanwhile, the input sensory data, the true labels and the neuron activities are regulated inside the range of (0, 1) to narrow down the convergence range of the neuron dynamics as well.

A multi-layer neural network can be constructed by assembling a number of basic two-layer sub-networks. More specifically, the first basic sub-network consists of the input layer (as the data layer) and the first hidden layer (as the feature layer). Then, the second basic sub-network includes the first hidden layer (as the data layer) and the second hidden layer (as the feature layer). This procedure continues all the way up to the output layer, which is the feature layer of the last basic sub-network. For a multi-layer neural network, the input layer is the sensory input, and the output layer is the corresponding labels. Any hidden layer can be either treated as the data layer or the feature layer depending on which basic sub-network it is referring to at the current moment.

IV. The Leaning Algorithms

A. The Learning Objective

All given training data **D** and the corresponding labels **L** can be treated as the samples of data-label variables $\{D, L\}$. On the other hand, all possible values of weights of the FBTP-NN model construct the weight variables $\{W, Q\}$, where W and Q refer to the synaptic weights of bottom-up and top-down pathways, respectively. The discriminative representation of the joint distribution of all above variables can be defined as:

$$P(D,L,W,Q) = P(L|D,W) \cdot P(D|Q) \cdot P(W,Q) \tag{6}$$

where $P(L|D,W)$ is the discriminative likelihood along the bottom-up pathway, representing the probability of true labels L given the data D and the current synaptic weights W of the bottom-up pathway. $P(W,Q)$ is the joint distribution of network weights. $P(D|Q)$ is the marginal data prior given the top-down generative weights. Therefore, if the network is considered to be parameterized by the weights of both pathways, the learning goal is to find proper $\{W,Q\}$ that can maximize Eqn. (6).

On the other hand, the generative representation of the joint distribution of Eqn. (6) can be defined as:

$$P(D,L,W,Q) = P(D|L,Q) \cdot P(L|W) \cdot P(W,Q) \tag{7}$$

where $P(D|L,Q)$ is the data expectation along the top-down generative pathway of the network, as the conditional probability of data D given the label vector L and the synaptic weights Q of the top-down pathway. $P(L|W)$ is the marginal label prior given the bottom-up weights.

Since the prior probability $P(W,Q)$ can be assumed to be a uniform distribution, we can derive Eqns. (6) and (7) to the following two equations, respectively:

$$P(D,L,W,Q) \propto P(L|D,W) \cdot P(D|Q) \tag{8}$$

$$P(D,L,W,Q) \propto P(D|L,Q) \cdot P(L|W) \tag{9}$$

where $\propto$ denotes proportionality.

So the learning objective is to find the corresponding weights $\{W,Q\}$ that can maximize the joint distribution $P(D,L,W,Q)$, which can be defined as:

$$\{W,Q\} = \underset{(W,Q)}{\arg\max}\{P(D,L,W,W)\}$$

It is difficult to learn both W and Q simultaneously when these two variables are tangled together. However, by using Eqns. (8) and (9), one of them can be learned with the other one being fixed. In Eqn. (8), the bottom-up weights W can be learned by maximizing the discriminative likelihood $P(L|D,W)$, with the data prior $P(D|Q)$ given fixed top-down weights Q. Similarly, in Eqn. (9), the top-down weights Q can be learned by maximizing the data expectation $P(D|L,Q)$ with the label prior $P(L|W)$ given the fixed bottom-up weights W.

To achieve this learning objective, a cost function can be constructed to maximize the discriminative likelihood $P(L|D,W)$ and the data expectation $P(D|L,Q)$ of Eqns. (8) and (9) with the data prior $P(D|Q)$ and the label prior $P(L|W)$, which can be treated as the desired neural activities.

Maximizing a likelihood probability can be achieved by minimizing a cost function that represents the errors between the ground truth and the outputs of the model with given parameters over training data[28]. In this manner, the maximization of the discriminative likelihood can be obtained by minimizing the errors between the network bottom-up outputs and the desired labels, whilst the maximization of the data expectation can be approached by minimizing the errors between the network top-down expectation and the desired data over all neurons. Therefore, the cost function of the learning algorithm can be defined as:

$$E(t) = \sum_{u=1}^{M}(l_u(t) - y_u(t))^2 + \sum_{i=1}^{N}(d_i(t) - x_i(t))^2 \tag{10}$$

where $l_u(t)$ is the desired neuron activity of latent neuron u and $y_u(t)$ is the current activity for the same neuron. Similarly, $d_i(t)$ is the desired neuron activity of data neuron i and $x_i(t)$ is the current activity for the same neuron.

The first part of Eqn. (10) actually corresponds to the maximization of the discriminative likelihood $P(L|D,W)$ and the second part corresponds to the maximization of the data expectation $P(D|L,Q)$. Since $l_u(t)$ denotes the desired neural activities of latent neuron, it can be used to represent the data prior $P(D|Q)$. Similar, $d_i(t)$ can be used to represent the label prior $P(L|W)$. We will define $l_u(t)$ and $d_i(t)$ in next section.

B. Fusing Bi-directional Pathways

For neurons of any hidden layer, their neuron activities depend on both the bottom-up pathway (when it works as the latent neuron) and the top-down pathway (when it acts as the data neuron). However, the desired neuron activities should be fixed if the joint distribution $P(D,L,W,Q)$ is settled. This means that the neuron activity along the bottom-up pathway and the neuron activity along the top-down pathway for the same neuron should be the same for a settled joint distribution.

