Chemometrics and Intelligent Laboratory Systems 189 (2019) 8–17

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

Image-based process monitoring using deep learning framework

Yuting Lyu[a], Junghui Chen[b,*], Zhihuan Song[a,**]

[a] State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, 310027, Zhejiang, China
[b] Department of Chemical Engineering and Research Center for Circular Economy, Chung-Yuan Christian University, Chung-Li, Taoyuan, Taiwan, 32023, Republic of China

ARTICLE INFO

Keywords:
Process monitoring
Fault detection
Deep learning
Deep belief network
Process images

ABSTRACT

With the advances in optical sensing and image capture systems, process images offer new perspectives to process monitoring. Compared to the process data collected by traditional sensors at local regions, process images enhance data-driven process monitoring a lot by capturing more significant variations in the whole space. Using the easily available industrial process images, a new deep learning framework based on the deep belief network (DBN) is proposed for feature extraction and timely fault detection. Unlike the traditional DBN methods inputting the images into the network directly, in the proposed framework, the sub-networks are used to extract local features from the sub-images. The global network fuses all of them for global feature extraction to remarkably improve the training efficiency without deteriorating the fault detection accuracy. Meanwhile, a new statistic is specially developed for the proposed deep learning framework. Finally a real combustion system is introduced to demonstrate the effectiveness of the proposed method.

1. Introduction

With the rapid development of the modern industry, industrial processes are becoming more and more complex. Moreover, these processes are highly automated because of the widely used applications of advanced process control systems. Although such development has brought considerable benefits, such as lower costs, production efficiency and consistent product quality, unexpected accidents can still happen and cause tremendous losses. As a result, promising progresses have been done in process monitoring and the fault diagnosis area with the increasing demand for safe operation and stable product quality. Among all the monitoring methods, data-driven methods without physical models or expert knowledge have won massive popularity in recent years. The most widely used data-driven techniques include principal component analysis (PCA), partial least square (PLS) and their extensions [1–5]. However, traditional data-driven methods are applied to the process measurements of physical properties, such as flow rates, temperatures, and pressures, which just provide limited process information because they are obtained at some local regions instead of the whole region. Furthermore, if engineers are lack of the process knowledge, the sensing points would be improper. Hence, the poor performance in process monitoring and fault diagnosis cannot be easily avoided.

With the advances in image capture systems and image processing techniques, process images have been increasingly emphasized in industrial processes. Lots of research papers have been done for the classification of defects and the predictions of variables that are difficult to measure. Prats-Montalbán and Ferrer utilized the images from ceramic tiles to accomplish defect classification [6]. Zhang and Song applied the smoke images to detect smoke concentration in the smelting process [7]. Yang et al. achieved a robust color measurement of flotation bubbles and predicted mineral concentration by taking advantage of froth images [8]. In addition to the variable prediction and the defect classification, such process images also offer new perspectives to image-based on-line monitoring in the area of combustion processes, forest products, rolled steel sheets, and so on. Inspired by these applications, process images are used in this paper instead of process measurements collected at the local regions to help monitoring larger spaces and recording significant variations of the operating unit which may be neglected by the sensors at the local regions. However, although the information from process images is enriched, how to convert the huge number of images into usable information is an enormous challenge.

In general, there are two types of image features used for process monitoring, namely spectral features and spatial features. With these features, the multivariate statistical process control (MSPC) model can be

[*] Corresponding author.
[**] Corresponding author.
E-mail addresses: jason@wavenet.cycu.edu.tw (J. Chen), songzhihuan@zju.edu.cn (Z. Song).

https://doi.org/10.1016/j.chemolab.2019.03.008
Received 17 January 2018; Received in revised form 14 March 2019; Accepted 14 March 2019
Available online 19 March 2019

established. Multivariate image analysis (MIA) [9] is the most widely used method of extracting spectral features. Bharati et al. applied MIA on lumber images to detect the occurrence of common lumber defects in forest products [10]. Yu and Macgregor combined MIA and PLS to efficiently extract information from the rapidly time-varying flame images and predict the boiler performance [11]. However, the MIA method still has some drawbacks. Firstly, the spatial information is lost in the process of unfolding each image to one-dimensional vectors. Secondly, the MIA method usually selects areas (ie. masks) in the score space to track the number of pixels. But these masks are always determined by experience. This will bring difficulties in practice.

Besides the spectral features, spatial features also play an important role in process monitoring by characterizing the spatial variation in pixel intensities. The wavelet texture analysis (WTA) and the grey level co-occurrence matrix (GLCM) are regarded as state-of-the-art methods for spatial feature extraction in the machine vision area. Recent papers introduced these algorithms to deal with industrial images. Wang and Ren utilized GLCM to extract the flame image features for the combustion condition recognition [12]. Bai et al. proposed a monitoring method for variable combustion conditions. Their method combined digital imaging, GLCM and random weight network techniques [13]. WTA [14] is widely used in the fields of paper formation [15], manufactured countertop slabs [16] and so on. A few approaches combining spectral and spatial features such as multi-resolution MIA (MRMIA) and soft color texture descriptors (SCTD) were also proposed [17,18]. Although these methods gave noticeable performance enhancement in applications, they suffered from the potential omission of some important features because of the inflexible feature extraction.

Recently, the deep learning [19,20] has been attracting extensive attention. With the capacity for extracting different levels of features for better classification and recognition, deep learning is considered as the most general and effective methodology for information extraction from massive data. One representation of deep learning algorithm is deep belief network (DBN) [21]. Using a series of stacked restricted Boltzmann machines (RBMs), DBN can be greedily trained layer by layer so that it can effectively extract high-level representations in an unsupervised manner. Moreover, these representations in different hidden layers of DBN are not designed by human engineers beforehand. They are automatically determined through unsupervised learning instead. DBN has been widely used in a variety of fields, such as speech synthesis, handwritten digits recognition, and many other applications [22–25]. In particular, DBN has also been applied to data-driven industrial processes for classification [26,27] and soft sensors [28]. However, although DBN was originally developed for image recognition, it is merely applied to simple black-and-white images and small-size color images in recent researches. Plenty of complex and large images which pose certain difficulty for efficient DBN training have not been investigated adequately. In order to fully exploit the extraordinary capacity of DBN in the high-level feature extraction, the conventional DBN model should be modified to accommodate such complex images.

In this paper, a novel DBN-based deep learning framework is proposed to extract features from process images and to monitor the current online process status. In this framework, each original image is segmented into small sub-images, which are subsequently inputted into corresponding sub-DBNs to extract local features (spectral and spatial features). Afterwards, all the local features are fused by a global DBN to extract global features. Meanwhile, a new statistic of the deep learning framework is first-ever proposed. In this deep learning framework, feature extraction and process monitoring can be realized in one composite model. This opens a new area in image-based process monitoring.

The remainder of this paper is organized as follows. The next section gives the background of the RBM and DBN. In Section 3, image preprocessing techniques are presented. Then a new DBN-based deep learning framework is designed for image segmentation and feature fusion. After that, a new statistic for this framework is defined and kernel density estimation is used to determine its control limits. Next, the whole monitoring procedure using process images and the proposed deep learning framework is summarized. In Section 4, the proposed method is evaluated using the flame images collected from a real combustion system. Finally, conclusions and the future research directions are presented.

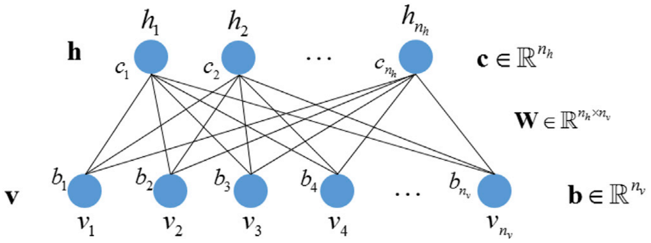


Fig. 1. The diagram of RBM.

