Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 1320780, 13 pages
https://doi.org/10.1155/2017/1320780

Review Article

Deep Learning in Visual Computing and Signal Processing

Danfeng Xie, Lei Zhang, and Li Bai

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19121, USA

Correspondence should be addressed to Danfeng Xie; danfeng.xie@temple.edu

Received 21 October 2016; Revised 15 December 2016; Accepted 15 January 2017; Published 19 February 2017

Academic Editor: Francesco Carlo Morabito

Deep learning is a subfield of machine learning, which aims to learn a hierarchy of features from input data. Nowadays, researchers have intensively investigated deep learning algorithms for solving challenging problems in many areas such as image classification, speech recognition, signal processing, and natural language processing. In this study, we not only review typical deep learning algorithms in computer vision and signal processing but also provide detailed information on how to apply deep learning to specific areas such as road crack detection, fault diagnosis, and human activity detection. Besides, this study also discusses the challenges of designing and training deep neural networks.

1. Introduction

Deep learning methods are a group of machine learning methods that can learn features hierarchically from lower level to higher level by building a deep architecture. The deep learning methods have the ability to automatically learn features at multiple levels, which makes the system be able to learn complex mapping function $f : X \rightarrow Y$ directly from data, without help of the human-crafted features. This ability is crucial for high-level feature abstraction since high-level features are difficult to be described directly from raw training data. Moreover, with the sharp growth of data, the ability to learn high-level features automatically will be even more important.

The most characterizing feature of deep learning methods is that their models all have deep architectures. A deep architecture means it has multiple hidden layers in the network. In contrast, a shallow architecture has only few hidden layers (1 to 2 layers). Deep architectures are loosely inspired by mammal brain. When given an input percept, mammal brain processes it using different area of cortex which abstracts different levels of features. Researchers usually describe such concepts in hierarchical ways, with many levels of abstraction. Furthermore, mammal brains also seem to process information through many stages of transformation and representation. A very clear example is that the information in the primate visual system is processed in a sequence of stages: edge detection, primitive shapes, and more complex visual shapes.

Inspired by the deep architecture of mammal brain, researchers investigated deep neural networks for two decades but did not find effective training methods before 2006: researchers only obtained good experimental results of neural network with one or two hidden layers but could not get good results of neural network with more hidden layers. In 2006, Hinton et al. proposed deep belief networks (DBNs) [1], with a learning algorithm that uses unsupervised learning algorithm to greedily train deep neural network layer by layer. This training method, which is called deep learning, turns out to be very effective and efficient in training deep neural networks.

Many other deep architectures, that is, autoencoder, deep convolutional neural networks, and recurrent neural networks, are successfully applied in various areas. Regression [2], classification [3–9], dimensionality reduction [10, 11], modeling motion [12, 13], modeling textures [14], information retrieval [15–17], natural language processing [18–20], robotics [21], fault diagnosis [22], and road crack detection [23] have seen increasing deep learning-related research studies. There are mainly three crucial reasons for the rapid development of deep learning applications nowadays: the big leap of deep learning algorithms, the significantly increased computational abilities, and the sharp drop of price in hardware.

This survey provides an overview of several deep learning algorithms and their emerging applications in several specific areas, featuring face recognition, road crack detection, fault diagnosis, and falls detection. As complementarity to existing review papers [24, 25], we not only review the state-of-the-art deep learning methods but also provide detailed information on how to apply deep learning to specific problems. The reminder of this paper is organized as follows. In Section 2, the two categories of deep learning algorithms are introduced: restricted Boltzmann machines (RBMs) and convolutional neural networks (CNNs). The training strategies are discussed in Section 3. In Section 4, we describe several specific deep learning applications, that is, face recognition, road crack detection, fault diagnosis, and human activity detection. In Section 5, we discuss several challenges of training and using the deep neural networks. In Section 6, we conclude the paper.

2. Deep Learning Algorithms

Deep learning algorithms have been extensively studied in recent years. As a consequence, there are a large number of related approaches. Generally speaking, these algorithms can be grouped into two categories based on their architectures: restricted Boltzmann machines (RBMs) and convolutional neural networks (CNNs). In the following sections, we will briefly review these deep learning methods and their developments.

2.1. Deep Neural Network. This section introduces how to build and train RBM-based deep neural networks (DNNs). The building and training procedures of a DNN contain two steps. First, build a deep belief network (DBN) by stacking restricted Boltzmann machines (RBMs) and feed unlabeled data to pretrain the DBN. The pretrained DBN provides initial parameters for the deep neural network. In the second step, labeled data is fed to train the DNN using back-propagation. After two steps of training, a trained DNN is obtained. This section is organized as follows. Section 2.1.1 introduces RBM, which is the basic component of DBN. In Section 2.1.2, RBM-based DNN is introduced.

2.1.1. Restricted Boltzmann Machines. RBM is an energy-based probabilistic generative model [26–29]. It is composed of one layer of visible units and one layer of hidden units. The visible units represent the input vector of a data sample and the hidden units represent features that are abstracted from the visible units. Every visible unit is connected to every hidden unit, whereas no connection exists within the visible layer or hidden layer. Figure 1 illustrates the graphical model of restricted Boltzmann machine.

As a result of the lack of hidden-hidden and input-input interactions, the energy function of a RBM is

$$\text{Energy}(\mathbf{v}, \mathbf{h}; \theta) = -\mathbf{b}^T\mathbf{v} - \mathbf{c}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{v}, \tag{1}$$

where $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ are the parameters of RBM and they need to be learned during the training procedure; $\mathbf{W}$ denotes the weights between the visible layer and hidden layer; $\mathbf{b}$ and $\mathbf{c}$ are the bias of the visible layer and hidden layer, respectively; this model is called binary RBM because the vectors $\mathbf{v}$ and $\mathbf{h}$ only contain binary values (0 or 1).

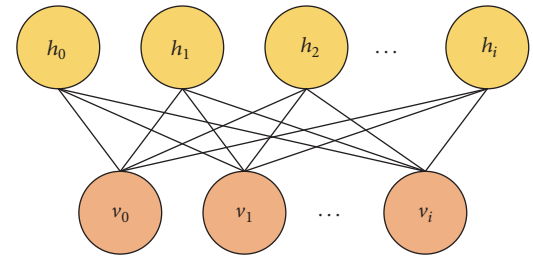


Figure 1: Restricted Boltzmann machine.