In Eqn. (8), given weights Q, the data prior $P(D|Q)$ actually represents the desired neuron activities along the top-down pathway, which should equal to that along the bottom-up pathway. This means that the desired neuron activity along the top-down pathway can be used to update the corresponding bottom-up weights. Therefore, the desired neuron activity $l_u(t)$ of a latent neuron can be defined as:

$$l_u(t) = \alpha y_u(t) + (1-\alpha)x_u(t-1) \tag{11}$$

where $y_u(t)$ is the current neuron activity of neuron u along the bottom-up pathway when the neuron is treated as a feature neuron. $x_u(t-1)$ is the desired neuron activity of the same neuron along the top-down pathway when the neuron is treated as a data neuron. Since the updates of the neuron activities along two pathways are unsynchronized, the desired neuron activity along the top-down pathway is from the last time step *t-1*. α is the fusion rate under the discriminative representation. Eqn. (11) actually shows that the bottom-up propagation tries to match the top-down expectation by pushing $y_u(t)$ towards $x_u(t-1)$. Therefore, the bottom-up weights *W* can be learned by maximizing the discriminative likelihood $P(L|D,W)$, which is the first part of Eqn. (10), where the desired *L* is given by data prior $P(D|Q)$ via Eqn. (11).

Similarly based on Eqn. (9) the desired neuron activity of a neuron under the generative representation can be defined as:

$$d_i(t) = \beta x_i(t) + (1-\beta)y_i(t-1) \tag{12}$$

where $x_i(t)$ is the current neuron activity of the data neuron *i* and $y_i(t-1)$ is the desired neuron activity of the same neuron along the bottom-up pathway. β is the fusion rate under the generative representation. Eqn. (12) shows that the top-down expectation also tries to match the bottom-up sensory stimuli. If we treat the neuron activities along the top-down pathway as the learned template, Eqn. (12) aims to minimize the average difference between the template and various sensory stimuli. Therefore, the top-down weights *Q* can be learned by maximizing the data expectation $P(D|L,Q)$, which is the second part of Eqn. (10), where the desired *D* is defined by label prior $P(L|W)$ via Eqn. (12).

In summary, Eqns. (10) (11) and (12) actually define an iterative fusion process for learning both bottom-up and top-down weights, in an unsynchronized manner. Since both pathways follow the same joint distribution, as shown in Eqns. (8) and (9), their desired neuron activities can be applied to update the corresponding weights for each other.

C. Weights Updates

Given Eqns. (10), (11) and (12), the gradient descent is applied to update the weights of both pathways so that the cost function defined in Eqn. (10) can be minimized. Although Eqn. (10) contains both label errors of discriminative likelihood and differences of data expectation, the synaptic weights of two pathways can be updated independently based on following derivations. For the purpose of simplicity, the time dependency will be omitted thereafter.

The derivative of the cost function with respect to the bottom-up weight w_{iu} can be obtained as follows:

$$\frac{dE}{dw_{iu}} = -2(d_i - x_i)\frac{dx_i}{dw_{iu}} - 2(l_u - y_u)\frac{dy_u}{dw_{iu}} \tag{13}$$

Substituting Eqns. (1) and (2) into Eqn. (13), we have:

$$\frac{dE}{dw_{iu}} \propto g' \cdot x_i \cdot (l_u - y_u) \tag{14}$$

where g' is the derivative of the activation function. For the sigmoid function $g(x) = 1/(1+e^x)$, $g'(x) = g(x)(1-g(x))$ is a constant for a given input. x_i represents the activity of the related data neuron. l_u and y_u are the desired neuron activity and the real neuron activity of the latent neuron, respectively. Therefore, to minimize the cost function *E*, the change of weight Δw_{iu} is defined as:

$$\Delta w_{iu} = r_1 \cdot x_i \cdot (l_u - y_u)\,, \tag{15}$$

where r_1 is the learning rate of the bottom-up weights. Eqn. (15) is a Hebbian-like error-driven learning method.

Similarly, we can get the derivative of the cost function with respect to the top-down weights *Q*, and the update rule for a specific q_{ui} can be derived as:

$$\frac{dE}{dq_{iu}} \propto g' \cdot y_u \cdot (d_i - x_i) \tag{16}$$

$$\Delta q_{ui} = r_2 \cdot y_u \cdot (d_i - x_i) \tag{17}$$

where r_2 is the learning rate of the top-down weights.

D. The Learning Algorithm for the FBTP-NN Model

By defining the desired neuron activities for all the layers, the supervised learning can be conducted through a number of bottom-up, top-down, and fusion iterations. Driven by the input data, the network generates hypotheses layer by layer through the bottom-up pathway. Then expectations are generated based on these hypotheses along the top-down pathway. For hidden layers, the stimulus and expectation are fused to generate the desired neuron activities. The supervised learning procedure of the FBTP-NN can be summarized as followings.

Generally, a FBTP-NN is a multi-layer neuron network consisting of a number of basic two-layer models described in Section III.A. Given a FBTP-NN model with randomly initialized weights $\{W,Q\}$ and a number of data-label pairs (D,L), the bottom-up process starts from the input layer *X* with input data *D*. Here, the input layer of the network is the data layer and the first hidden layer is the feature layer in the basic two-layer model.

- Step 1. Calculate the neuron activities on the feature layer *Y* via Eqn. (2).
- Step 2. Update the bottom-up weights *W* via Eqn. (15). *L* is the feature information for the current feature layer, which can be defined in two cases. If the current feature layer is the output layer of the neural network, *L* is the true label. If the current feature layer is a hidden layer, *L* is defined in Eqn. (11).
- Step 3. Move up one layer to build a new basic model.

The current feature layer becomes the data layer and the adjacent top layer becomes the feature layer in the new basic structure. Repeat steps 1-2 for the learning of the new basic structure.

- Step 4. Repeat steps 1-3 until the output layer of the whole neural network.

Now we perform the top-down and fusion process from the output layer. Here, we start with a basic model using the output layer as the feature layer and the last hidden layer as the data layer.