2. Background

2.1. RBM model

RBM given in Fig. 1 is a shallow two-layer neural network which constitutes the building blocks of DBN. The first layer of RBM is called the visible layer representing the input data, and the second layer is the hidden layer representing the latent variables. In Fig. 1, n_v is the number of visible nodes, and n_h is the number of hidden nodes. $\mathbf{v} = (v_1, v_2, \cdots, v_{n_v})^T$ is the state vector of the visible layer, and v_i represents the value of the ith visible node. $\mathbf{h} = (h_1, h_2, \cdots, h_{n_h})^T$ is the state vector of the hidden layer, and h_j represents the value of the jth hidden node. The nodes in RBM are connected to each other across layers, but no two nodes of the same layer are connected. This means that there is no intra-layer communication, and this is exactly the restriction in a restricted Boltzmann machine.

In RBM, the input vector $\mathbf{v}$ is mapped into the hidden layer via nonlinear mapping, and each hidden node corresponds to a feature extracted by RBM. Thus, RBM is an unsupervised model, and its learning objective is to best fit the probability distribution of the observed data. $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ are the parameters of RBM, consisting of the weight matrix $\mathbf{W} = (w_{i,j}) \in \mathbb{R}^{n_h \times n_v}$, the bias vector of the visible layer $\mathbf{b} = (b_1, b_2, \cdots, b_{n_v})^T \in \mathbb{R}^{n_v}$, and the bias vector of the hidden layer $\mathbf{c} = (c_1, c_2, \cdots, c_{n_h})^T \in \mathbb{R}^{n_h}$. The energy function [29] of RBM is established as follows, indicating the energy level of the structure in the current condition.

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{n_v} b_i v_i - \sum_{j=1}^{n_h} c_j h_j - \sum_{i=1}^{n_v} \sum_{j=1}^{n_h} h_j w_{ij} v_i \tag{1}$$

This equation consists of the energy of all the visible nodes, the energy of all the hidden nodes, and the energy produced by the interaction between visible nodes and hidden nodes. And $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ are the parameters of the energy function. This function can be viewed as a measurement of the model stability given any possible configuration $(\mathbf{v}, \mathbf{h})$. The lower the energy value is, the more stable the model is and the higher compatibility between the input $\mathbf{v}$ and the feature $\mathbf{h}$ is. If any fault occurs, the characteristics of the fault data are quite different from those of the normal data; the feature $(\mathbf{h})$ of the fault data extracted by RBM will not be compatible with the input $(\mathbf{v})$. Consequently, RBM produces a large energy value.

According to the Boltzmann distribution [30], the probability of any possible visible and hidden vector $(\mathbf{v}, \mathbf{h})$ is given based on the energy function as

$$P(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v}, \mathbf{h})}}{Z}. \tag{2}$$

where $Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}$ is the normalizing factor to ensure the probability distribution sums to 1. Similarly, the marginal probability of a visible vector is the sum over all possible hidden layer configurations:

$$P(\mathbf{v}) = \sum_{\mathbf{h}} P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{3}$$

Since there are no direct connections within each layer, one is able to derive the conditional probabilities $P(\mathbf{v}|\mathbf{h})$ and $P(\mathbf{h}|\mathbf{v})$ as follows:

$$P(h_j = 1|v) = \sigma\left(c_j + \sum_{i=1}^{n_v} w_{ij}v_i\right) \tag{4}$$

$$P(v_i = 1|\mathbf{h}) = \sigma\left(b_i + \sum_{i=1}^{n_h} w_{ij}h_j\right) \tag{5}$$

where σ denotes the sigmoid function, $\sigma(x) = (1+e^{-x})^{-1}$.

As RBM is an unsupervised model, only the visible layer $\mathbf{v}$ is available and the hidden layer $\mathbf{h}$ needs to be estimated. Therefore, it is reasonable to set the training objective of RBM as highly fitting the probability distribution of the observed data, $P(\mathbf{v})$. The log-likelihood function of the total visible variables $\log L(\boldsymbol{\theta})$ is established as:

$$\ln L_{\theta,S} = \ln \prod_{i=1}^{n_s} P(\mathbf{v}^i) = \sum_{i=1}^{n_s} \ln P(\mathbf{v}^i) \tag{6}$$

where $\mathbf{S} = \{\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^{n_s}\}$ is the training set, n_s is the number of training samples and $\mathbf{v}^i = (v_1^i, v_2^i, \cdots, v_{n_v}^i)^T$, $i = 1, 2, \cdots, n_s$. Then the stochastic gradient descent (SGD) and contrastive divergence (CD) algorithms [31] are used for this optimization objective. The training results can be summarized as follows.

$$\Delta w_{ij} = \varepsilon\left(< v_i h_j >_{data} - < v_i h_j >_{recon}\right) \tag{7}$$

$$\Delta b_i = \varepsilon\left(< v_i >_{data} - < v_i >_{recon}\right) \tag{8}$$

$$\Delta c_j = \varepsilon\left(< h_j >_{data} - < h_j >_{recon}\right) \tag{9}$$

where ε is a learning rate, $< \cdot >_{data}$ denotes expectation over the training data, $< \cdot >_{recon}$ denotes expectation over the reconstructed data.

Although RBM with the energy function of Eq. (1) can handle binary data, such as handwritten images whose colors are either black or white, industrial images are much more complex with the pixel data typically represented by real-valued vectors. In order to tackle those data, binary RBMs are extended to the Gaussian–Bernoulli restricted Boltzmann machine (GRBM) [32]. The corresponding energy function is:

$$\begin{aligned} E(\mathbf{v},\mathbf{h}) &= \sum_{i=1}^{n_v} \frac{(v_i - b_i)^2}{2} - \sum_{j=1}^{n_h} c_j h_j - \sum_{i=1}^{n_v}\sum_{j=1}^{n_h} h_j w_{j,i} v_i \\ &= \frac{1}{2}\mathbf{v}^T\mathbf{v} - \mathbf{b}^T\mathbf{v} - \mathbf{c}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{v} \end{aligned} \tag{10}$$

In GRBM, Eq. (4) remains the same, but Eq. (5) becomes

$$P(v_i|\mathbf{h}) \sim N\left(b_i + \sum_{j=1}^{n_h} w_{j,i}h_j, 1\right) \tag{11}$$

Note that the differences between GRBM and RBM only lie in the energy function and in the way of generating the reconstructions, but all update rules for weights and biases remain the same. In this paper, all the RBMs in DBN are selected as the GRBM.

2.2. DBN model

DBN is composed of a series of stacked RBMs shown in the Fig. 2. The weights in DBN can be obtained by stacking a series of RBMs, whose hidden layer serves as the visible layer for the next. This structure aims at learning features at multiple levels, with features from higher levels of the hierarchy formed by the composition of lower level features [33,34]. Thus, the upper layers of DBN are supposed to represent more abstract features that explain the input data, whereas the lower layers extract simpler low-level features. With enough transformations of such composition, very complex nonlinear functions can be learned and progressively more complicated statistical structure in the input data can be represented. Meanwhile, the feature extractors in DBN are learned by the training process rather than being designed manually. The unknown structure of the feature extractor can be learned from the observed data to exploit the input distribution and to discover good representations. Therefore, the features at multiple levels in DBN are extracted automatically by the complex function mapping learned from the input data rather than being completely designed by engineers. However, good fault detection results of DBN are determined by the number of parameters (weights and biases), which in turn are determined by the number of neurons. To avoid overfitting and poor generation, early stopping is used to find out the parameters of DBN in this paper.

