International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:7, No:4, 2013

Analysis of Relation between Unlabeled and Labeled Data to Self-Taught Learning Performance

Ekachai Phaisangittisagul and Rapeepol Chongprachawat

Abstract—Obtaining labeled data in supervised learning is often difficult and expensive, and thus the trained learning algorithm tends to be overfitting due to small number of training data. As a result, some researchers have focused on using unlabeled data which may not necessary to follow the same generative distribution as the labeled data to construct a high-level feature for improving performance on supervised learning tasks. In this paper, we investigate the impact of the relationship between unlabeled and labeled data for classification performance. Specifically, we will apply difference unlabeled data which have different degrees of relation to the labeled data for handwritten digit classification task based on MNIST dataset. Our experimental results show that the higher the degree of relation between unlabeled and labeled data, the better the classification performance. Although the unlabeled data that is completely from different generative distribution to the labeled data provides the lowest classification performance, we still achieve high classification performance. This leads to expanding the applicability of the supervised learning algorithms using unsupervised learning.

Keywords—Autoencoder, high-level feature, MNIST dataset, self-taught learning, supervised learning.

I. Introduction

MACHINE learning plays the major roles in many areas of application: classification, regression, and clustering. It mainly focuses on implementing algorithms capable of learning or adapting their structure (parameters) based on a set of observed data from physical system. The task of learning from labeled data is basically called *supervised learning* that has been widely used in many applications. In supervised learning, the state-of-the-art algorithms for recognizing handwritten images require thousands of labeled handwritten images and similarly, face recognition uses thousands of labeled face images for training the supervised learning algorithms. In general, the supervised learning algorithms are implemented based on the assumption that the training and testing data are drawn from the same generative distribution. However, in many real-world problems, this assumption may not always hold true leading to degradation of the generalization performance. Another challenging problem in supervised learning arise when the number of labeled training data is small or expensive to collect such as object images, speech data, DNA microarray data. So, the supervised learning is limited on that situation. Based on M. Banko's and E. Brill's research [1], they found that to get better performance more labeled data is needed to provide to the learning algorithm. Sometimes, an aphorism "it is not who has the best algorithm that wins. It's who has the most data" has been coined in machine learning community. Consequently, even an inferior learning algorithm can often outperform a superior one if it has more training data [2].

For complex applications in computer vision, suppose we want to implement the machine learning model for car recognition. Suppose input to a learning algorithm consists of a set of 100×100 pixel image which can be represented as a high-dimensional pixel intensity vector $x \in \mathbb{R}^{10,000}$. In practice, if we have enough number of labeled training data, it should be possible to achieve high performance from the supervised learning algorithm. However, this prediction is not easy in practice because a slight shift on the camera angle, or a lighting conditions, or occlusion of the object, or shape variation can affect the prediction performance dramatically. So, we need the hypothesis function (h) that is capable of capturing useful image features and also is robust to many factors of variations. Based on those requirements, more complex learning algorithm is needed leading to more number of parameters to be determined. In such case, thousands of labeled data at least are required to train the learning algorithm. Having a small number of labeled training data for implementing the complex model highly causes the learning algorithm to be overfitting. As a result, the traditional supervised learning approach cannot perform well in this situation.

In learning process, each input image as a high-dimensional input vector x is forwarded to train the learning algorithm. In this case, each vector representing pixel intensity value in the image is usually called *low-level feature*. In practice, if we have large amount of labeled data, it should be possible to build a specific machine learning model to recognize whether the object in the image is car or non-car from the 10,000-dimensional input vector. In an alternative setting, suppose we have a feature representation that is able to represent special characteristic parts in the car image, e.g., car's wheel, a car's door. Providing these high-level features, it is much easier to predict by the learning algorithm with a high accuracy whether the image contains car or not. This formulation illustrates the impact of good feature representation for the supervised learning algorithms. However, it is not obvious to specify these high-level features in practice. Therefore, it is desirable to have a method that automatically constructs high-level feature representation without hand-tuning feature representation by experts for improving prediction performance of the supervised learning.