- Step 5. Calculate the neuron activities in data layer *X* via Eqn. (1).
- Step6. Update the top-down weights *Q* using Eqn. (17). Here, the data information *D* can be defined in two cases. If the current data layer is the input layer, *D* is the true sensory data. If the current data layer is a hidden layer, *D* is defined in Eqn. (12).
- Step 7. Move down one layer to build a new basic model. The current data layer becomes the feature layer and the adjacent bottom layer becomes the data layer in the new basic structure. Repeat steps 5 and 6 for the learning of this new basic structure.
- Step 9. Repeat steps 5-8 until the input layer of the whole neural network.
- Step 10. Repeat steps 1 to 9 until the stop condition is met.

By repeating the above steps, the network will learn the labels in the output layer as well as the corresponding stimulus in the input layer by updating synaptic weights in both pathways.

E. The Testing Process of FBTP-NN Model

Once the FBTP-NN model has been trained, it has both discriminative and generative abilities. When unseen data are presented, object recognition can be achieved by running the bottom-up discriminative process only. The output neuron with the highest activation value is considered to be the recognized object class. However, if more than one output neuron has similar activation values, the top-down process will be activated to help the selection of the object class. By firing a single output neuron and keeping others silent, the top-down process can generate the expectation of the corresponding class at the input layer, which can be compared with the current sensory data to estimate the difference. Combining the discriminative confidence on the output layer and the generative difference on the input layer, the overall decision for object classification can be made.

V. Experimental Results without Feature Extraction

A. Experiment Settings

To demonstrate the learning process of the proposed FBTP-NN algorithm, a three-class classification experiment on visual object recognition has been performed. The data of bicycle, revolver, and treadmill are taken from Caltech 256, as shown in Fig. 3. The original images are transformed into gray images, where objects are presented as white pixels and the background as black pixels. For each category, objects with different appearances, sizes, orientations, backgrounds and lightening conditions are selected. For simplicity, no advanced feature extraction is conducted and only the pixel values of images are applied as inputs for the FBTP-NN model. The recognition with more advanced features as inputs may help to improve the recognition performance and will be discussed in the later sections.

Then a three-layer FBTP-NN is built for object recognition. . The number of neurons in the input layer equals to the size of the training images, i.e. 32x24=768. The hidden layer has 0.5x768=384 neurons and the output layer has 3 neurons. Neurons of adjacent layers are fully connected.

Table I provides the parameter settings used in the experiments. They are chosen empirically by trial and error. The learning rates are set up as the same value for different weights. A fusion ratio of 0.01 produces good recognition performance for both pathways. For the supervised learning, generally 1000 iterations are adequate to achieve satisfactory results. The trade-off rate ω_1 and ω_2 of neuron activities are both set as 0.8 for all experiments. Sensitivity analysis of these parameters on the recognition performance will be discussed in next section.

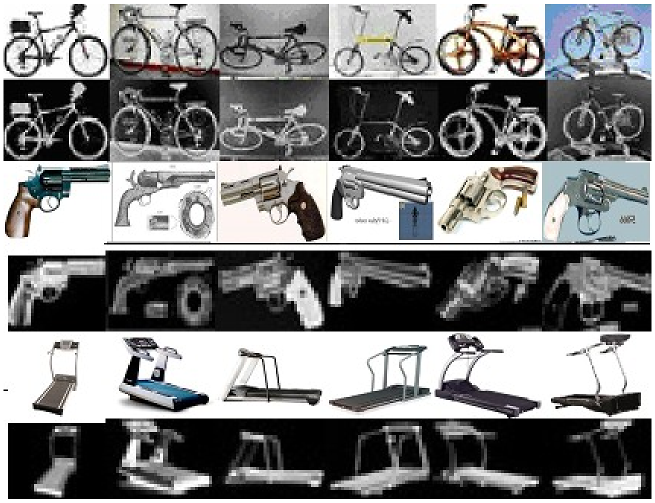


Fig. 3. Experimental data taken from Caltech 256.

TABLE I. Parameter setups

Coefficient	Value
Learning rate r_1 r_2	0.05
Fusion ratio α β	0.01
Max training loop	1000

The FBTP-NN model can be trained by using either the online learning mode or the batch learning mode. For the online learning mode, the training data are mixed randomly and presented to the network sequentially to reduce the forgetting influence.

B. Learning and Recognition Performance

Fig. 4 shows the evolvements of top-down expectation on input layer. Since we applied raw pixel values of images as network inputs, the expectation naturally looks like original images. At the beginning, the network has learned nothing and the expectation is just noise, as shown on the top part of Fig.4. With more learning samples shown on the left of Fig. 4, the network can better capture the features of the object. The generated expectation has been evolved from some pure noise (top in Fig. 4) to a much more clear expectation prototype (bottom in Fig. 4).

For this 3-class recognition problem, each class has 50 samples with various sizes, appearances and orientations. 50% data of each class are applied for training and the rest for testing. Fig.5 and 6 show the two types of error changes of the cost function defined in Eqn. (10) over training loops, respectively. Fig.5 shows the average label errors between the true labels and the outputs of the neural network over all three classes with all the training data. Fig. 6 shows the average data discrepancy between the sensory data and the neuron activities of the input layer of the network over all three classes. From Fig. 5 we can see that the label error will converge to zero over time, which will ensure the recognition convergence. From Fig. 6, the data discrepancy converges to a stable value (which is not zero) over time, this is reasonable because it is impossible to obtain the expectation template which is the same with all various input data. Fig. 5 and 6 show that by applying the fusion process described in Eqns. (11) and (12), the cost function (10) can be minimized monotonically over learning steps.