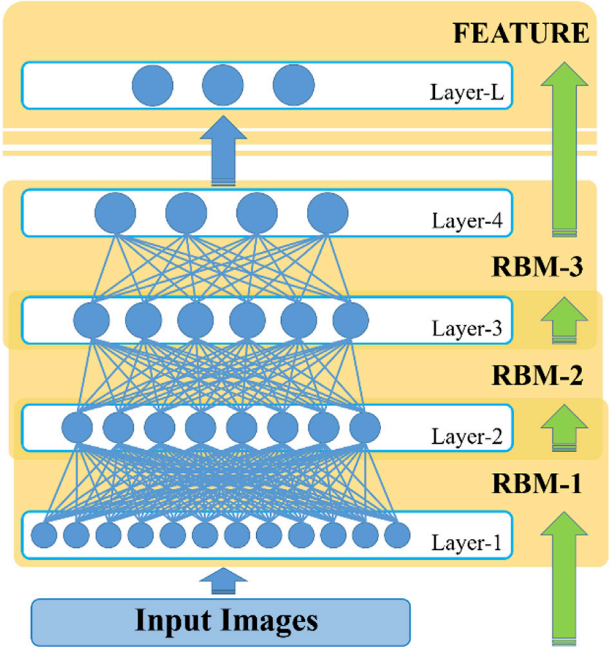


Fig. 2. The structure of DBN.

3. Methodology

3.1. Image pre-processing

Because the flame images with irregular blinks are usually unstable, it is difficult to get an ideal and reliable monitoring result. To eliminate the influence of such a problem, the time domain averaging filter, which takes the mean values of the image data in a time window as the input data, should be utilized before the modeling procedure. It is noted that this smooth procedure will not prevent the detection of the abnormal condition because the variation engendered by the abnormal event is much more intense than the normal pixel value.

In addition, the images obtained by digital color cameras are encoded in an RGB format in practical industrial processes. The color of each pixel in R, G, and B channels is quantified by numerical values from 0 to 255; in other words, each image corresponds to a three-dimensional matrix. However, DBN can only handle a one-dimensional vector or a batch of one-dimensional vectors during the training and testing procedure. In

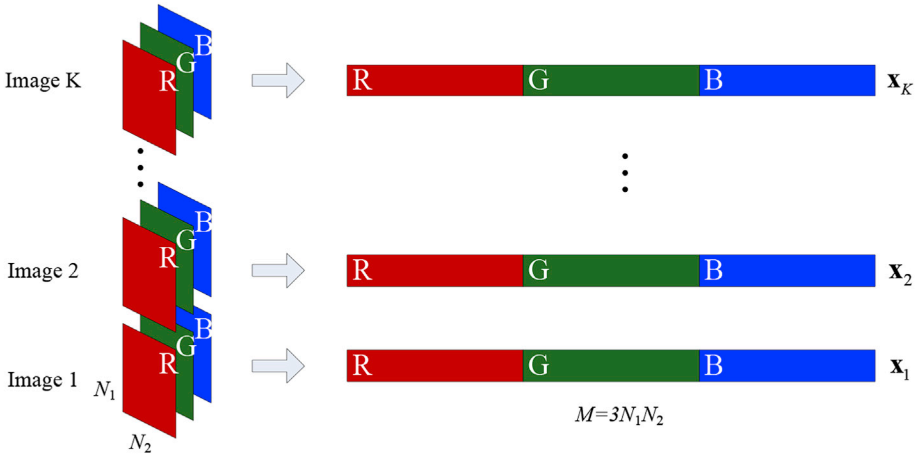


Fig. 3. Unfolded structures of the images with RGB channels.

order to extract useful features from the process images and perform on-line process monitoring by DBN, the three-dimensional images need to be reshaped into one-dimensional vectors as shown in Fig. 3. Assume the color images in three channels are of the same size $N_1 \times N_2$, then an RGB image can be reshaped into a row vector of the length $M = N_1 \times N_2 \times 3$. In other words, with a sequence of K process images, each RGB image corresponds to a one-dimensional vector $\mathbf{x}_k \in \mathbb{R}^M$, $k = 1, \cdots, K$, and the matrix input into DBN is $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_K]^T \in \mathbb{R}^{K \times M}$. After pre-processing the process images, the network is trained with these data layer by layer, and the parameters of DBN are determined.

Nonetheless, there will be two serious problems if the three-channel image is directly converted to a vector and inputted into DBN for training. First, the large dimension of the input vectors will lead to a huge amount of model parameters since DBN contains fully-connected layers. As a result, the model training efficiency will be significantly reduced. In addition, the on-line timely monitoring will be seriously influenced. Second, the network is usually over-fitting because the number of training samples is much less than the number of model parameters.

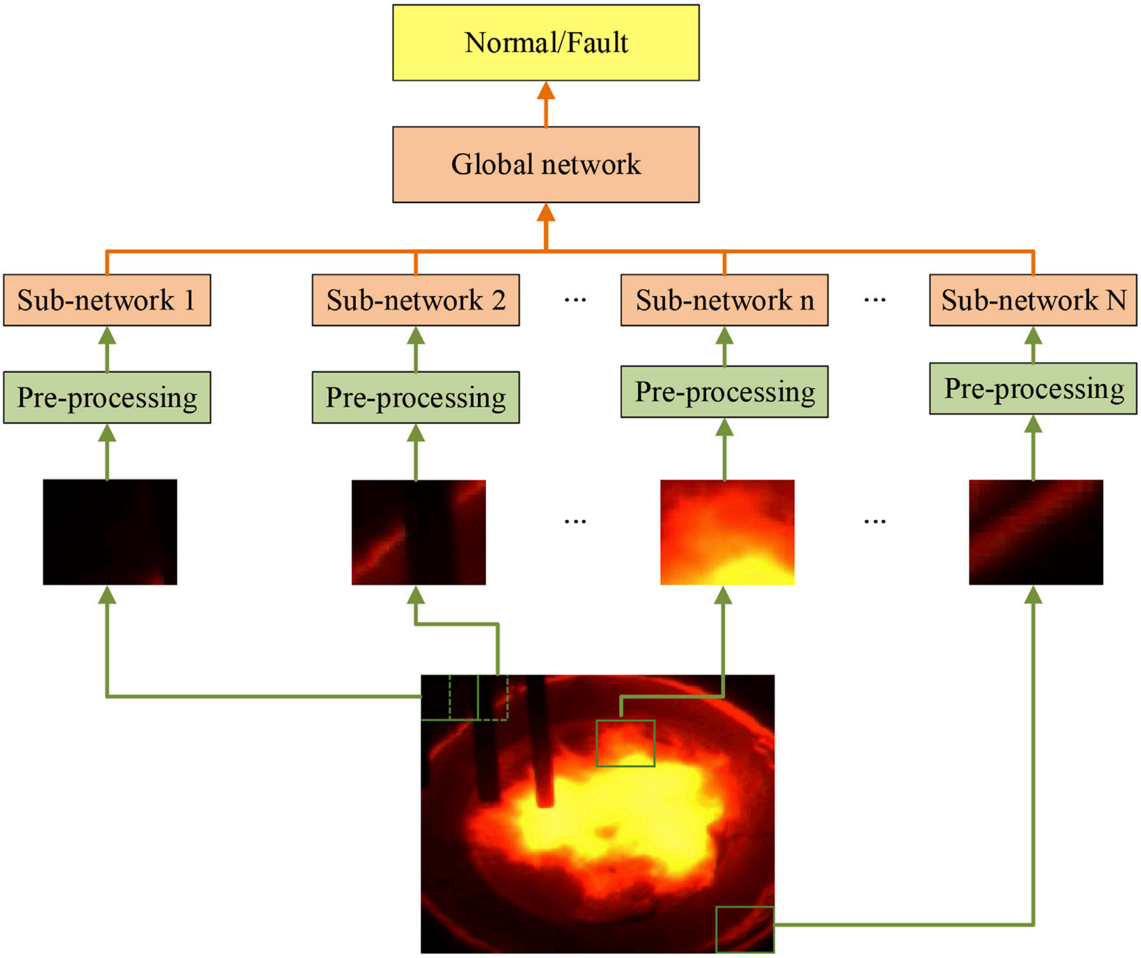


Fig. 4. Structure of the DBN-based deep learning framework.