Inspired by these observations, there is a research group at Stanford University [3] who has proposed a new formal-

E. Phaisangittisagul is with the Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, Bangkok, 10900 Thailand, e-mail: fengecp@ku.ac.th

R. Chongprachawatare is a graduate student studying at the Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, e-mail: cshadowt@gmail.com

Fig. 1. Machine learning pipeline of image classification

ism, called *self-taught learning*, by using unlabeled data to construct a high-level feature for improving the supervised learning performance. The advantage of this new framework doesn't assume that the unlabeled data follows the same generative distribution as the labeled training data. However, there are several key ingredients to choose: number of features to learn, number of layers of features, learning rates, sparsity penalty, and relation of distribution between unlabeled and labeled data that are important to the self-taught learning performance.

In this paper, the objective is to investigate the impact of the distribution relation between unlabeled and labeled data on the classification performance. We will start by reviewing related work of high-level feature construction algorithms in section II, then move on to an unsupervised feature learning based on unlabeled data (self-taught learning) in section III. Experimental procedures and results are given in section IV and V, respectively. Finally, a conclusion is draw in section VI.

II. High-Level Feature Construction

It appears that number of labeled training data and good feature representation are very important to successfully implement the learning algorithm. One possible method that is normally used is to create a new set of features with containing relevant information to the learning algorithm. Feature extraction is a general method that has been widely used for dimensionality reduction and feature discovery. A basic concept of feature extraction is to map an input space ($x \in \mathbb{R}^n$) to a new feature space, $f(x) \in \mathbb{R}^d$ where $d \ll n$. In particular, instead of using raw input pixel intensities of the image $x \in \mathcal{X}$, we create a new set of informative input features from certain functions $\varphi_i(x)$ of the original input vector. Then, the learning algorithm will try to map the new features $f : \varphi(x) \rightarrow y$ rather than using the original n-dimensional input vector. So, the pipeline of object recognition task can be illustrated in Fig.1. Unfortunately, it has no straightforward recipe to choose the functions that provides good feature representation without trial and error approach by human efforts. So, it is cumbersome and expensive to obtain high-quality feature representation based on this method. Principle component analysis (PCA) is one of the unsupervised learning methods that is widely used in feature construction. In PCA, the new features are obtained by projecting the features into a low-dimensional subspace with maximal variant of the data. Other approaches proposed in literature such as kernel PCA, ISOMAX, independent component analysis (ICA), latent semantic analysis (LSA), etc.

In this paper, we will apply a multilayer neural network to perform feature construction [4],[5]. Basically, a neural network is a mathematical model which consists of an interconnected group of neurons and it processes the information using an activation function. Most of the neural network

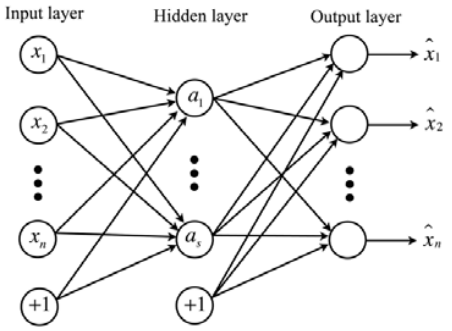

Fig. 2. An autoencoder feature extractor

models are arranged as layers such that outputs of each layer (n-th layer) forward to neuron in the next layer ($n+1$-th layer) until the last layer to produce the output(s) $y \in \mathcal{Y}$. This neural network will learn to mapping relationship between labeled input and output data so that the error is minimized. A special neural network designed by setting the target values to be equal to the inputs itself is called an autoencoder neural network [7] shown in Fig. 2. This network applies backpropagation algorithm to adjust the parameters by setting $y^{(i)} = h_{w,b}(x^{(i)}) = x^{(i)}$ where w, b are parameters of the neural network model. Basically, this network tries to learn an approximation to the identity function, so as to the output $y^{(i)}$ is similar to the input $x^{(i)}$. By limiting the number of hidden units less than n, the network is forced to learn a compressed representation of the input.