We can obtain a tractable expression for the conditional probability $P(h \mid v)$ [30]:

$$\begin{aligned} P(h \mid v) &= \frac{\exp\left(\mathbf{b}^T\mathbf{v} + \mathbf{c}^T\mathbf{h} + \mathbf{h}^T\mathbf{W}\mathbf{v}\right)}{\sum_{\tilde{\mathbf{h}}} \exp\left(\mathbf{b}^T\mathbf{v} + \mathbf{c}^T\tilde{\mathbf{h}} + \tilde{\mathbf{h}}^T\mathbf{W}\mathbf{v}\right)} \\ &= \frac{\prod_i \exp\left(\mathbf{c}_i\mathbf{h}_i + \mathbf{h}_i\mathbf{W}_i\mathbf{v}\right)}{\prod_i \sum_{\tilde{\mathbf{h}}_i} \exp\left(\mathbf{c}_i\tilde{\mathbf{h}}_i + \tilde{\mathbf{h}}_i\mathbf{W}_i\mathbf{v}\right)} \\ &= \prod_i \frac{\exp\left(\mathbf{h}_i\left(\mathbf{c}_i + \mathbf{W}_i\mathbf{v}\right)\right)}{\sum_{\tilde{\mathbf{h}}_i} \exp\left(\tilde{\mathbf{h}}_i\left(\mathbf{c}_i + \mathbf{W}_i\mathbf{v}\right)\right)} = \prod_i P\left(\mathbf{h}_i \mid v\right). \end{aligned} \tag{2}$$

For binary RBM, where $h_i \in \{0, 1\}$, the equation for a hidden unit's output given its input is

$$P(h_i = 1 \mid v) = \frac{e^{c_i + W_i v}}{1 + e^{c_i + W_i v}} = \text{sigm}\left(c_i + W_i v\right). \tag{3}$$

Because v and h play a symmetric role in the energy function, the following equation can be derived:

$$P(v \mid h) = \prod_i P\left(v_i \mid h\right), \tag{4}$$

and for the visible unit $v_j \in \{0, 1\}$, we have

$$P\left(v_j = 1 \mid h\right) = \text{sigm}\left(b_j + W_{\cdot j}^T h\right), \tag{5}$$

where $W_{\cdot j}$ is the jth column of W.

Although binary RBMs can achieve good performance when dealing with discrete inputs, they have limitations to handle continuous-valued inputs due to their structure. Thus, in order to achieve better performance on continuous-valued inputs, Gaussian RBMs are utilized for the visible layer [4, 31]. The energy function of a Gaussian RBM is

$$\text{Energy}(\mathbf{v}, \mathbf{h}) = \sum_i \frac{\left(v_i - a_i\right)^2}{2\sigma_i^2} - \sum_{ij} w_{ij} h_j \frac{v_i}{\sigma_i} - \sum_j c_j h_j, \tag{6}$$

where a_i and σ_i are the mean and the standard deviation of visible unit i. Note here that only the visible layer v is continuous-valued and hidden layer h is still binary. In practical situation, the input data is normalized, which makes $a_i = 0$ and $\sigma_i = 1$. Therefore, (6) becomes

$$\text{Energy}(\mathbf{v}, \mathbf{h}) = \frac{1}{2}\mathbf{v}^T\mathbf{v} - \mathbf{c}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{v}. \tag{7}$$

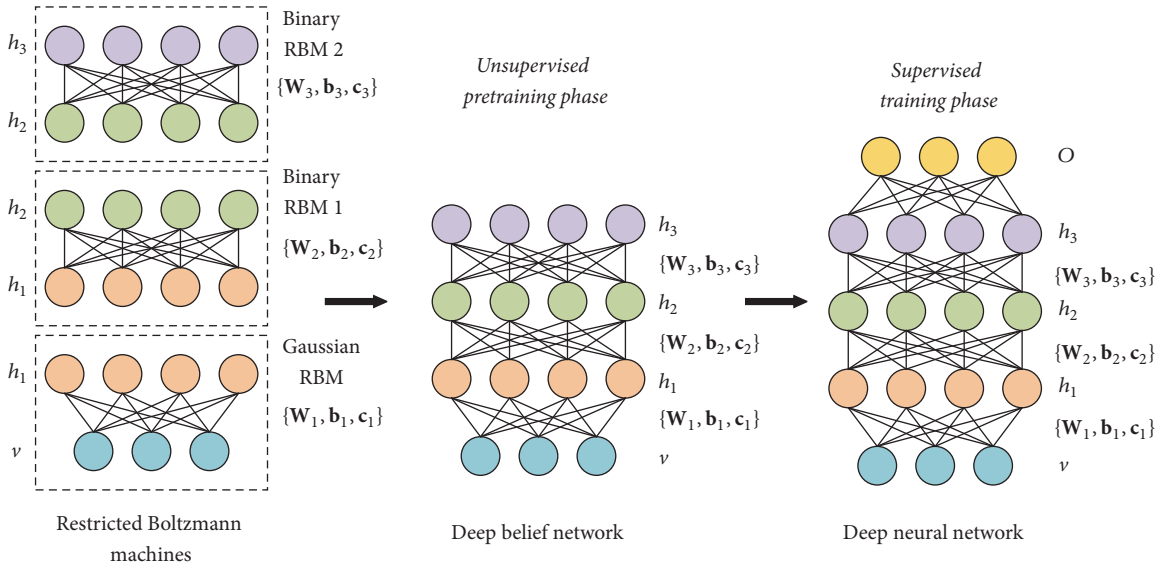


Figure 2: Deep belief network structure.

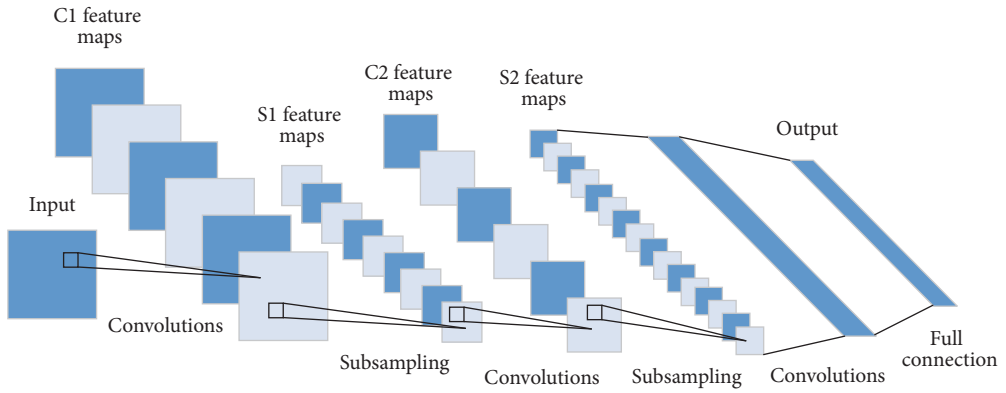


Figure 3: The architecture of convolution neural network.