Consequently, a new deep learning structure based on image segmentation and feature fusion is proposed in this paper.

3.2. Image segmentation and feature fusion

To overcome the curse of dimensionality and complexities, the original images are segmented into sub-images with a moving square window as shown at the bottom of Fig. 4. In this paper, the size of moving window is $m \times m$ and the moving stride (or the width of overlap region) is l. The quantity of sub-images N can be obtained as follows:

$$N = ((N_1 - m)/(m - l) + 1) \times ((N_2 - m)/(m - l) + 1) \tag{12}$$

With these sub-images, a new deep learning framework can be constructed as shown in Fig. 4. It contains N sub-networks and one global network that coordinates the outputs of all the sub-networks. In this paper, the sub-networks and global network are all DBNs. All the sub-images are first pre-processed by the aforementioned approach before being sent into the corresponding sub-networks. Then each sub-network is trained by the SGD method and the CD algorithm. To improve the efficiency of sub-network training, the SGD method divides the data set into smaller mini-batches stochastically and updates all the parameters in each epoch to search the global optimal solution.

After the optimum parameters of each sub-network are determined, the local features of the original images are obtained by sub-networks. During training, by the virtue of the first filter in the pre-processing, the variability of the extracted features is decreased, but there is still a large variation in the local feature space which poses certain difficulties on subsequent global feature extraction. Because DBN extracts different levels of features by stacking RBMs on the previous RBMs. This means that each RBM in DBN corresponds to a feature extractor. Thus, features in DBN can be smoothed at different feature spaces. But for fair comparisons with other traditional methods, only the outputs of all the sub-network are smoothed. That is, a second filter is applied to the extracted local features [11]. Subsequently, the global network fuses these local features from the sub-networks to extract global features for process monitoring. Such a deep learning framework possesses two remarkable advantages. On the one hand, all the sub-networks can be trained in a parallel way to greatly reduce the training time of the network. On the other hand, the optimal solution's search area for each sub-network is significantly reduced as the dimension of input variables decreases. Therefore, the trained network in the proposed structure is able to converge faster.

3.3. Construction of statistic

In this section, an on-line process monitoring scheme is carried out to detect the abnormal condition after off-line training. In conventional process monitoring algorithms, statistics play an important role in this procedure. Generally, for some traditional multivariate statistical methods like PCA and PLS, the Hotelling T^2 statistic and the *SPE* statistic are constructed to monitor the principal component space and its residual space, separately. Whether the test data are abnormal can be estimated by comparing the statistics of the data to the corresponding control limits.

In the proposed deep learning framework, process monitoring is carried out in the global network. For this global network, however, the application of the T^2 statistic is limited because DBN is a nonlinear feature extraction strategy without any assumption upon the distribution of latent features. Once the features are nonlinearly correlated, the T^2 statistic would not consider such nonlinearity as it is defined as the Mahalanobis distance between each sample and the center of all the samples. Thus, the T^2 statistic may be poor at monitoring performance. In this paper, a new statistic *Energy* is proposed as the energy of the top-level RBM in the global network.

$$E(\mathbf{v}_L, \mathbf{h}_L) = \frac{1}{2}\mathbf{v}_L^T\mathbf{v}_L - \mathbf{b}_L^T\mathbf{v}_L - \mathbf{c}_L^T\mathbf{h}_L - \mathbf{h}_L^T\mathbf{W}_L\mathbf{v}_L \tag{13}$$

where L denotes the *L-th* layer on the top of the global network. This statistic can be theoretically explained in two aspects. From the perspective of model training, RBM is trained with the objective of maximizing the log-likelihood function of $P(\mathbf{v})$, which is identified with minimizing the energy of the model. In other words, when no faults occur in online monitoring, the energy of the testing data is relatively low with small fluctuations. Once a fault occurs, the fault sample will produce a larger deviation from the statistical model and the energy of RBM will exceed its control limit. As for the model's physical meaning, RBM originates from the Ising model [35], and each variable in $\mathbf{v}$ and $\mathbf{h}$ corresponds to the atom in the Ising model, $\mathbf{W}$ corresponds to the interaction between atoms, $\mathbf{b}$ and $\mathbf{c}$ correspond to the influence of the external magnetic field. In the Ising model, the lower the energy is, the more stable the model will be. Similarly in RBM models, the normal operating condition (corresponding to the stable model state) is located within a certain range of the minimum points of the energy function. Thus, the stable bound of the energy function can be used to determine whether the input data are fault data.

The reason why we choose the top-level RBM in the global network to construct statistic is obvious. Each hidden layer in DBN extracts features from the outputs of its previous layer. As a result, the features extracted at the low levels may contain more irrelevant information, which may potentially deteriorate the monitoring effect. Thus, it is more suitable and beneficial to use the top-level RBM in the present study for online process monitoring. Besides the *Energy* statistic, the *SPE* statistic in the residual space is also adopted. It is based on the reconstruction error between the input of the global network $\mathbf{v}_1$ and its reconstruction $\widehat{\mathbf{v}}_1$, and is defined as follows.

$$SPE = (\mathbf{v}_1 - \widehat{\mathbf{v}}_1)^T(\mathbf{v}_1 - \widehat{\mathbf{v}}_1) \tag{14}$$

Finally, to estimate whether fault information exists at a sample point, the confidence limit of the *Energy* statistic and the *SPE* statistic should be determined. Because the distribution of these statistics are unknown, kernel density estimation [36], which is a non-parametric method to estimate unknown probability density function of a random variable, is used here to calculate the control limit of the statistics.

Here is the brief concept. Assume that given the measured (or estimated from other measured variables) $X = \{x_i\}_{i=1,\cdots,N}$ from a distribution with the density function $f(x)$, an estimate of the density at x can be calculated by

$$\widehat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(x - x_i) = \frac{1}{N\sigma}\sum_{i=1}^{N} K\left(\frac{x - x_i}{\sigma}\right) \tag{15}$$

where $\int K(x)dx = 1$. In this estimator, K_σ is a kernel function with a bandwidth σ, and it is generally the standard normal density function; N is the sample size. This estimator is able to estimate the density of any distribution as long as there are enough samples. Thus, with estimated $Energy_\alpha$ and SPE_α in Eqs. (13) and (14), the control limits $Energy_\alpha$ and SPE_α in this paper are consequently determined by taking the $1 - \alpha$ quantile from its corresponding probability density estimator, and typically $\alpha = 0.01$.

3.4. Procedure of process monitoring based on the deep learning framework

In summary, deep learning framework based process monitoring goes through the following steps:

(Modeling Phase)

Step 1. Acquire process images from the digital color camera under the normal operation; then smooth the images to eliminate the influence of instability and blink.
Step 2. Determine the parameters of segmentation and segment the original images into sub-images.
Step 3. Initialize the parameters of each sub-DBN, and input each sub-image into corresponding sub-DBN for training and local feature extraction.
Step 4. Initialize the parameters of the global DBN, and utilize all the outputs of sub-networks as input to train the global DBN.
Step 5. Calculate the *Energy* statistic and the *SPE* statistic of the normal image data, and determine the control limit of *Energy* and *SPE* by the kernel density estimation.

(On-line Monitoring Phase)

Step 6. Acquire testing images and smooth them, then input them into the proposed deep learning framework to extract useful features and calculate the monitoring statistics *Energy* and *SPE*.
Step 7. Determine whether a fault occurs by comparing the statistics of the testing images with the control limit of *Energy* and *SPE*.