III. Self-Taught Learning

Due to the difficulty and expensive to obtain labeled data, some research group [3], [6] has proposed a new idea to learn a good feature representation from a massive amount of *unlabeled* data which is significantly easier to obtain than the given labeled data. This new learning framework is known as self-taught learning. At the beginning, assume we are given a labeled training data for object classification task of m examples which are drawn from a particular distribution $\mathcal{D}$.

$$T = \left\{ (x_l^{(1)}, y^{(1)}), (x_l^{(2)}, y^{(2)}), \ldots, (x_l^{(m)}, y^{(m)}) \right\}$$

where $y^{(i)} \in \{1, 2, \ldots, C\}$ is the class label corresponding to the input $x_l^{(i)} \in \mathbb{R}^n$.

In addition, suppose that a set of k unlabeled data is available in which these unlabeled data don't have any constraint to the labeled data.

$$\left\{ x_u^{(1)}, x_u^{(2)}, \ldots, x_u^{(k)} \right\}, \quad x_u^{(i)} \in \mathbb{R}^n$$

The approach to construct a higher-level feature representation based on self-taught learning consists of three separate stages as follows [3], [9]:

A. Learning higher-level feature representation:

To construct higher-level feature representation, a modified version of the sparse autoencoder algorithm based on Olshausen & Field [7] is applied to learn the basic elements that comprises in the input image (such as edges in natural image and strokes in handwritten characters [8]). The formulation in this step can be set up as follows:

$$\begin{aligned} \min_{a,b} J(a,b) &= \sum_{i=1}^{k} \|x_u^{(i)} - \sum_{j=1}^{s} a_j^{(i)} b_j\|_2^2 + \beta\|a^{(i)}\| \quad (1) \\ \text{s.t. } \|b_j\|_2^2 &\leq 1 \end{aligned}$$

The optimized parameters in (1) are a set of basis vectors $b = \{b_1, b_2, \ldots, b_s\}$ where $b_j \in \mathbb{R}^n$ and s is the number of hidden neurons. All activations in the hidden layer are represented as $a = \{a^{(1)}, a^{(2)}, \ldots, a^{(k)}\}$ where $a^{(i)} \in \mathbb{R}^s$. So, $a_j^{(i)}$ is the activation associated with the basis b_j of the input $x_u^{(i)}$. The goal of this optimization problem is two-fold: (i) to minimize the error between $x_u^{(i)}$ and a weighted linear combination of the basis b_j (ii) to force the activation vector to have low L_1-norm leading to most of $a^{(i)}$ elements to be nearly zero. Therefore, $x_u^{(i)} = \sum_{j=1}^{s} a_j^{(i)} b_j$ can be represented as a sparse weighted combination of small number of bases.

B. Constructing a higher-level feature to the labeled data:

After implementing sparse autoencoder algorithm to learn from the unlabeled data $x_u^{(i)}$, we obtain a set of bases b as described in the previous step. Then, a higher-level feature representation of the labeled training data $x_l^{(i)}$ is created based on a sparse linear combination of the bases b_j. This can be obtained by solving the following optimization problem:

$$a(x_l^{(i)}) = \min_{a^{(i)}} \|x_l^{(i)} - \sum_{j=1}^{k} a_j^{(i)} b_j\|_2^2 + \beta\|a^{(i)}\| \quad (2)$$

The objective of the optimization in (2) is to approximately express $x_l^{(i)}$ in terms of a sparse weighted sum of the bases feature b_j so that $a(x_l^{(i)})$ is a new feature representation of $x_l^{(i)}$. Consequently, the new high-level feature representation of the labeled data can be written as:

$$\hat{T} = \left\{(a(x_l^{(1)}), y^{(1)}), (a(x_l^{(2)}), y^{(2)}), \ldots, (a(x_l^{(m)}), y^{(m)})\right\}$$

C. Training high-level feature to classifier:

Finally, a supervised learning algorithm such as logistic regression, support vector machine, neural network, etc. can be trained by applying the high-level feature representation $\hat{T}$ from the previous step to obtain a classification model $h_{w,b}(a(x_l^{(i)}))$ that makes predictions on the $y^{(i)}$ values. For prediction of the testing sample x_{test}, we first have to compute the high-level feature $a(x_{test})$ by solving:

$$a(x_{test}) = \min_{a} \|x_{test} - \sum_{j=1}^{k} a_j b_j\|_2^2 + \beta\|a\| \quad (3)$$

Then, the learning algorithm will make a prediction by computing $h_{w,b}(a(x_{test}))$.