2.1.2. Deep Neural Network. Hinton et al. [1] showed that RBMs can be stacked and trained in a greedy manner to form so-called deep belief networks (DBNs) [32]. DBNs are graphical models which learn to extract deep hierarchical representation of the training data. A DBN model with l layers models the joint distribution between observed vector v and ℓ hidden layers h^k as follows [30]:

$$P\left(v, h^1, \ldots, h^\ell\right) = \left(\prod_{k=0}^{\ell-2} P\left(h^k \mid h^{k+1}\right)\right) P\left(h^{\ell-1}, h^\ell\right), \quad (8)$$

where $v = h^0$, $P(h^{k-1} \mid h^k)$ is a conditional distribution for the visible units conditioned on the hidden units of the RBM at level k and $P(h^{\ell-1}, h^\ell)$ is the visible-hidden joint distribution in the top-level RBM. This is illustrated in Figure 2.

As Figure 2 shows, the hidden layer of low-level RBM is the visible layer of high-level RBM, which means that the output of low-level RBM is the input of high-level RBM. By using this structure, the high-level RBM is able to learn high-level features from low-level features generated from the low-level RBM. Thus, DBN allows latent variable space in its hidden layers. In order to train a DBN effectively, we need to train its RBM from low level to high level successively.

After the unsupervised pretraining step for DBN, the next step is to use parameters from DBN to initialize the DNN and do supervised training for DNN using back-propagation. The parameters of the N-layer DNN are initialized as follows: parameters $\{W_n, c_n\}$ $(l = 1, \ldots, N)$ except the top layer parameters are set the same as the DBN, and the top layer weights $\{W_N, c_N\}$ are initialized stochastically. After that, the whole network can be fine-tuned by back-propagation in a supervised way using labeled data.

2.2. Convolutional Neural Network. Convolutional neural network is one of the most powerful classes of deep neural networks in image processing tasks. It is highly effective and commonly used in computer vision applications [33]. The convolution neural network contains three types of layers: convolution layers, subsampling layers, and full connection layers. The whole architecture of convolutional neural network is shown in Figure 3. A brief introduction to each type of layer is provided in the following paragraphs.

3	15	64	22	55	62
92	213	7	32	145	34
17	178	86	33	12	21
231	87	48	5	23	234
59	56	55	45	3	218
82	97	94	33	238	44

1	1	1
1	0	2
1	0	1

Figure 4: Digital image representation and convolution matrix.

2.2.1. Convolution Layer. As Figure 4 shows, in convolution layer, the left matrix is the input, which is a digital image, and the right matrix is a convolution matrix. The convolution layer takes the convolution of the input image with the convolution matrix and generates the output image. Usually the convolution matrix is called filter and the output image is called filter response or filter map. An example of convolution calculation is demonstrated in Figure 5. Each time, a block of pixels is convoluted with a filter and generates a pixel in a new image.

2.2.2. Subsampling Layer. The subsampling layer is an important layer to convolutional neural network. This layer is mainly to reduce the input image size in order to give the neural network more invariance and robustness. The most used method for subsampling layer in image processing tasks is max pooling. So the subsampling layer is frequently called max pooling layer. The max pooling method is shown in Figure 6. The image is divided into blocks and the maximum value of each block is the corresponding pixel value of the output image. The reason to use subsampling layer is as follows. First, the subsampling layer has fewer parameters and it is faster to train. Second, a subsampling layer makes convolution layer tolerate translation and rotation among the input pattern.

2.2.3. Full Connection Layer. Full connection layers are similar to the traditional feed-forward neural layer. They make the neural network fed forward into vectors with a predefined length. We could fit the vector into certain categories or take it as a representation vector for further processing.

3. Training Strategy

Compared to conventional machine learning methods, the advantage of the deep learning is that it can build deep architectures to learn more multiscale abstract features. Unfortunately, the large amount of parameters of the deep architectures may lead to overfitting problem.

3.1. Data Augmentation. The key idea of data augmentation is to generate additional data without introducing extra labeling costs. In general, the data augmentation is achieved by deforming the existing ones. Mirroring, scaling, and rotation are the most common methods for data augmentation [34–36]. Wu et al. extended the deforming idea to color space, the provided color casting, vignetting, and lens distortion

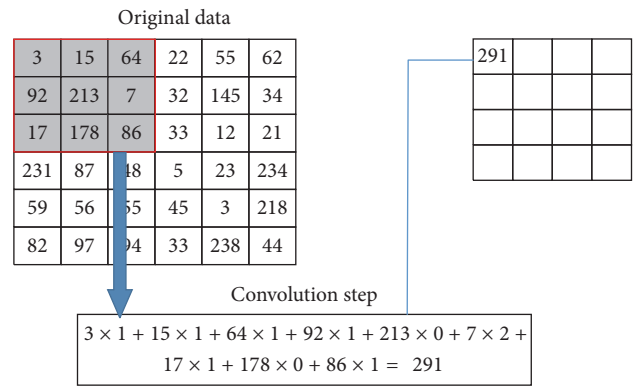


Figure 5: An example of convolution calculation.

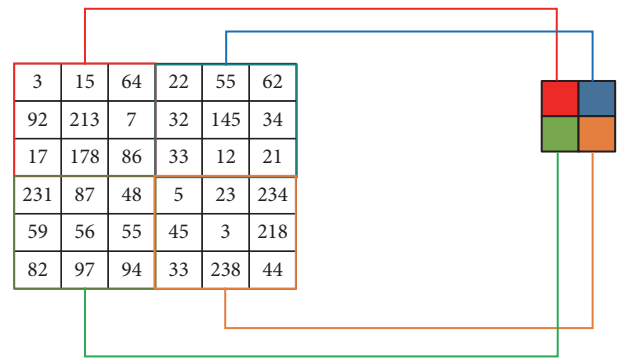


Figure 6: The example of the subsampling layer.

techniques in their work, which enlarged the training set significantly [37].

3.2. Pretraining and Fine-Tuning. Training a deep learning architecture is a time-consuming and nontrivial task. On one hand, it is difficult to obtain enough well-labeled data to train the deep learning architecture in real application, although the data augmentation can help us obtain more training data.

For visual tasks, when it is hard to get sufficient data, a recommendable way is to fine-tune the pretrained CNN by natural images (e.g., ImageNet) and then use specific data set to fine-tune the CNN [36, 38, 39]. Tajbakhsh et al. showed that, for medical applications, the use of a pretrained CNN with adequate fine-tuning outperformed or, in the worst case, performed as well as a CNN trained from scratch [38].