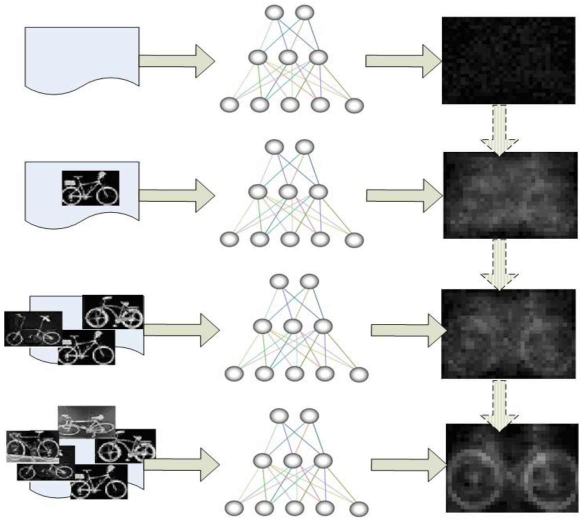


Fig. 4. Illustrative example of evolvements of the top-down expectation. With more samples (on the left), the network is able to generate better expectation of the object (on the right) from top to bottom.

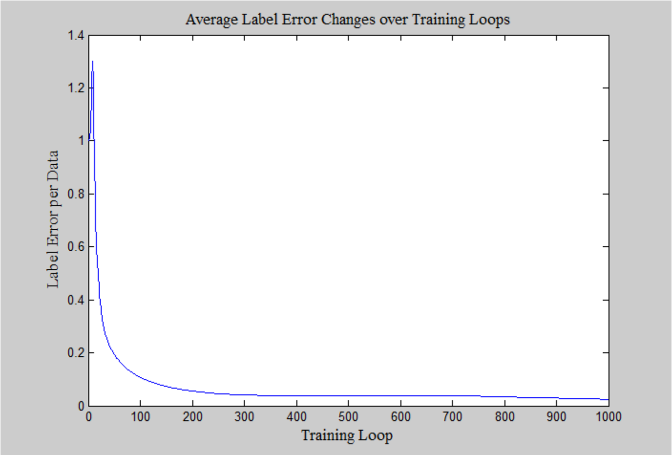


Fig. 5.The average label error changes over training loops for the 3-class experiment.

Table II lists the average recognition rate over three classes with different learning rates. With a fixed training loop of 1000, a learning rate of 0.05 seems to result in the best performance. Table III lists the sensitivity analysis of the fusion rate on the recognition performance with the training loop as 1000 and learning rate as 0.05. It can be seen from Table III that a fusion rate of 0.01 has the best recognition performance. A too big fusion rate (e.g. 0.1 and 0.5) may cause oscillation in neural dynamics.

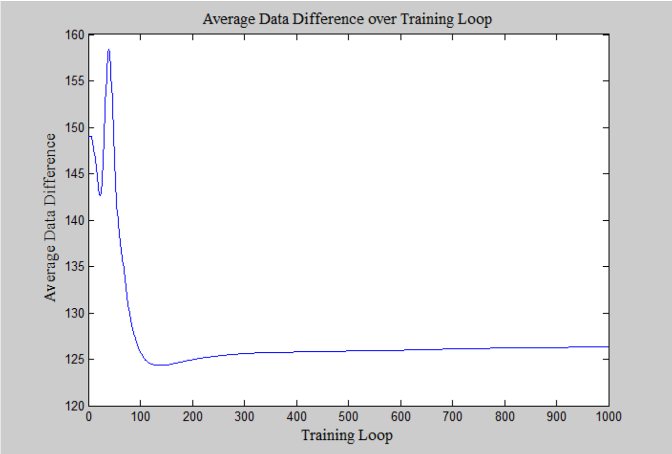


Fig. 6. The average data discrepancy changes over training loops for the 3-class experiment.

Similarly, the sensitivity analysis of the training loops on the recognition performance is conducted, as listed in Table IV. From Table IV, it can be seen that once the network has learned the object features after a certain training loop, increasing training loop will not improve the recognition performance significantly.

TABLE II. RECOGNITION RATES OVER DIFFERENT LEARNING RATES FOR THE THREE-CLASS EXPERIMENTS

Learning Rate	0.01	0.05	0.1	0.5
Average Recognition Rate	89.74%	92.15%	83.33%	71.79%

TABLE III. RECOGNITION RATES OVER DIFFERENT FUSION RATES FOR THE THREE-CLASS EXPERIMENTS

Fusion Rate	0.01	0.1	0.5
Average Recognition Rate	92.31%	86.15%	79.48%

TABLE IV. RECOGNITION RATES OVER DIFFERENT TRAINING LOOPS FOR THE THREE-CLASS EXPERIMENTS

Training Loop	500	800	1000	1500
Average Recognition Rate	84.62%	91.75%	92.20%	92.15%

In summary, to achieve a better recognition performance, the learning rate, fusion rate, and training loop needs to be set up by trial-and-error for a specific application.

Then we compare the proposed algorithm with the FF-NN using the back-propagation learning. FF-NN is implemented using the Netlab toolbox [29]. The data are divided into two sets: the training set and the testing set, according to different training ratios. In this experiment, the training ratio is 60%, which means that 60% of the data are used for training and the rest 40% are used for testing. The data will be trained and tested for five times and the average recognition rate will be calculated to evaluate the performance of the algorithm. Table V shows the comparison results of the recognition rates using both FF-NN and FBTP-NN. It can be seen that FBTP-NN outperforms the FF-NN on all object classes.

TABLE V. RECOGNITION RATES OVER DIFFERENT TRAINING DATA SIZES FOR THE THREE-CLASS EXPERIMENTS

Training Ratio	Bicycle		Revolver		Treadmill	
Network	FBTP	FF	FBTP	FF	FBTP	FF
60%	100	96.7	100	95.2	98.2	97.8

Due to the top-down process, the FBTP-NN model is more robust to noisy and incomplete data compared with feed-forward networks. As shown in Fig.7, some incomplete samples are generated by randomly removing part of the data from images. 50% missing data means 50% of pixels in the images are picked out randomly and set to zero. Therefore, pixels missing from of the objects to be recognized are usually less than 50%. Then both FBTP-NN and FF-NN are trained by samples without missing data, and are tested using the samples with incomplete information. In this experiment, 40% samples are used as training data and all incomplete data are used for testing, i.e. 50 testing data for each class and the experimental results are listed in Table VI. From Table VI, it can be seen that FBTP-NN can achieve much better recognition performance than FF-NN.