Now, let's see a detail of implementing a new high-level sparsity feature which is detailed by Andrew Ng et al [2], [9]. Suppose we have only a set of data $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$, where $x^{(i)} \in \mathbb{R}^n$. Let's define $a_j^{(2)}(x)$ to represent the activation of this hidden unit j in the second layer when the network is given a specific input x. Further, let's denote $\hat{\rho}$ be the average activation value of hidden unit j over all the k unlabeled data.

$$\hat{\rho}_j = \frac{1}{k} \sum_{i=1}^{k} \left[a_j^{(2)}(x^{(i)})\right] \quad (4)$$

To form succinct feature representation, the hidden unit's activations are enforced to mostly be zero (sparse) by setting the following optimization constraint.

$$\sum_{j=1}^{s} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \quad (5)$$

where ρ is called a sparsity parameter, usually setting to a small value close to zero. In addition, this penalty term can be written in the form of *Kullback-Leibler (KL)-divergence* by:

$$\mathrm{KL}(\rho||\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \quad (6)$$

The KL-divergence in (6) has the property that $\mathrm{KL}(\rho||\hat{\rho}_j) = 0$ if $\hat{\rho}_j = \rho$. Thus, our goal is to minimize this penalty term so that $a^{(2)}$ considered a new high-level feature has sparse representation. As a result, the overall cost function can be defined as:

$$J_{\text{sparse}}(w,b) = J(w,b) + \beta \sum_{j=1}^{s_2} \mathrm{KL}(\rho||\hat{\rho}_j) \quad (7)$$

where $J(w,b) = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{1}{2} \left\|h_{w,b}(x^{(i)}) - y^{(i)}\right\|^2\right)$ and β is a weight to control the sparsity penalty term. So, this new feature representation is referred to as a high-level representation and is sequentially forwarded to train the supervised learning algorithm.

IV. Experimental Procedure

In our experiments, we will use the self-taught learning framework with the sparse autoencoder as the feature extractor and softmax regression as a classifier for MNIST database of handwritten digit gray-scale images. The handwritten digit database consists of 60,000 examples for training and 10,000 examples for testing, respectively. Each handwritten digit image has been size-normalized and centered in a fixed-size 28×28 image field.

To implement the softmax regression [2] as a digit classifier, $\hat{T}$ is used to determine the model parameters. Basically, the softmax regression is designed to estimate the probability that $p(y = j|x^{(i)})$ for each value of $j = 1, \ldots, 10$ by:

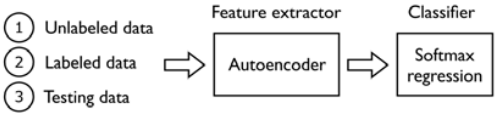

Fig. 3. Block diagram of the experiment procedure

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1|x^{(i)};\theta) \\ p(y^{(i)}=2|x^{(i)};\theta) \\ \vdots \\ p(y^{(i)}=10|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{10} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_{10}^T x^{(i)}} \end{bmatrix}$$

where $\theta = [\theta_1, \ldots, \theta_{10}]^T$ is a matrix parameter of the model. The parameters of the softmax regression can be obtained by optimizing the following cost function.

$$\begin{aligned} J(\theta) &= -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{10} 1\left\{y^{(i)}=j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right] \\ &+ \frac{\lambda}{2}\sum_{i=1}^{10}\sum_{j=1}^{n}\theta_{ij}^2 \end{aligned} \quad (8)$$

where $1\{\cdot\}$ is called an indicator function which provides value 1 if $1\{True\,statement\}$ and 0 otherwise. For classification of the handwritten digit image, the softmax regression will classify to a class which has highest probability of $p(y=j|x;\theta)$.

The procedure of the experiment can be displayed by Fig.3 which can be summarized by the following steps.