On the other hand, the deep learning architecture contains hundreds of thousands of parameters to be initialized even with sufficient data. Erhan et al. provided the evidence to explain that the pretraining step helps train deep architectures such as deep belief networks and stacked autoencoders [40]. Their experiments supported a regularization explanation for the effect of pretraining, which helps the deep-learned model obtain better generalization from the training data set.

4. Applications

Deep learning has been widely applied in various fields, such as computer vision [25], signal processing [24], and speech recognition [41]. In this section, we will briefly review several recently developed applications of deep learning (all the results are referred from the original papers).

4.1. CNN-Based Applications in Visual Computing. As we know, convolutional neural networks are very powerful tools for image recognition and classification. These different types of CNNs are often tested on well-known ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) data set and achieved state-of-the-art performance in recent years [42–44]. After winning the ImageNet competition in 2012 [42], the CNN-based methods have brought about a revolution in computer vision. CNNs have been applied with great success to the object detection [35, 45, 46], object segmentation [47, 48], and recognition of objects and regions in images [49–54]. Compared with hand-crafted features, for example, Local Binary Patterns (LBP) [55] and Scale Invariant Feature Transform (SIFT) [56], which need additional classifiers to solve vision problems [57–59], the CNNs can learn the features and the classifiers jointly and provide superior performance. In next subsection, we review how the deep-learned CNN is applied to recent face recognition and road crack detection problem in order to provide an overview for applying the CNN to specific problems.

4.1.1. CNN for Face Recognition. Face recognition has been one of the most important computer vision tasks since the 1970s [60]. Face recognition systems typically consist of four steps. First, given an input image with one or more faces, a face detector locates and isolates faces. Then, each face is preprocessed and aligned using either 2D or 3D modeling methods. Next, a feature extractor extracts features from an aligned face to obtain a low-dimensional representation (or embedding). Finally, a classifier makes predictions based on the low-dimensional representation. The key to get good performances for face recognition systems is obtaining an effective low-dimensional representation. Face recognition systems using hand-crafted features include [61–64]. Lawrence et al. [65] first proposed using CNNs for face recognition. Currently, the state-of-the-art performance of face recognition systems, that is, Facebook's DeepFace [66] and Google's FaceNet [67], are based on CNNs. Other notable CNN-based face recognition systems are lightened convolutional neural networks [68] and Visual Geometry Group (VGG) Face Descriptor [69].

Figure 7 shows the logic flow of CNN-based face recognition systems. Instead of using hand-crafted features, CNNs are directly applied to RGB pixel values and used as a feature extractor to provide a low-dimensional representation characterizing a person's face. In order to normalize the input image to make the face robust to different view angles, DeepFace [66] models a face in 3D and aligns it to appear as a frontal face. Then, the normalized input is fed to a single convolution-pooling-convolution filter. Next, 3 locally connected layers and 2 fully connected layers are used to make final predictions. The architecture of DeepFace is shown in Figure 8. Though DeepFace achieves the best performance on face recognition up to date, its representation is difficult to interpret and use because the faces of the same person are not clustered necessarily during the training process. In contrast, FaceNet defines a triplet loss function directly on the representation, which makes the training procedure learn to cluster face representation of the same person [70]. It should also be noted that OpenFace uses a simple 2D affine transformation to align face input.

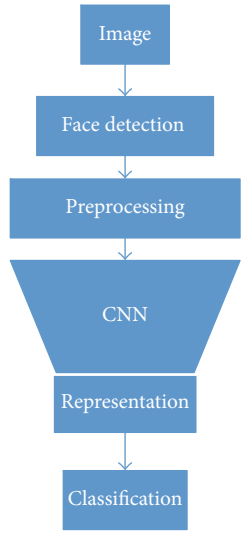


Figure 7: Logic flow of CNN-based face recognition [70].

Table 1: Experiment results on LFW benchmark [70].

Technique	Accuracy
Human-level (cropped) [74]	0.9753
FaceNet [67]	0.9964 ± 0.009
DeepFace-ensemble [66]	0.9735 ± 0.0025
OpenFace [70]	0.9292 ± 0.0134

Nowadays, face recognition in mobile computing is a very attractive topic [71, 72]. While DeepFace and FaceNet remain private and are of large size, OpenFace [70] offers a light-weighted, real-time, and open-source face recognition system with competitive accuracy, which is suitable for mobile computing. OpenFace implements FaceNet's architecture but it is one order of magnitude smaller than DeepFace and two orders of magnitude smaller than FaceNet. Their performances are compared on Labeled Faces in the Wild data set (LFW) [73], which is a standard benchmark in face recognition. The experiment results are demonstrated in Table 1. Though the accuracy of OpenFace is slightly lower than the state of the art, its smaller size and fast execution time show great potential in mobile face recognition scenarios.

4.1.2. CNN for Road Crack Detection. Automatic detection of pavement cracks is an important task in transportation maintenance for driving safety assurance. Inspired by recent success in applying deep learning to computer vision and

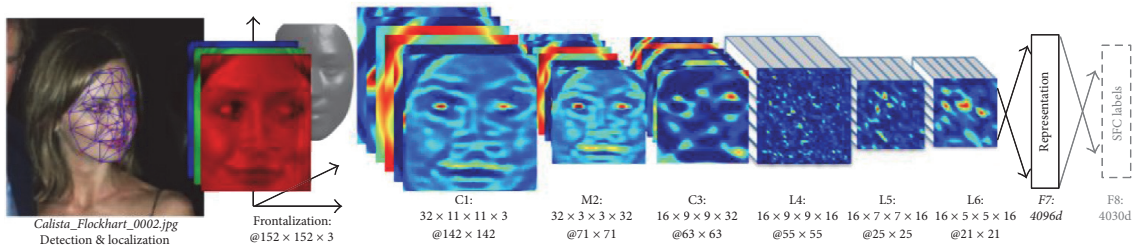


Figure 8: Outline of DeepFace architecture [66].

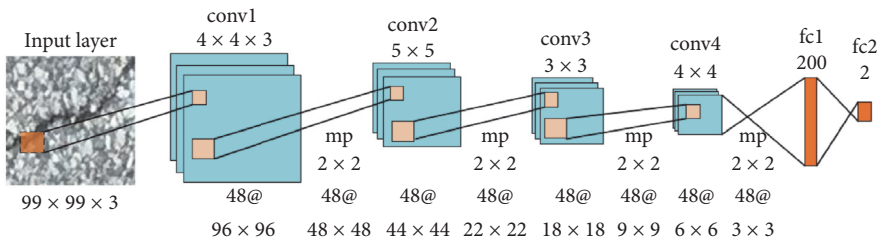


Figure 9: Illustration of the architecture of the proposed ConvNet [23].

medical problems, a deep learning based method for crack detection is proposed [23].