4. Case study

4.1. Case description

In order to evaluate the proposed framework for image-based process monitoring, flame images collected from a real combustion furnace are used in this paper. The schematic diagram of combustion system is shown in Fig. 5. It consists of an experimental furnace, a burner, an air compressor, a fuel tank, a cooling system and a digital camera. The model of the burner is NA 5514-6, which is used in North American Manufacturing Company. The dimension of the experimental furnace is 2.5m × 2.5m × 1.5m. During the operation, the process data such as the steam pressure, the furnace temperature, and the air/fuel flow rates are automatically recorded through a SCADA system. Meanwhile, the concentrations of NOx, SOx, O_2 and CO in the exhaust gas are measured by a gas analyzer. The flame images in the furnace are captured by a digital color camera with the pixel size of 658 × 492 and the resolution of 24 bits per pixel.

In this system, the flame images are highly related to combustion efficiency and emission concentrations. Lower combustion efficiency leads to more energy waste. Meanwhile, emission concentrations exert an influence on the environment. Hence, the flame images, which provide sufficient information of the combustion system, are utilized to monitor the process. In this way, operators are able to restore the system timely once a fault occurs, which is beneficial for reducing the operating cost and environmental pollution.

4.2. Data preparation

In the present study, two types of fault are selected for research:

(1) Fault 1 is caused by the decrease of air ratio in fuel, a simple visualization and a comparison of five sequential images from the normal condition and the abnormal condition are respectively shown in Fig. 6(a) and Fig. 6(b). Since the abnormal operation behavior is minor, the differences between the normal and the abnormal conditions cannot be distinguished by directly observing the images.
(2) Fault 2 caused by aperture adjustment is also adopted. As shown in Fig. 5, the digital camera is installed outside the furnace to protect the camera from damage caused by high temperature and keep the camera from dust. However, the camera is easily affected by the changes in the intensity of the external light because of the installation location. Thus, the aperture should be adjusted in time to guarantee the quality of images. A set of improper aperture data is collected for testing.

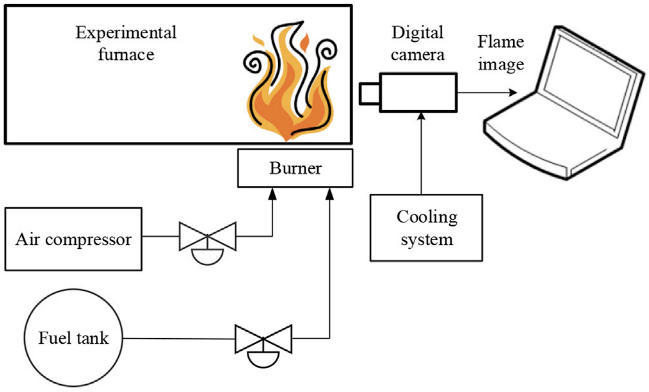


Fig. 5. Scheme of the experimental combustion system.

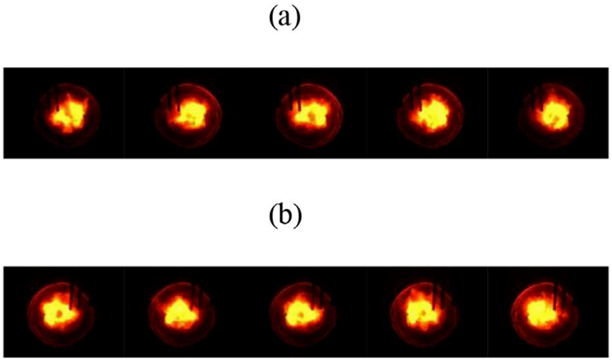


Fig. 6. Five sequential images of (a) normal and (b) abnormal conditions.

In order to discover such fault timely by the proposed method, a dataset containing 750 flame images, which corresponds to a recent period of normal process operation, is selected as the training set. To detect these two faults, two testing sets are used. The first testing set with 2100 flame images, which involves both the normal and the abnormal conditions (Fault 1), is taken to validate the effectiveness of the proposed deep learning framework and statistic. The second testing set contains 950 images with both normal and abnormal conditions (Fault 2).

4.3. Settings of the deep learning framework

The original flame images obtained from the aforementioned combustion furnace are of size 300 × 300, they are segmented into sub-images with a moving square window of size 60 × 60 and the moving stride is set as 20. Hence, the quantity of sub-images N can be obtained as 49 via Eq. (12). For simplicity, all the sub-networks are set to be the same structure. During the procedure of training DBN layer by layer, the initial weights in each layer are selected from a Gaussian distribution with zero mean and the standard deviation of 0.01. All the biases are initialized as zero. A deep 10800-300-50 sub-network is finally determined by trial and error to extract the local features from each sub-image. Then, the global network is trained in the same way as the sub-networks and its structure is determined as 2450-100-20.

4.4. Results and discussions

4.4.1. Comparison with DBN

To illustrate the efficiency and accuracy of the proposed framework, conventional DBN model is introduced for comparisons. In the

Table 1
The comparison of the model structure and the corresponding training time.

		Input variables	Model structure	Training time
Single DBN		300 × 300 × 3	270000-600-300-20	5693.147632s
Deep learning framework	Sub-DBNs	60 × 60 × 3	10800-300-50	80.062074s
	Global DBN	2450	2450-100-20	14.146096s

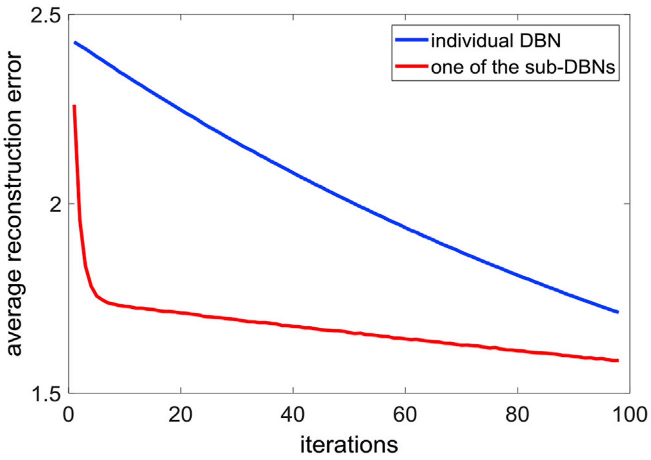


Fig. 7. The convergence curves of the first layers in different methods.

conventional DBN model, the images are directly put into an individual DBN without segmentation. To evaluate the model complexity, the number of input variables and the model structures for both methods are listed in Table 1. As shown in Table 1, the model structure of the proposed method is downsized after image segmentation, so the number of model parameters is significantly reduced. Thus, the training burden of the proposed method is consequently lower than that of the conventional DBN. Another advantage lies in the faster convergence speed. For visualized comparison, the convergence curves of the 1st RBM layer in both methods are shown in Fig. 7. As it can be seen in Fig. 7, the search area of optimum solution is narrowed down with the decrease of parameters numbers, which helps to find the global optimal value overall situation and converge at a faster rate.

The training time of both methods is also given in Table 1 for further comparison. Because all the sub-networks in the deep learning framework can be trained in parallel, the training time of all the sub-networks is equal to the longest training time among these sub-networks. The whole training time of the deep learning framework is the sum of the training time of sub-networks and the global network. In this case, the training time of our method is 94.21 seconds. Apparently, the proposed method outperforms conventional individual DBN in training efficiency.