1) Learning high-level feature representation from unlabeled data: First of all, the sparse autoencoder shown in Fig.2 is trained by the unlabeled dataset to find a set of bases. In our experiment, the sparse autoencoder comprises 784 input neurons and one hidden layer with 200 neurons.
2) Constructing high-level features to the labeled training data: The new feature representation of the training handwritten digit images ($x_l^{(i)} \in \mathbb{R}^{784}$) is created by forwarding to the trained sparse autoencoder from the previous step. So, the new features can be expressed as $a(x_l^{(i)}) \in \mathbb{R}^s$ where $s = 200$.
3) Training high-level feature to classifier: A softmax regression is implemented by using the new feature representation $a(x_l^{(i)})$ where $l = 1, \cdots, m$ from the output of the trained sparse autoencoder as a training data.
4) Evaluating classification performance on testing data: For evaluation of classification performance, a testing data ($x_{test}^{(i)}$) will be mapped to a new set of high-level feature representation, $a(x_{test}^{(i)})$ by the trained sparse autoencoder. Finally, the trained softmax regression will make a prediction and we then compute the accuracy.

As mentioned earlier, the objective of this paper is to investigate the impact of the relation between the unlabeled data and the labeled data to the classification performance.

TABLE I
EXPERIMENTAL RESULTS ON MNIST HANDWRITTEN DIGIT DATASET

Unlabeled data domain	% classification accuracy
- Without unsupervised learning	92.64
- Natural images	93.47
- USPS handwritten database	93.82
- MNIST handwritten database	95.12

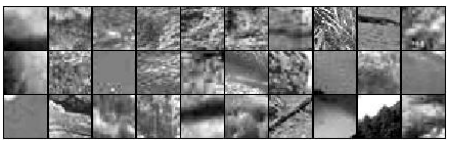

Fig. 4. Samples of natural image data to learn by sparse autoencoder

Fig. 5. Samples of USPS handwritten character unlabeled data to learn by sparse autoencoder

Fig. 6. Smples of MNIST unlabeled data to learn by sparse autoencoder

So, we apply 3 different sources of the unlabeled data to train the sparse autoencoder which has different degree of relation to the labeled data: (i) natural images , (ii) handwritten digits and english characters (”0”-”9” and ”A”-”Z”) from USPS database, and (iii) handwritten digit image (”0”-”9”) from training dataset itself. The numbers of unlabeled sample images of each data are 50,000 for natural image and MNIST handwritten data, and 1,404 for USPS handwritten character data, respectively. The samples of these unlabeled data are illustrated in Fig.4-6.

V. EXPERIMENTAL RESULTS

The classification results of MNIST handwritten digit with different unlabeled data learning by sparse autoencoder are reported in Table.I. It appears that classification accuracies of the new high-level features on all unlabeled data domains obtained from the sparse autoencoder outperform the raw feature alone which yields 92.64% in accuracy. In comparison among different unlabeled data of unsupervised feature learning, the new set of features from MNIST unlabeled data achieves top of the performance of 95.12%.

The visualizations of some bases of each unlabeled data learned by the sparse autoencoder are shown in Fig.7-9. It is not surprising that the learned bases from MNIST handwritten digit data appears to represent the digit texture which is most similar to the original handwritten digit data comparing

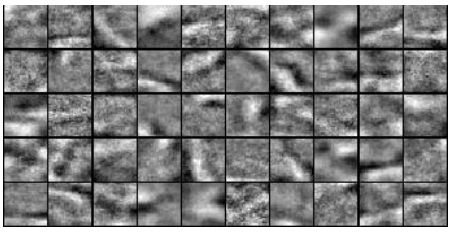

Fig. 7. Visualization samples of learned bases of 50 hidden neurons from natural image

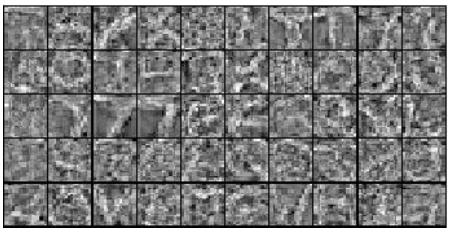

Fig. 8. Visualization samples of learned bases of 50 hidden neurons from USPS handwritten character

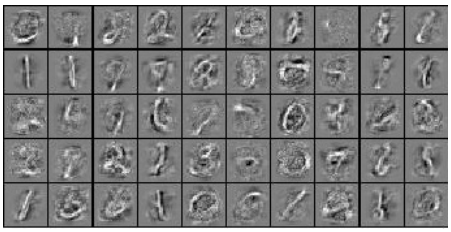

Fig. 9. Visualization samples of learned bases of 50 hidden neurons from MNIST handwritten digit

to the learned bases from other unlabeled data. Thus, the classification performance of new input features obtained from MNIST unlabeled data achieves highest accuracy. The reason of significant improvement is because the softmax regression classifier is allowed to learn the concept of digit structure rather than pixel intensity of the digit image.