Data Preparation. A data set with more than 500 pavement pictures of size 3264 × 2448 is collected at the Temple University campus by using a smartphone as the data sensor. Each image is annotated by multiple annotators. Patches of size 99 × 99 are used for training and testing the proposed method. 640,000 patches, 160,000 patches, and 200,000 patches are selected as training set, validation set, and testing set, respectively.

Design and Train the CNN. A deep learning architecture is designed, which is illustrated in Figure 9 and *conv*, *mp*, and *fc* represent convolutional, max pooling, and fully connected layers, respectively. The CNNs are trained using the stochastic gradient descent (SGD) method on GPU with a batch size of 48 examples, momentum of 0.9, and weight decay of 0.0005. Less than 20 epochs are needed to reach a minimum on the validation set. The dropout method is used between two fully connected layers with a probability of 0.5 and the rectified linear units (ReLU) as the activation function.

Evaluate the Performance of the CNN. The proposed method is compared against the support vector machine (SVM) and the Boosting methods. The features for training the SVM and the Boosting method are based on color and texture of each patch which are associated with a binary label indicating the presence or absence of cracked pavement. The feature vector is 93-dimensional and is composed of color elements, histograms of textons, and LBP descriptor within the patch.

The Receiver Operating Characteristic (ROC) curves of the proposed method, the SVM, and the Boosting method are shown in Figure 10. Both the ROC curve and Area under the Curve (AUC) of the proposed method indicate that the proposed deep learning based method can outperform

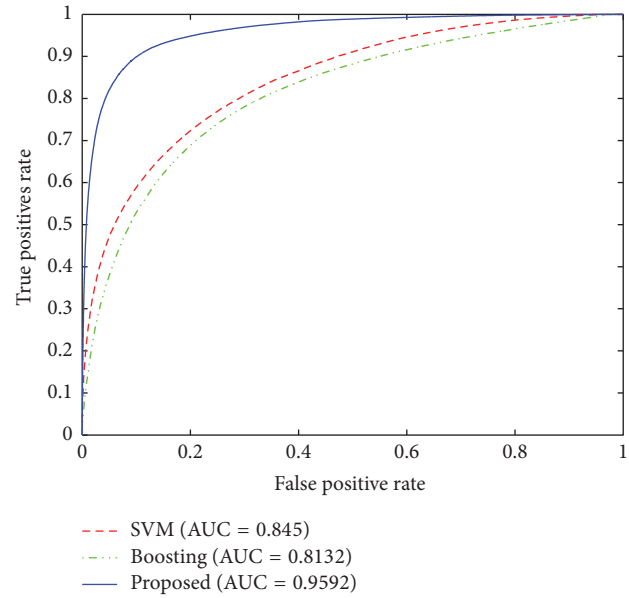
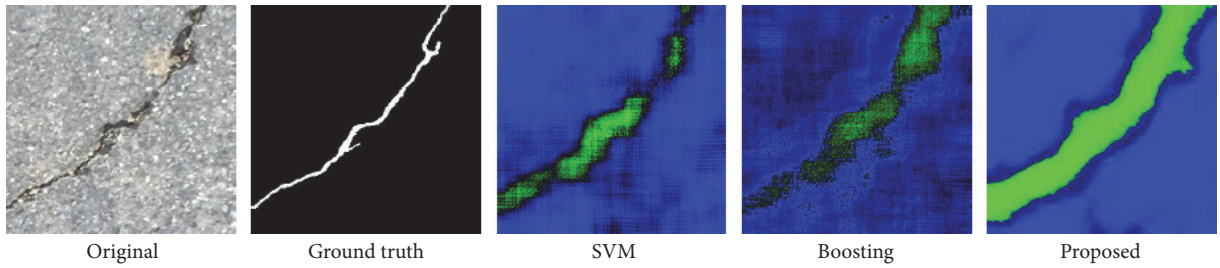


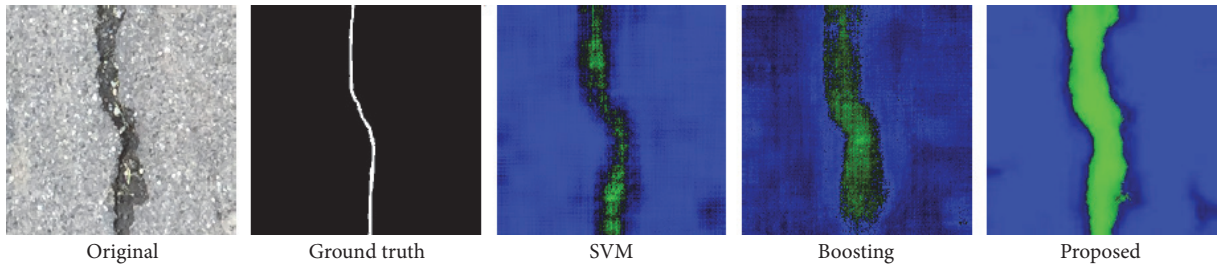
Figure 10: ROC curves [23].

the shallow structure learned from hand-crafted features. In addition, more comprehensive experiments are conducted on 300 × 300 scenes as shown in Figure 11.

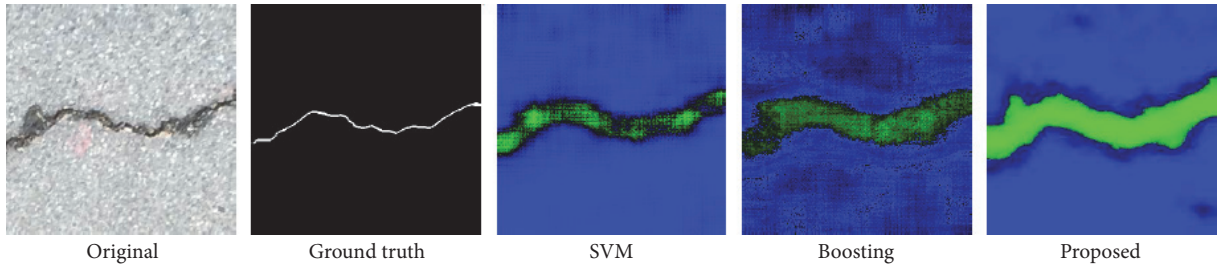
For each scene, each row shows the original image with crack, ground truth, and probability maps generated by the SVM and the Boosting methods and that by the ConvNet. The pixels in green and in blue denote the crack and the noncrack, respectively, and higher brightness means higher confidence. The SVM cannot distinguish the crack from the background, and some of the cracks have been misclassified. Compared to the SVM, the Boosting method can detect the cracks with a higher accuracy. However, some of the background



(a) Scene 1



(b) Scene 2



(c) Scene 3

Figure 11: Probability maps.

patches are classified as cracks, resulting in isolated green parts in Figure 11. In contrast to these two methods, the proposed method provides superior performance in correctly classifying crack patches from background ones.