However, the segmentation of images may potentially destroy the image features and have a negative effect on the fault detection accuracy. A natural idea is to get a trade-off between training efficiency and fault detection accuracy. Therefore, the model performance is tested with various sizes of moving square windows and moving strides. At last, the final size of moving window is set to be 60 × 60 and the final moving stride is set to be 20. To evaluate the feature extraction effect of the proposed deep learning framework, two monitoring methods are adopted. One uses the *Energy* statistic and the *SPE* statistic directly in the proposed deep learning framework (denoted as DLF), which is a global network with the integration of local DBN, and the other one applies PCA to the features extracted by the proposed framework (denoted as DLF-PCA). The monitoring effect of these two methods on the two testing datasets is measured. Furthermore, to show the advantage of DLF, one single DBN monitoring model with the *Energy* statistic and the *SPE* statistic is included for comparisons. For a quantitative comparison, the two indices, detection delay and fault detection rate (FDR), are used. Detection delay represents the time from the occurrence of the fault to the detection from the monitoring scheme. With this index, we are able to find the best model which can early detect the fault as soon as possible when something abnormal is affecting the process, so it is beneficial for reducing losses timely. Although the abnormal behavior can be detected by comparing the statistic of individual measurements with the control limits, the individual measurements are easily contaminated by the measured noises. To provide more reliable information, here an additional rule is included for detection delay. The rule states that a fault is declared when 10 consecutive observations exceed the control limits. The detection delay for these two testing sets is shown in Table 2. The symbol "–" indicates that the fault has not been detected. Another index, FDR, measures the percentage of the fault samples that have been identified as abnormal. For a fair monitoring performance comparison, FDR is used on the premise that the false alarm rates (FAR) of all the methods are identical. For details, FDR and FAR are calculated as follows.

Table 2
Detection delay of two faults based on different methods.

Data Type	Algorithm	Statistic	Data Set 1 (Fault 1)	Data Set 2 (Fault 2)
			Detection delay	Detection delay
Process Data	PCA	T^2	–	–
		SPE	–	–
Image Data	WTA-PCA	T^2	68	–
		SPE	–	
	SCTD-PCA	T^2	112	–
		SPE	–	**56**
	MRMIA-PCA	T^2	138	371
		SPE	204	239
	DBN	*Energy*	88	77
		SPE	64	111
	DLF-PCA	T^2	102	225
		SPE	–	69
	DLF	*Energy*	**34**	82
		SPE	42	116

$$FDR = \frac{N_{F,F}}{N_F} \tag{16}$$

$$FAR = \frac{N_{N,F}}{N_N} \tag{17}$$

where $N_{F,F}$ is the number of fault samples in the testing dataset that has been identified as abnormal, $N_{N,F}$ is the number of normal samples in the testing dataset that has been flagged as abnormal, N_F is the total number of fault samples in the testing dataset, and N_N is the total number of normal samples in the testing dataset. The quantitative fault detection rate and false alarm rate for these two testing sets are shown in Table 3. The false alarm rates of all the methods in Table 3 are kept around 0.01 by adjusting the corresponding $1-\alpha$ quantiles of the corresponding distribution.

For the first testing set (Fault 1), the visualized monitoring results are shown in Fig. 8. The blue curves represent the statistics of each testing sample and the red dashed lines represent the control limits. When the fault caused by the decrease in the air/fuel ratio occurs at the 700th sample, the combustion state deviates from the normal condition. The features of the flames are also accordingly influenced by the fault recorded in the flame images in real time. Thus, using the conventional DBN known for remarkable feature extraction capability, such a fault can be detected timely as shown in Fig. 8(a). In this figure, the *Energy* statistics and the *SPE* statistics of the majority of the samples are below the corresponding control limits before the 700th sample while they exceed the control limits after the fault happens at the 700th sample. In spite of

Table 3
Fault detection rates and false alarm rates of the combustion furnace based on different methods.

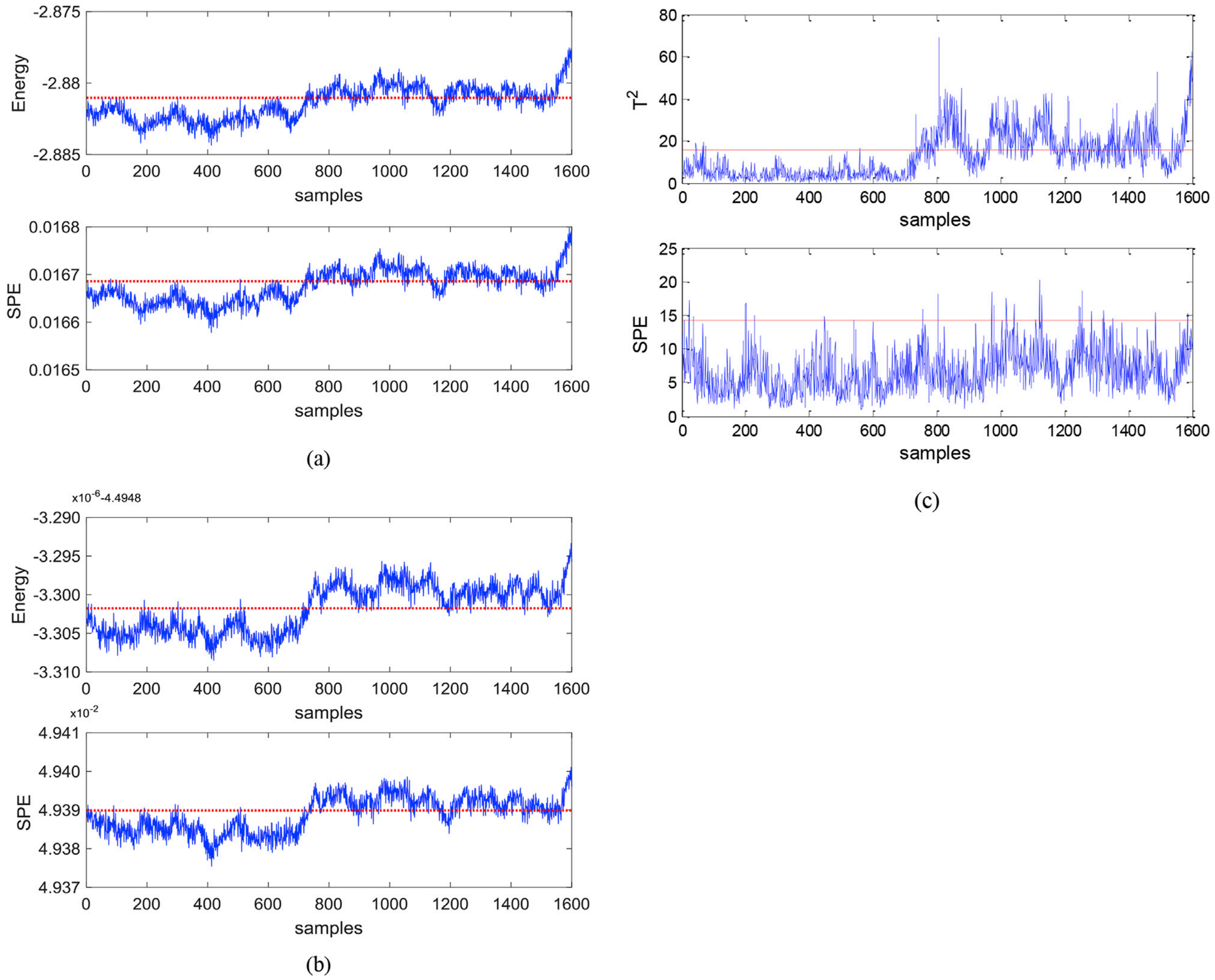
Data Type	Algorithm	Statistic	Data Set 1 (Fault 1)		Data Set 2 (Fault 2)	
			FDR	FAR	FDR	FAR
Process Data	PCA	T^2	0.0511	0.0100	–	–
		SPE	0.6189	0.0100	–	–
Image Data	WTA-PCA	T^2	0.8778	0.0100	0.0140	0.0111
		SPE	0.6589	0.0100	0.0520	0.0111
	SCTD-PCA	T^2	0.7800	0.0100	0.0440	0.0111
		SPE	0.5744	0.0100	0.8640	0.0111
	MRMIA-PCA	T^2	0.7244	0.0100	0.2920	0.0111
		SPE	0.7922	0.0100	0.4780	0.0111
	DBN	*Energy*	0.8044	0.0100	0.8280	0.0111
		SPE	0.7900	0.0100	0.8300	0.0111
	DLF-PCA	T^2	0.5922	0.0100	0.0620	0.0111
		SPE	0.0211	0.0100	**0.8888**	0.0111
	DLF	*Energy*	**0.9156**	0.0100	**0.8840**	0.0111
		SPE	0.7956	0.0100	0.7360	0.0111

the satisfactory detection result, using the single DBN directly for image-based process monitoring gives rise to the curse of dimensionality in the first layer of DBN. As a result, the efficiency of training and feature extraction is severely reduced. The monitoring effect of the proposed deep learning framework is shown in Fig. 8(b) and (c). Considering the quantitative monitoring effect in Tables 2 and 3 comprehensively, the proposed DLF not only signals Fault 1 in the shortest time but also achieves the highest detection rate. The performance of DLF-PCA is unsatisfactory in this dataset mainly because the features extracted by the proposed framework are not linearly related, which violates the assumption of PCA.