In semi-supervised learning, the learning algorithm is implemented to improve the performance using labeled and unlabeled data but it requires strong assumption that both data must draw from the same distribution. Therefore, new features from MNIST unlabeled data in the experiment can be considered a kind of semi-supervised learning. However, while the USPS unlabeled data seems to have close relation to the MNIST handwritten digit data, it cannot be convincedly categorized into the semi-supervised learning since the USPS data is drawn from different space which has different characteristics such as different pen strokes and it is not gray-scale image.

VI. Conclusion

From the experiment, we can find that sparse autoencoder learned from unlabeled data can produce a higher-level feature representation which is useful to the supervised learning algorithm. In addition, the relation between labeled and unlabeled data has direct impact to the performance of the self-taught learning. Based on the experimental results, the closer relation between labeled and unlabeled data, the higher the performance accuracy will achieve. Thus, careful chosen unlabeled data can help to achieve even better performance improvement.

Although the semi-supervised learning achieves higher classification performance in handwritten digit than the self-taught learning algorithm, it can be applicable to a broader problem and also could be used to improve the supervised learning performance. In addition, the self-taught learning can be applied to mitigate the problem of having small number of labeled training data by using unlabeled data which is not necessary to share the same distribution with the labeled data. Consequently, the ability to use readily unlabeled data based on self-taught learning formalism has potential to make supervised learning significantly easier and widely applicable.

References

[1] Banko, M., Brill, E. "Mitigating the paucity-of-data problem: exploring the effect of training corpus size on classifier performance for natural language processing", 1^{st} Intl. conf. on Human language technology research, pp. 1-5, 2001.
[2] Ng A. *et al*., Lecture note on unsupervised feature learning and deep learning, http://deeplearning.stanford.edu/wiki, 2011.
[3] Raina, R., Battle, A., Lee. H., Packer, B., Ng, A. Y. Self-taught learning: Transfer learning from unlabeled data, In Proc. of the 24^{th} Intl. Conf. on Machine Learning, 2007
[4] Hinton, G. E., Salakhutdinov, R. R. "Reducing the dimensionality of data with neural networks", Science, 313, pp. 504-507, 2006.
[5] Hinton, G. E. "Learning multiple layers of representation", Trends in Cognitive sciences, vol.11, no.10, pp. 428-434, 2007.
[6] Pan, S. J., Yang, Q. "A survey on Transfer Learning", IEEE trans. on knowledge and data engineering, vol. 22, no.10, pp. 1345-1359, 2010.
[7] Olshausen, B. A., Field, D. J. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images", Nature, 381, pp. 607-609, 1996.
[8] Lee, H., Grosse R., Ranganath R., Ng A. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations", In Proc. 26th Intl. conf. on Machine Learning , pp. 609-616, 2009.
[9] Lee, H., Battle, A., Raina, R., Ng, A. Y. "Efficient sparse coding algorithms" NIPS. 19, pp. 801-808, 2007.

Ekachai Phaisangittisagul received his BSEE degree from King Mongkuts Insti- tute Technology of Thonburi (KMITT), Bangkok, Thailand, in 1996, the MS and PhD degrees in electrical engineering from North Carolina State University, Raleigh, in 2003 and 2007, respectively. He is currently an assistant professor of the electrical engineering department at Kasetsart University, Thailand. From 1996 to 2001, he worked as a design engineer for harmonic filter banks at ABB, Thailand. His research interests include pattern classification/recognition applications in robotics, machine olfaction and signal processing.

Rapeepol Chongprachawat received the BSEE degree from Kasetsart University in 2012. He is currently a graduate student in electrical engineering department at Kasetsart University, Thailand. His research interest reside in deep learning.