4.2. DBN-Based Applications in Signal Processing

4.2.1. DNN for Fault Diagnosis. Plant faults may cause abnormal operations, emergency shutdowns, equipment damage, or even casualties. With the increasing complexity of modern plants, it is difficult even for experienced operators to diagnose faults fast and accurately. Thus, designing an intelligent fault detection and diagnose system to aid human operators is a critical task in process engineering. Data-driven methods for fault diagnosis are becoming very popular in recent years, since they utilize powerful machine learning algorithms. Conventional supervised learning algorithms used for fault diagnosis are Artificial Neural Networks [76–81] and support vector machines [82–84]. As one of emerging machine learning techniques, deep learning techniques are investigated for fault diagnosis in a few current studies [22, 85–88]. This subsection reviews a study which uses Hierarchical Deep Neural Network (HDNN) [22] to diagnose faults in a well-known data set called Tennessee Eastman Process (TEP).

TEP is a simulation model that simulates a real industry process. The model was first created by Eastman Chemical Company [75]. It consists of five units: a condenser, a compressor, a reactor, a separator, and a stripper. Two liquid products G and H are produced from the process with the gaseous inputs A, C, D, and E and the inert component B. The flowsheet of TEP is shown in Figure 12.

Data Preparation. The TEP is monitored by a network of M sensors that collect measurement at the same sampling time. At the ith sample, the state of mth sensor is represented by a scalar x_i^m. By combining all M sensors, the state of the whole process in ith sampling interval is represented as a row vector $x_i = [x_i^1, x_i^2, \ldots, x_i^M]$. The fault occurring at the ith sampling interval is indicated with class label $y_i \in \{1, 2, \ldots, C\}$, where value 1 to C represents one of C fault types. There are total N historical observations collected from all M sensors to form a data set $D = \{(x_i, y_i), i = 1, 2, \ldots, N, y_i \in \{1, 2, \ldots, C\}\}$. The objective of fault diagnosis is to train a classification $h : x_i \rightarrow y_i$ given data set $D = \{(x_i, y_i), i = 1, 2, \ldots, N\}$.

For each simulation run, the simulation starts without faults and the faults are introduced at sample 1. Each run collects a total of 1000 pieces of sample data. Each single fault type has 5 independent simulation runs. The Tennessee Eastman Process has 20 different predefined faults but faults

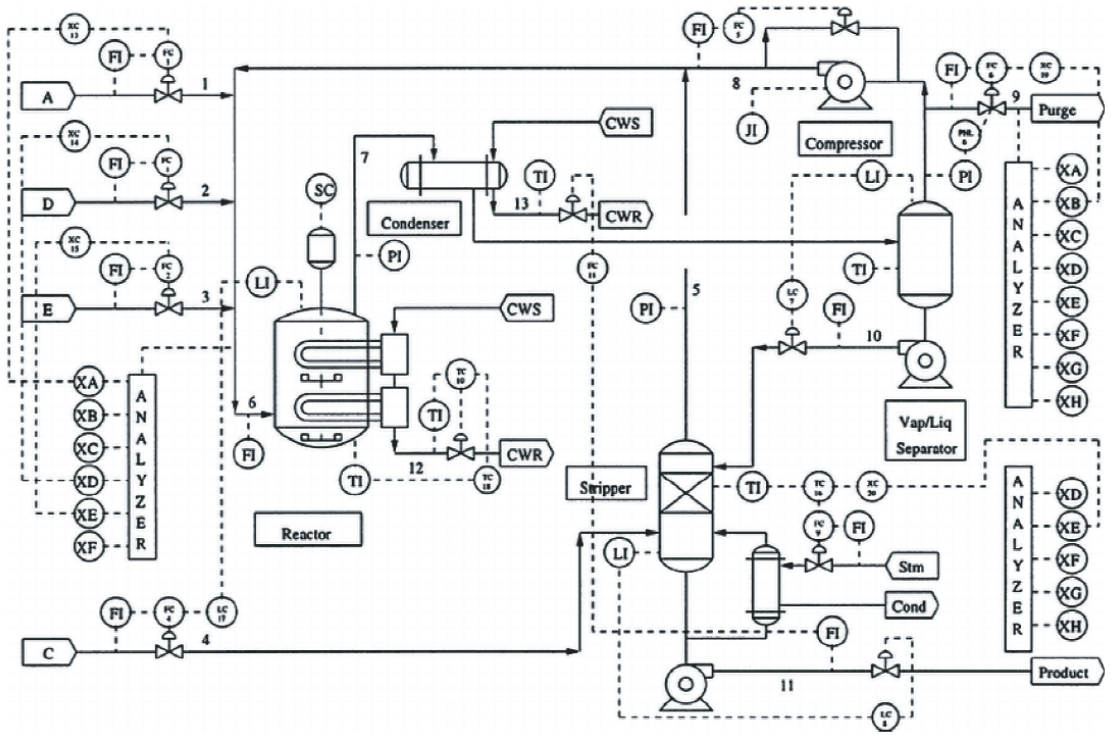


Figure 12: Tennessee Eastman Process [75].

3, 9, and 15 are excluded for fault diagnosis due to no effect or subtle effect on all the sensors [82, 84]. Thus, the training data set has a total of $N = 5 \times 17 \times 1000 = 85000$ data samples; that is, $D_{\text{train}} = \{(x_i, y_i), i = 1, 2, \ldots, N, y_i \in \{0, 1, \ldots, C\}\}$, $N = 85000$, $C = 17$. Then, test data is generated using the same method. Because only fault diagnosis methods are investigated in this work, normal operation data is not considered. Data normalization and data augmentation techniques are used to achieve better performance.

Design and Train the HDNN. The general diagnosis scheme of HDNN [22] is as follows. The symptom data generated by simulation is transmitted to a supervisory DNN. The supervisory DNN then classifies symptom data into different groups and triggers the DNN which is specially trained for that group to do further fault diagnosis. Figure 13 illustrates the fault diagnosis scheme of the HDNN, where each agent represents a DNN.

Evaluate the Performance of the DNN. The experiment result of the HDNN is compared to single neural network and Duty-Oriented Hierarchical Artificial Neural Network (DOHANN) [76] and is shown in Figure 14. 7 out of 17 faults have been diagnosed with 90% accuracy. The highest Correct Classification Rate (CCR) is 99.6% from fault 4, while the lowest CCR is 50.4% from fault 13. The average CCR of our method is 80.5%, while the average of CCRs of SNN and DOHANN is 49.7% and 70.7%, respectively. It demonstrates that the DNN-based algorithm outperforms other conventional NN-based algorithms.