For the second testing set (Fault 2), the performance of DLF is slightly inferior to DLF-PCA, and both DLF and DLF-PCA outperform the performance of DBN in detection delay and the fault detection rate. Such monitoring results indicate that the global network makes up for the negative effect caused by such segmentation through fusing all the local features although the images are segmented with the purpose of lowering the training load. Thus, the proposed framework not only achieves more efficient training but also extracts useful features for process monitoring in both testing datasets. Based on these extracted features, how to conduct monitoring influences the two performance indices. In Tables 2 and 3, DLF-PCA performs better than DLF in Fault 2 but DLF-PCA only detects about half of the fault samples in Fault 1. Compared to the detection uncertainty of PCA among different types of faults, the proposed DLF attains a high detection rate and short detection delay in both testing datasets. This indicates that DLF is more universal for detecting different types of faults.

4.4.2. Comparison with traditional approaches

In order to demonstrate the good feature extraction ability of the proposed deep learning framework, some traditional feature extraction



(a)

(b)

(c)

Fig. 8. Process monitoring results of deep learning for the first testing set (Fault 1): (a) single DBN, (b) deep learning framework, (c) deep learning framework-PCA.

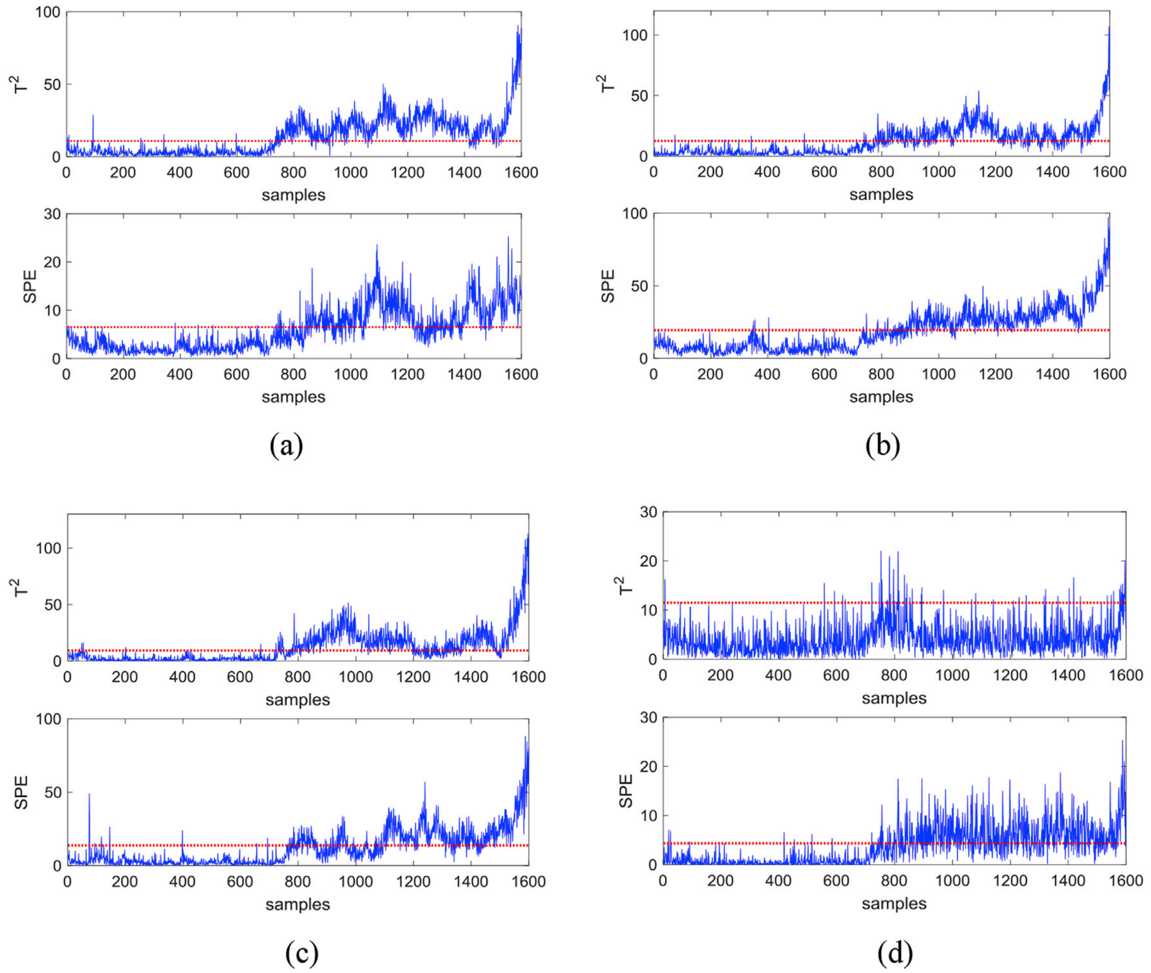


Fig. 9. Process monitoring results of the traditional methods: (a) WTA-PCA, (b) MRMIA-PCA, (c) SCTD–PCA, (d) PCA. WTA-PCA, MRMIA-PCA, SCTD–PCA are based on process images and PCA is based on process variable data.

methods for images, namely WTA, MRMIA, and SCTD, are also applied to furnace monitoring for comparisons. In WTA, each image is decomposed to the 3rd stage ($J = 3$) using Daubechies-1 (Db-1) wavelets, which is orthogonal and nearly symmetric. And the energy signatures of sub-images are employed as textural features. In MRMIA, PCA is first applied to the original multivariate image; then multi-resolution decomposition on each score image is conducted. The energy signatures are used as image features. In SCTD, a set of statistical features describing colors and textures, namely mean values of color bands and the 2nd to 5th order mathematical statistics, are extracted from the images. Since these methods can merely extract features from flame images, further analysis of these features is still performed by the PCA model. The time domain averaging filter is applied to the original image space and feature space, respectively. And the width of the time window is the same as the proposed method. In addition, the process variables (such as the concentration of CO, the concentration of NO, etc.) are also used with PCA for comparison. The statistics T^2 and *SPE* are constructed separately for WTA-PCA, MRMIA-PCA, SCTD–PCA, and process data based PCA, but the newly proposed statistic *Energy* and the *SPE* statistic are used in the deep learning framework. The performance of fault detection is tested by these models using two sets of testing data.

For better visualization, the detailed monitoring results of the Fault 1 case using four traditional models are shown in Fig. 9. The blue curves represent the statistics of each testing sample and the red dashed lines represent the control limits. Compared with the monitoring results in Fig. 8, all the traditional methods are also able to detect a majority of fault samples. Among them, the process data based PCA in Fig. 9(d) has relatively poor performance.