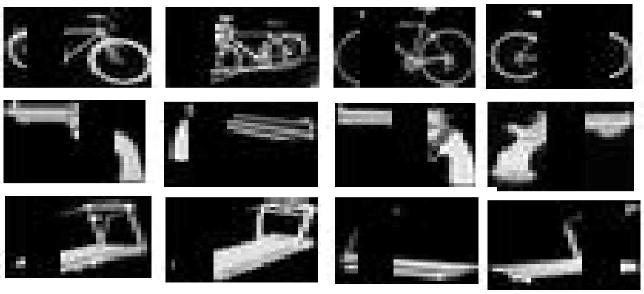


Fig. 7. Some incomplete samples with part of the pixels missing.

TABLE VI. RECOGNITION RATE COMPARISON ON INCOMPLETE DATA FOR THREE-CLASS EXPERIMENT

Testing Data	50% Missing	
Network	FBTP	FF
Bicycle	92%	90%
Revolver	78%	70%
Treadmill	84%	78%

The above results can be explained as follows. First, the information fusion process is applied in the FBTP-NN can push the bottom-up stimuli towards the top-down expectation. Therefore, the learned prototypes of the objects from sensory data to output labels become more compact since the average of all the previous samples of the same class is used. When incomplete input data is presented, the FBTP-NN is able to generate better classification results with the help of this top-down expectation. The second reason comes from the adoption of the top-down pathway during the testing process. When more than one hypothesis is activated from bottom-up, these hypotheses will be verified via the top-down pathway sequentially. A better classification decision can be made by combining the bottom-up output score and top-down expectation score. During the experiments, up to 13% data are verified by the top-down pathway.

VI. EXPERIMENTAL RESULTS WITH AN ADVANCED FEATURE EXTRACTION METHOD

A. Local Features for Object Recognition

Efficient feature extraction is also critical to improve the overall recognition performance. Recently, the gradient-based local features have demonstrated to be effective for object recognition. The scale-invariant feature transform (SIFT) method was proposed to search for key points and construct descriptors of corresponding regions around key points [30]. Histograms of Oriented Gradients (HOG) used the gradient orientations to describe local object appearance [31], where the whole image was represented by combining such orientation histograms across a number of small regions, called cells.

The above approaches have several limitations when applied to visual object recognition. SIFT-based algorithms focus on object matching between different images. After a few key points are found, high-dimensional descriptors are needed to represent local appearances. HOG-based methods partition the whole images into small cells with overlapping, and each cell holds a histogram, which leads to a large number of histograms. However, for general object recognition, we would like to have features with a lower dimension to reduce computational cost.

Therefore, in this paper, a new feature extraction algorithm is proposed, which is called multi-scale HOG (MS-HOG). First, the image is partitioned into a fixed number of cells and a histogram is constructed for each cell. Inside each cell, an edge-map and a contour-map are applied to filter out trivial gradients. Then, to enhance the robustness, three scales of the original image are built and the HOG on each scale is extracted individually. The setting of three scales has demonstrated a trade-off between the representation ability and the computational cost. By applying this multi-scale strategy, the information loss due to the reduction of cell populations can be alleviated. On the other hand, the coarse scale can provide some global description of the objects, which is an advantage for object recognition compared with those approaches using local features only. In the following section, we will discuss the details about the MS-HOG features.

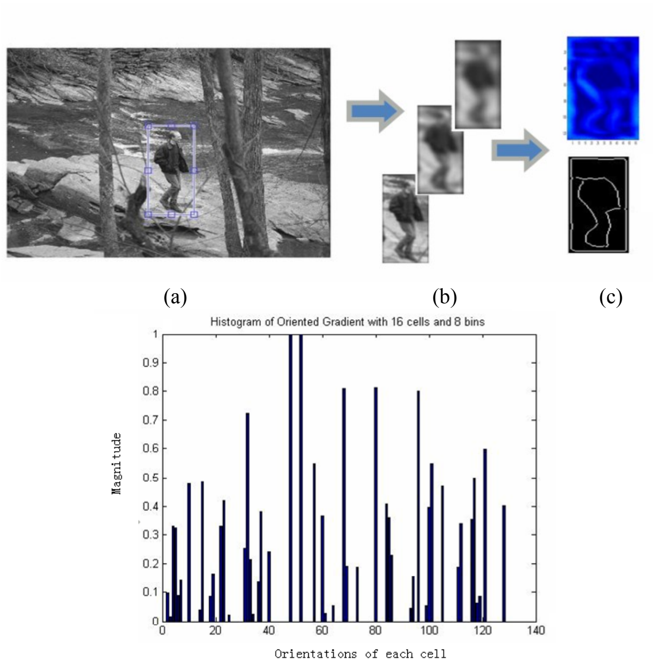
B. Extracting MS-HOG Features

Firstly, a multi-scale representation of the image is constructed using a scale-space theory. In a scale space, an image can be represented at different scales parameterized by the size of the smoothing kernel. The kernel size is controlled by its scale parameter, which also decides the size of image spatial structures that will be smoothed away in the corresponding scale-space level. The most widely used scale-space is called Gaussian scale-space with the Gaussian function as the kernel. Given any 2D images $I(x,y)$, its scale-space representations $I(x,y,\delta)$ are defined as convolutions of the original image and the Gaussian kernel as:

$$I(x,y;\delta)=(g_{\delta}*I)(x,y) \tag{17}$$

Where g_{δ} is the Gaussian kernel with size δ as $g_{\delta}=\frac{1}{2\pi\delta}e^{-(x^2+y^2)/(2\delta)}$. When $\delta=0$, $I(x,y,\delta)$ is the original image itself. When δ increases, $I(x,y,\delta)$ is the result of smoothing the image with a larger filter and more details of the images are removed. Fig.8 shows three representations smoothed by a Gaussian kernel with an increasing δ. It can be seen that details about the face, clothes and backgrounds are smoothed out and the overall contour of the person is kept from the bottom image to the top image.