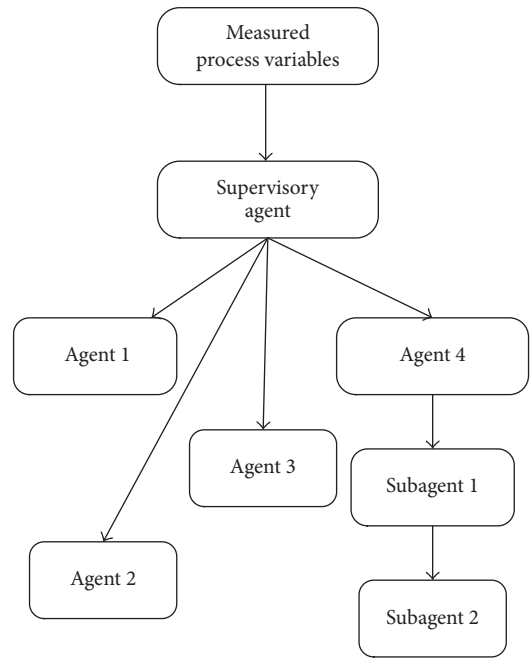


Figure 13: Schematic diagram of HDNN [22].

4.2.2. DNN for Human Activity Detection. Human activity detection has drawn much attention from researchers due to high demands for security, law enforcement, and health care [90–93]. In contrast to using cameras to detect human

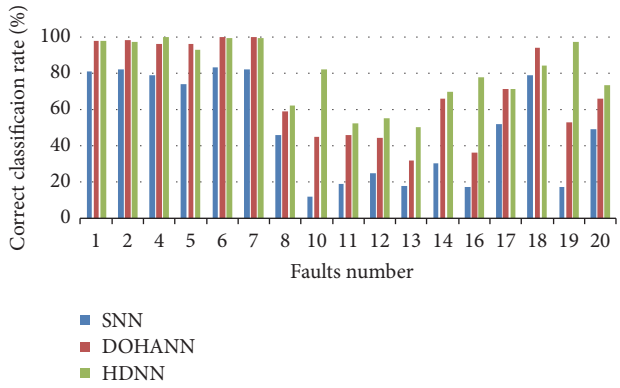


Figure 14: Correct classification rate of SNN, DOHANN [76], and HDNN [22].

activity, sensors such as worn accelerometers or in-home radar which use signals to detect human activities are robust to environmental conditions such as weather conditions and light variations [94–99]. Nowadays, there are a few emerging research works that focus on using deep learning technologies to detect human activities based on signals [89, 92, 100].

Fall detection is one of the very important human activity detection scenarios for researchers, since falls are a main cause of both fatal and nonfatal injuries for the elderly. Khan and Taati [100] proposed a deep learning method for falls detection based on signals collected from wearable devices. They propose an ensemble of autoencoders to extract features from each channel of sensing data. Unlike wearable devices which are intrusive and easily broken and must be carried, in-home radars which are safe, nonintrusive, and robust to lighting conditions show their advantages for fall detection. Jokanovic et al. [89] proposed a method that uses deep learning to detect fall motion through in-home radar. The procedure is demonstrated in Figure 15. They first denoise and normalize the spectrogram as input. Then, stacked autoencoders are performed as a feature extractor. On top of the stacked autoencoders, a softmax regression classifier is used to make predictions. The whole model is compared with a SVM model. Experiment results show that the overall correct classification rate for deep learning approach is 87%, whereas the overall correct classification rate for SVM is 78%.

5. Challenges

Though deep learning techniques achieve promising performance on multiple fields, there are still several big challenges as research articles indicate. These challenges are described as follows.

5.1. Training with Limited Data. Training deep neural network usually needs large amounts of data as larger training data set can prevent deep learning model from overfitting. Limited training data may severely affect the learning ability of a deep neural network. Unfortunately, there are many applications that lack sufficient labeled data to train a DNN. Thus, how to train DNN with limited data effectively and efficiently becomes a hot topic.

Recently, two possible solutions draw attention from researchers. One of the solutions is to generalize new training data from original training data using multiple data augmentation methods. Traditional ones include rotation, scaling, and cropping. In addition to these, Wu et al. [37] adopted vignetting, color casting, and lens distortion techniques. These techniques can further produce more different training examples. Another solution is to obtain more training data using weak learning algorithms. Song et al. [101] proposed a weakly supervised method that can label image-level object-presence. This method helps to reduce laborious bounding box annotation costs while generating training data.

5.2. Time Complexity. Training deep neural network is very time-consuming in early years. It needs a large amount of computational resources and is not suitable for real-time applications. By default, GPUs are used to accelerate training of large DNNs with the help of parallel computing technique. Thus, it is important to make the most of GPU computing ability when training DNNs. He and Sun [102] investigated training CNN under time cost constrains and proposed fast training methods for real-world applications while having similar performance as existing CNN models. Li et al. [103] remove all the redundant computations during training CNNs for pixel wise classification, which leads to a speedup of 1500 times.

5.3. Theoretical Understanding. Though deep learning algorithms achieve promising results on many tasks, the underlying theory is still not very clear. There are many questions that need to be answered. For instance, which architecture is better than other architectures in certain task? How many layers and how many nodes in each layer should be chosen in a DNN? Besides, there are a few hyperparameters such as learning rate, dropout rate, and the strength of regularizer which need to be tuned with specific knowledge.

Several approaches are developed to help researchers to get better understanding in DNN. Zeiler and Fergus [43] proposed a visualization method that illustrates features in intermediate layers. It displays intermediate features in interpretable patterns, which may help design better architectures for future DNNs. In addition to visualizing features, Girshick et al. [49] tried to discover the learning pattern of CNN by testing the performance layer by layer during the training process. It demonstrates that convolutional layers can learn more generalized features.

Although there is progress in understanding the theory of deep learning, there is still large room to improve in deep learning theory aspect.