For a more accurate comparison, FDR and detection delay of the above different models for two testing sets is summarized quantitatively in Tables 2 and 3. The best detection performance of each fault among different models is shown in bold. For Fault 1, the DLF with the *Energy* statistic outperforms the other four traditional models in both indices. For Fault 2, DLF is slightly inferior to SCTD-PCA in terms of the detection delay but it achieves a better result in FDR than SCTD-PCA. Therefore, in general, the proposed DLF with the *Energy* statistic performs better in both fault cases than the other four traditional models. Although WTA-PCA has a good performance in Fault 1, it fails to detect Fault 2 as only spatial features can be extracted by wavelets. For SCDT, it achieves the smallest detection delay in Fault 2 while being inferior to the proposed method in detection rates both in Fault 1 and Fault 2. It is because SCTD can only extract global features but omit some essential local features. MRMIA works relatively well in Fault 1 but fails to detect Fault 2 and it is not sensitive to both faults. Its insufficiency mainly lies in the choice of wavelets and the number of decomposition stages. For process data based PCA, only Fault 1 is studied since Fault 2 is caused by the camera and the process remains normal. Fault 1 is seldom detected by the T^2 statistic. The detection performance of *SPE* is much better than that of the T^2 statistic, but there are still many fault samples remaining undetected because of the limited process information. If we just refer to the detection delay, Fault 1 is not declared. Therefore, the proposed method can extract features from images and model these features more accurately, and its process monitoring performance is superior in most cases.

5. Conclusion

This paper develops a novel DBN-based deep learning framework to monitor industrial process in real time. Instead of using process variables, which just cover limited information of a process, this paper uses the process images captured by the digital camera to train the framework and achieve process monitoring with the newly constructed statistic. Compared to the single DBN, the proposed framework can avoid the curse of dimensionality by segmenting the original images into sub-images. The number of model parameters in sub-network is consequently reduced to a great extent, which significantly lowers the model training burden and results in a faster convergence speed. What's more, all the sub-networks in the proposed deep learning framework can be trained in parallel, the whole training time of the deep learning framework is much shorter than that of the single DBN. In case that the segmentation of images potentially destroys the image features, the global network is introduced for local feature fusion to eliminate such a negative effect. Compared to the traditional image feature extraction methods, the advantages of the proposed method mainly reflect in the excellent feature extraction ability of DBN. It is more flexible in feature extraction and able to dig out more informative features via deep network structure. Moreover, the proposed method integrates feature extraction and abnormal condition detection into one model rather than separately accomplish them, making it more convenient and efficient.

The effectiveness of the proposed method is demonstrated through the flame images obtained from the combustion system. The experiment results show that the proposed method is superior to the other traditional image-based monitoring methods and the single DBN-based monitoring method. In the future, the process data and process images will be combined to make up a more complete dataset, which will have positive effects on process monitoring.

Acknowledgment

The authors would like to acknowledge sponsorship and financial support from the National Science Foundation of China (61833014) and the Ministry of Science and Technology, Taiwan, R.O.C. (MOST 106-2221-E-033-060-MY3).

References

[1] Z. Ge, Z. Song, Mixture Bayesian regularization method of PPCA for multimode process monitoring, AIChE J. 56 (11) (2010) 2838–2849.
[2] Y. Zhang, Z. Hu, On-line batch process monitoring using hierarchical kernel partial least squares, Chem. Eng. Res. Des. 89 (10) (2011) 2078–2084.
[3] J. Yu, Local and global principal component analysis for process monitoring, J. Process Control 22 (7) (2012) 1358–1373.
[4] Y. Hu, H. Ma, H. Shi, Enhanced batch process monitoring using just-in-time-learning based kernel partial least squares, Chemometr. Intell. Lab. Syst. 123 (2013) 15–27.
[5] S. Joe Qin, Statistical process monitoring: basics and beyond, J. Chemom. 17 (8-9) (2003) 480–502.
[6] J. Prats-Montalbán, A. Ferrer, Integration of colour and textural information in multivariate image analysis: defect detection and classification issues, J. Chemom. 21 (1-2) (2007) 10–23.
[7] H. Zhang, Z. Song, Smelting process smoke detection using multivariate image analysis, in: Intelligent Control and Automation (WCICA), 2012 10th World Congress on, IEEE, 2012.
[8] C. Yang, et al., Application of highlight removal and multivariate image analysis to color measurement of flotation bubble images, Int. J. Imaging Syst. Technol. 19 (4) (2009) 316–322.
[9] J. Prats-Montalbán, A. De Juan, A. Ferrer, Multivariate image analysis: a review with applications, Chemometr. Intell. Lab. Syst. 107 (1) (2011) 1–23.
[10] M. Bharati, J. MacGregor, W. Tropper, Softwood lumber grading through on-line multivariate image analysis techniques, Ind. Eng. Chem. Res. 42 (21) (2003) 5345–5353.
[11] H. Yu, J.F. MacGregor, Monitoring flames in an industrial boiler using multivariate image analysis, AIChE J. 50 (7) (2004) 1474–1483.
[12] J.-S. Wang, X.-D. Ren, GLCM based extraction of flame image texture features and KPCA-GLVQ recognition method for rotary kiln combustion working conditions, Int. J. Autom. Comput. 11 (1) (2014) 72–77.
[13] X. Bai, et al., Multi-mode combustion process monitoring on a pulverised fuel combustion test facility based on flame imaging and random weight network techniques, Fuel 202 (2017) 656–664.
[14] B. Walczak, Wavelets in chemistry, Technometrics 43 (2) (2013) 246.
[15] M.S. Reis, A. Bauer, Wavelet texture analysis of on-line acquired images for paper formation assessment and monitoring, Chemometr. Intell. Lab. Syst. 95 (2) (2009) 129–137.
[16] J.J. Liu, J.F. MacGregor, Estimation and monitoring of product aesthetics: application to manufacturing of "engineered stone" countertops, Mach. Vis. Appl. 16 (6) (2006) 374–383.
[17] J.J. Liu, J.F. Macgregor, On the extraction of spectral and spatial information from images, Chemometr. Intell. Lab. Syst. 85 (1) (2007) 119–130.
[18] F.L. García, Real-time Surface Grading of Ceramic Tiles, Universitat Politècnica De València, 2005.
[19] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[20] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[21] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[22] Y. Liu, S. Zhou, Q. Chen, Discriminative deep belief networks for visual data classification, Pattern Recogn. 44 (10) (2011) 2287–2296.
[23] S. Kang, X. Qian, H. Meng, Multi-distribution deep belief network for speech synthesis. in Acoustics, Speech and Signal Processing (ICASSP), in: IEEE International Conference on. 2013. IEEE, 2013.
[24] Z. Zhao, et al., Discriminant deep belief network for high-resolution SAR image classification, Pattern Recogn. 61 (2017) 686–701.
[25] X.-l. Hao, M. Tian, Deep belief network based on double Weber local descriptor in micro-expression recognition, in: Advanced Multimedia and Ubiquitous Engineering, Springer, 2017, pp. 419–425.
[26] X. Gao, et al., Refinery scheduling with varying crude: a deep belief network classification and multimodel approach, AIChE J. 60 (7) (2014) 2525–2532.
[27] Z. Zhang, J. Zhao, A deep belief network based fault diagnosis model for complex chemical processes, Comput. Chem. Eng. 107 (2017) 395–407.
[28] C. Shang, et al., Data-driven soft sensor development based on deep learning technique, J. Process Control 24 (3) (2014) 223–233.
[29] Y. Lecun, et al., A tutorial on energy-based learning, in: Predicting Structured Data, 2006.
[30] E. Lifshitz, L. Landau, Statistical Physics (Course of Theoretical Physics), Search PubMed, 1984, p. 5.
[31] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (8) (2002) 1771–1800.
[32] G.E. Hinton, A practical guide to training restricted Boltzmann machines, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 599–619.
[33] Y. Bengio, Deep learning of representations for unsupervised and transfer learning, in: Workshop on Unsupervised & Transfer Learning, 7, 2012, pp. 1–20.
[34] Y. Bengio, Learning deep architectures for AI, Found. Trends® Mach. Learn. 2 (1) (2009) 1–127.
[35] E. Ising, Beitrag zur theorie des ferromagnetismus, Z. für Physik A Hadrons Nucl. 31 (1) (1925) 253–258.
[36] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (3) (1962) 1065–1076.