After getting multi-scale representations of the object, the HOG features can be extracted for each representation individually. For simplicity, the scale parameter is omitted and the image of current scale is noted as $I(x,y)$. First, the gradient map is calculated, which will be filtered by the edge-map and the contour-map to remove trivial gradients. The orientations are computed on the remaining gradients, and the histograms of orientations can be generated accordingly. The magnitudes of gradients are calculated by the convolution of the image with 2D differential of Gaussian (DOG) as $mag(I)=\sqrt{(\partial I/\partial y)^2+(\partial I/\partial x)^2}$. To suppress the trivial gradients, we hope to keep the gradients that are local maximums only. Therefore we filter the gradients by edge map and the gradients that are identified as edges will be kept. To fine tune the gradients further, the contour map is applied to remove the isolated gradients and backgrounds. Fig.8 also shows the gradient-map without pruning and the corresponding contour map. It can be seen that small gradients are filtered out.



(a) (b) (c)

(d)

Fig. 8. An example of extracting MSHOG features. (a) the original image with a person to be marked; (b) Representations of three scales are shown from top-scale to bottom-scale, which corresponds from the coarse contour to the fine image; (c) The edge and contour maps of the top-scale of (b); (d) the MSHOG vector of the top-scale representation. The image is divided into 16 grid cells and each cell contains a histogram of 8 orientations, which gives 128-length vector for each scale (as the x axis). Inside each cell, the histogram is normalized (as the y axis). Therefore, the complete MSHOG with three scales is a 384-length vector.

Then we calculate orientations for the filtered gradients map and construct the histogram of orientation gradients. We adopt histograms with 8 channels spread over 0 to 360 degrees. The whole image is divided into a number of cells evenly and each cell contains a separate histogram. Many HOG-based algorithms extract histograms on cells with a fixed size, which will be affected by the image size and the width-height ratio. Our method is independent of such parameters. Inside each cell, the histogram is then normalized over all channels for invariance under intensity changes. Fig.6 shows the HOG of one scale of 16 cells and 8 channels, given the feature vector with a length of 128 dimensions. The MSHOG with three scales will have 384 dimensions totally.

For any given image, we construct multi-scale representations and extract HOG for each scale, which generates the multi-scale HOG of the image. Then we apply the MSHOG as feature vectors to train the FBTP-NN for object recognition tasks

C. Experiments on MIT Pedestrians Dataset

Additional experiments have been conducted to evaluate the performance of the MS-HOG based feature extraction and the FBTP-NN model for classification in object recognition using MIT pedestrians dataset [32] with 924 images of people, where 2772 background patches (three patches per image) are randomly extracted as negative data. Fig.9 shows some examples of adopted images. Then different ratios of data are randomly chosen from datasets as training samples, and the rest for testing samples.

Table VII shows the comparison results of the recognition rates of our proposed method (MS-HOG plus FBTP-NN) and a few selected state-of-the-art algorithms. Combining the body parts model with SVM can provide up to 88% recognition rate [33]. In [34], a probabilistic assembly of robust part detectors was applied with boosting, and the recognition rate is 87% with one false positive per 1.8 image. In [35], a PCA-based reconstruction combined with a SVM can reach a recognition rate of 90.69% (99.02% was claimed with more false positive tolerance). The proposed FBTP-NN combined with the MSHOG can achieve a recognition rate of 98.02% with 2% false positive misclassification rate of the background patches. Each image adopts MSHOG features of 3 scales with 36 cells for each scale. 370 image samples are used for the experiment where 40% is for training and 60% for testing. Under the same training ratio and training iterations, the feed-forward neural network (FFNN) combined with the MSHOG can achieve a correct recognition rate of 94.60% with 3% false positive. Some near-perfect recognition results using a fine tuned SVM was reported in [31] with more than 1000 images for training in [31].

Fig. 9. Some examples of the MIT pedestrian dataset.

TABLE VII. PERFORMANCE COMPARISON WITH ALTERNATIVE ALGORITHMS

	Body model + SVM [33]	Parts+ Boosting [34]	PCA+ SVM [35]	MSHOG + FFNN	MSHOG + FBTPNN
Recognition Rate (%)	88	87	90.69	94.60	98.01

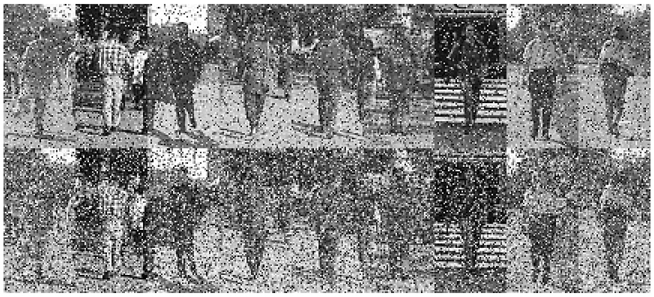


Fig. 10. Some examples of distorted images with salt and pepper noises.

TABLE VIII. RECOGNITION RATE OF MSHOG + FBTPNN FOR MIT PEDESTRIAN DATASET WITH DISTORTIONS

Training Data	50% Data
15% noise	95.89%
30% noise	92.01%

To test the robustness of the proposed method to noise in images, the image data are intentionally distorted by adding salt and pepper noise. Fig.10 shows some examples of the distorted data with 15% or 30% pixels contaminated by noise. In some images the person contour has been distorted and is hard to recognize even by human. Table VIII shows the recognition results of the proposed method (MS-HOG plus FBTP-NN) on the MIT pedestrian datasets with the training ratio of 50% and distortion ratios of 15% and 30%. It can be seen from Table VIII that we can still achieve reasonable good recognition performance even with 30% pixels being distorted by noise. The main reason for this is that the MS-HOG based feature extraction method can capture the global information of the object so that the noise affection can be alleviated, and the top-down estimation of the FBTP-NN model can compensate the distortion at some level.

D. Experiments on Caltech Dataset

In this subsection, we evaluate the proposed algorithms on multi-category object recognition by using the MIT pedestrians dataset and Caltech dataset with motorcycles, airplanes, cars and faces. Fig.11 shows some examples from the Caltech dataset. Although these images have various sizes and width-height ratios, we can apply the same MS-HOG feature extraction method on them without tailoring scanning windows from dataset to dataset.

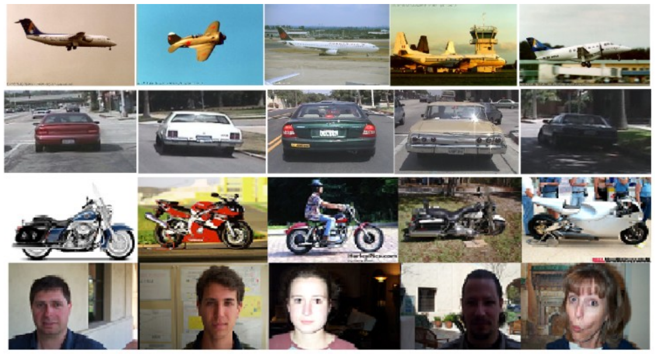


Fig. 11. Some image examples of Caltech datasets.

TABLE VIIII. PERFORMANCES COMPARISON BETWEEN SS-HOG AND MS-HOG FOR TWO-CLASS EXPERIMENT

Category	SSHOG	MSHOG
Plane	63.2%	85.5%
Car	96.1%	99.0%

First we evaluate the efficiency of the MSHOG feature. We choose airplane and car as objects and pick out 100 images for each category. For each image, we calculate the single-scale HOG (SS-HOG) by using the procedures described in Section VI.B on the original images, which turns out as a 128-length vector by applying 16 cells and 8 orientations for each cell. It is different to the HOG described in [31], which scanned the image with a fixed-size window and generated much larger histogram vectors. The MS-HOG features with 3 scales, 16 cells and 8 orientation channels are also extracted on all images for comparison. For simplicity we just demonstrate the experimental results with 1500 training loops and 25% training ratio. Table VIIII shows that apparently the recognition rate has been improved by using the MS-HOG features over SS-HOG. The recognition rate of plane has been raised by 22.3%. It is worth noting that the computational cost of the MS-HOG is low due to its sparse feature extraction. All the above experiments only adopt 16 cells over each image. More cells should be able to provide more detailed information about the object.

Finally, the entire Caltech dataset with 4-category i.e., car, human face, motor and plane images, are conducted and all images are applied. Table X lists the comparison results of the proposed method with a few the state-of-the-art approaches reported recently. It can be seen from Table X that the proposed algorithm can achieve a comparable recognition rate. More importantly, instead of constructing different classifiers for different classes, such as SVM-based classifiers, our algorithm is applicable to multi-category datasets using one single model. On the other hand, we have not found many approaches using neural networks as classifiers on the Caltech dataset, which can be compared with our algorithm. In [36] a three-layer feed-forward neural network is applied on face data of Caltech dataset, and it can achieve a recognition rate of 84.44%, which is much lower than the recognition rate using the proposed method listed in Table X. We also applied the FF-NN combined with the MS-HOG on this 4-class recognition problem. However the adopted FFNN was unable to recognize 4 classes simultaneously at a reasonable level and the results are omitted here.

TABLE X. PERFORMANCE COMPARISON ON CALTECH DATASET

Class	Constellation [2]	Boosting context[37]	Forest-ECOC [38]	MSHOG+ FTBPNN
Car	90.3	96.9	99.35	99.42
Face	96.4	89.5	97.72	96.68
Motor	92.5	95.0	93.58	94.92
Plane	90.2	94.5	92.50	95.97

VII. CONCLUSION AND FUTURE WORK

In this paper, a novel neural network model, called FBTP-NN, with both bottom-up and top-down information processing pathways is proposed for object recognition. Instead of applying symmetric weights or interpolating the weights of two pathways heuristically, a joint probability distribution is produced to describe the fusion of neural activities in bi-directional pathways. Correspondingly, a cost function is constructed to contain both the recognition label error and the data discrepancy based on this joint probability distribution. Then a learning method is proposed to minimize the cost function by updating the weights of both pathways. To improve the recognition rate, a new MS-HOG based feature extraction method is developed. Various experimental results demonstrate the efficiency and robustness of the proposed algorithms using normal datasets and heavily distorted dataset.

In the future, we will evaluate the proposed model on more complex datasets like Caltech 101 and compare its performance with the state-of-the-art approaches. We would also like to extend the proposed algorithm for more advanced recognition scenarios that may include the rotations, transportations and multiple objects. In addition, we intend to investigate more advanced fusion techniques from new neuron science evidences. Beyond the visual object recognition, the proposed model can also be applied to other real-world applications, such as data reconstruction, data synthesis, and knowledge-based reasoning.

REFERENCES

[1] B. Ommer and J. M. Buhmann, "Learning the Compositional Nature of Visual Objects," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8.

[2] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003, pp. 264-271.

[3] W. Gang, Z. Ye, and F.-F. Li, "Using Dependent Regions for Object Categorization in a Generative Framework," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1597-1604.

[4] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *IEEE International Conference on Computer Vision*, 2003, pp. 750-757.

[5] Z. Hao, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2126-2136.

[6] P. Viola and M. Jones, "Robust real-time face detection," in *IEEE International Conference on Computer Vision*, 2001, pp. 747-747.

[7] H. Pan and L. Z. Xia, "Efficient Object Recognition Using Boundary Representation and Wavelet Neural Network," *IEEE Transactions on Neural Networks,* vol. 19, pp. 2132-2149, 2008.

[8] A. Georgieva and I. Jordanov, "Intelligent Visual Recognition and Classification of Cork Tiles With Neural Networks," *IEEE Transactions on Neural Networks,* vol. 20, pp. 675-685, 2009.

[9] G. Xin, Z. Zhi-Hua, and K. Smith-Miles, "Individual Stable Space: An Approach to Face Recognition Under Uncontrolled Conditions," *IEEE Transactions on Neural Networks,* vol. 19, pp. 1354-1368, 2008.

[10] F. Scarselli, M. Gori, T. Ah Chung, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks,* vol. 20, pp. 61-80, 2009.

[11] A. K. Engel, P. Fries, and W. Singer, "Dynamic predictions: Oscillations and synchrony in top-down processing," *Nat Rev Neurosci,* vol. 2, pp. 704-716, 2001.

[12] S. Treue, "Visual attention: the where, what, how and why of saliency," *Current Opinion in Neurobiology,* pp. 428-432, 2003.

[13] S. Grossberg, "Competitive learning: from interactive activation to adaptive resonance," in *Connectionist models and their implications: readings from cognitive science*, ed: Ablex Publishing Corp., 1988, pp. 243-283.

[14] J. J. Hopfield, "Neural networks and phsical systems with emergent collective computational abilities," *Biophysics,* vol. 79, pp. 2554-2558, 1982.

[15] D. M. Beck and S. Kastner, "Top-down and bottom-up mechanisms in biasing competition in the human brain," *Vision Research,* vol. 49, pp. 1154-1165, 2009.

[16] D. Chen, L. Zhang, and J. Weng, "Spatio-Temporal Adaptation in the Unsupervised Development of Networked Visual Neurons," *IEEE Transactions on Neural Networks,* vol. 20, pp. 992-1008, 2009.

[17] M. Ursino, E. Magosso, and C. Cuppini, "Recognition of Abstract Objects Via Neural Oscillators: Interaction Among Topological Organization, Associative Memory and Gamma Band Synchronization," *IEEE Transactions on Neural Networks,* vol. 20, pp. 316-335, 2009.

[18] J. Duncan, G. Humphreys, and R. Ward, "Competitive brain activity in visual attention," *Current Opinion in Neurobiology,* vol. 7, pp. 255-261, 1997.

[19] Y. Zheng, Y. Meng, and Y. Jin, "Fusing Bottom-up and Top-down Pathways in Neural Networks for Visual Object Recognition," presented at the IEEE International Joint Conference on Neural Networks 2010.

[20] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled Hybrids of Generative and Discriminative Models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 87-94.

[21] H. Grabner, P. M. Roth, and H. Bischof, "Eigenboosting: Combining Discriminative and Generative Information," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8.

[22] Y. LeCun, H. Fu Jie, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. 97-104.

[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science,* vol. 313, pp. 504-507, 2006.

[24] M. W. Spratling and M. H. Johnson, "A feedback model of perceptual learning and categorization," *Visual Cognition,* vol. 13, pp. 129 - 165, 2006.

[25] E. Salinas and T. J. Sejnowski, "Book Review: Gain Modulation in the Central Nervous System: Where Behavior, Neurophysiology, and Computation Meet," *Neuroscientist,* vol. 7, pp. 430-440, October 1, 2001 2001.

[26] K. Fukushima, "Neural network model restoring partly occluded patterns," *International Journal of Knowledge-Based and Intelligent Engineering Systems,* vol. 8, pp. 59-67, 2004.

[27] G. E. H. P. Dayan, R, Neal, and R.S. Zemel, "The Helmholtz Machine," *Neural Comput.,* vol. 7, pp. 1022-1037, 1995.

[28] H. Gish, "A probabilistic approach to the understanding and training of neural network classifiers," *1990 International Conference on Acoustics, Speech, and Signal Processing*, pp. 1361-1364 vol.3.

[29] *Netlab*. Available: http://www1.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/

[30] D. G. Lowe, "Object recognition from local scale-invariant features," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1150-1157.

[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[32] MIT. *MIT Pedestrian Dataset*. Available: http://cbcl.mit.edu/software-datasets/PedestrianData.html

[33] R. Ronfard, C. Schmid, and B. Triggs, "Learning to Parse Pictures of People," ed, 2002, pp. 700-714.

[34] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human Detection Based on a Probabilistic Assembly of Robust Part Detectors," presented at the ECCV, 2004.

[35] L. Malagón-Borja and O. Fuentes, "Object detection using image reconstruction with PCA," *Image and Vision Computing,* vol. 27, pp. 2-9, 2009.

[36] L. Thai Hoang and B. Len Tien, "An approach to combine AdaBoost and Artificial Neural Network for detecting human faces," in *IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 3411-3416.

[37] J. Amores, N. Sebe, and P. Radeva, "Fast Spatial Pattern Discovery Integrating Boosting with Constellations of Contextual Descriptors," presented at the Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02, 2005.

[38] S. Escalera, O. Pujol, and P. Radeva, "Boosted Landmarks of Contextual Descriptors and Forest-ECOC: A novel framework to detect and classify objects in cluttered scenes," *Pattern Recognition Letters,* vol. 28, pp. 1759-1768, 2007.