6. Conclusion

This paper gives an overview of deep learning algorithms and their applications. Several classic deep learning algorithms such as restricted Boltzmann machines, deep belief networks, and convolutional neural networks are introduced. In addition to deep learning algorithms, their applications are

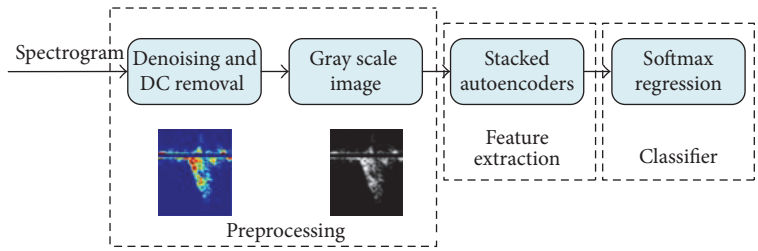


Figure 15: Block diagram of the deep learning based fall detector [89].

reviewed and compared with other machine learning methods. Though deep neural networks achieve good performance on many tasks, they still have many properties that need to be investigated and justified. We discussed these challenges and pointed out several new trends in understanding and developing deep neural networks.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[2] R. Salakhutdinov and G. E. Hinton, "Using deep belief nets to learn covariance kernels for Gaussian processes," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS '07)*, Vcancouver, Canada, December 2007.

[3] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. Xing, "Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks," in *Proceedings of the in European Conference on Computer Vision*, pp. 69–82, Springer, Marseille, France, October 2008.

[4] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19 (NIPS '06)*, pp. 153–160, MIT Press, 2007.

[5] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 473–480, ACM, Corvalis, Ore, USA, June 2007.

[6] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pp. 609–616, ACM, Quebec, Canada, June 2009.

[7] M. Ranzato, Y.-L. Boureau, and Y. LeCun, "Sparse feature learning for deep belief networks," in *Advances in Neural Information Processing Systems*, pp. 1185–1192, 2008.

[8] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS '06)*, pp. 1137–1144, Vancouver, Canada, December 2006.

[9] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, ACM, Helsinki, Finland, July 2008.

[10] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[11] R. Salakhutdinov and G. E. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Proceedings of the 8th International Conference on Artificial Intelligence and Statistics (AISTATS '07)*, pp. 412–419, San Juan, Puerto Rico, March 2007.

[12] G. W. Taylor and G. E. Hinton, "Factored conditional restricted Boltzmann machines for modeling motion style," in *Proceedings of the 26th Annual International Conference on Machine Learning(ICML '09)*, pp. 1025–1032, ACM, Quebec, Canada, June 2009.

[13] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Advances in Neural Information Processing Systems*, pp. 1345–1352, 2006.

[14] S. Osindero and G. E. Hinton, "Modeling image patches with a directed hierarchy of Markov random fields," in *Advances in Neural Information Processing Systems*, pp. 1121–1128, 2008.

[15] M. Ranzato and M. Szummer, "Semi-supervised learning of compact document representations with deep networks," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 792–799, ACM, Helsinki, Finland, July 2008.

[16] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

[17] P. E. Utgoff and D. J. Stracuzzi, "Many-layered learning," *Neural Computation*, vol. 14, no. 10, pp. 2497–2529, 2002.

[18] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, ACM, Helsinki, Finland, July 2008.

[19] A. Mnih and G. Hinton, "A scalable hierarchical distributed language model," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS '08)*, pp. 1081–1088, British Columbia, Canada, December 2008.

[20] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*, pp. 639–655, Springer, Berlin, Germany, 2012.

[21] R. Hadsell, A. Erkan, P. Sermanet, M. Scoffier, U. Muller, and Y. LeCun, "Deep belief net learning in a long-range vision system for autonomous off-road driving," in *Proceedings of the*

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08), pp. 628–633, Nice, France, September 2008.

[22] D. Xie and L. Bai, "A hierarchical deep neural network for fault diagnosis on Tennessee-Eastman process," in *Proceedings of the IEEE 14th International Conference on Machine Learning and Applications (ICMLA '15)*, pp. 745–748, IEEE, Miami, Fla, USA, December 2015.

[23] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '16)*, pp. 3708–3712, Phoenix, Ariz, USA, September 2016.

[24] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.

[25] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: a review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[26] P. Smolensky, "Information processing in dynamical systems: foundations of harmony theory," Tech. Rep. DTIC Document, 1986.

[27] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.

[28] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," in *Predicting Structured Data*, vol. 1, MIT Press, 2006.

[29] Y. LeCun and F. J. Huang, "Loss functions for discriminative training of energy-based models," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS '05)*, January 2005.

[30] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–27, 2009.

[31] M. Welling, M. Rosen-Zvi, and G. E. Hinton, "Exponential family harmoniums with an application to information retrieval," in *Advances in Neural Information Processing Systems*, pp. 1481–1488, 2004.

[32] G. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, article no. 5947, 2009.

[33] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, p. 1995, 1995.

[34] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 2843–2851, December 2012.

[35] H. R. Roth, L. Lu, J. Liu et al., "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1170–1181, 2016.

[36] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1395–1403, IEEE, Santiago, Chile, December 2015.

[37] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: scaling up image recognition," https://arxiv.org/abs/1501.02876.

[38] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu et al., "Convolutional neural networks for medical image analysis: full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[39] H.-C. Shin, H. R. Roth, M. Gao et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[40] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[41] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 4277–4280, IEEE, Kyoto, Japan, March 2012.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, December 2012.

[43] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I*, vol. 8689 of *Lecture Notes in Computer Science*, pp. 818–833, Springer, 2014.

[44] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: integrated recognition, localization and detection using convolutional networks," https://arxiv.org/abs/1312.6229.

[45] X. Wang, L. Zhang, L. Lin, Z. Liang, and W. Zuo, "Deep joint task learning for generic object extraction," in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS '14)*, pp. 523–531, ACM, Montreal, Canada, December 2014.

[46] J. Liu, N. Lay, Z. Wei et al., "Colitis detection on abdominal CT scans by rich feature hierarchies," in *Medical Imaging 2016: Computer-Aided Diagnosis*, vol. 9785 of *Proceedings of SPIE*, San Diego, Calif, USA, February 2016.

[47] G. Luo, R. An, K. Wang, S. Dong, and H. Zhang, "A deep learning network for right ventricle segmentation in short-axis mri," in *Proceedings of the Computing in Cardiology Conference (CinC '16)*, pp. 224–227, Vancouver, Canada, September 2016.

[48] H. R. Roth, L. Lu, A. Farag, A. Sohn, and R. M. Summers, "Spatial aggregation of holistically-nested networks for automated pancreas segmentation," https://arxiv.org/abs/1606.07830.

[49] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, IEEE, Columbus, Ohio, USA, June 2014.

[50] R. Girshick, "Fast R-CNN," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1440–1448, December 2015.

[51] J. Liu, C. Gao, D. Meng, and W. Zuo, "Two-stream contextualized CNN for fine-grained image classification," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 4232–4233, Phoenix, Ariz, USA, February 2016.

[52] K. Wang, L. Lin, W. Zuo, S. Gu, and L. Zhang, "Dictionary pair classifier driven convolutional neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 2138–2146, Las Vegas, Nev, USA, June 2016.

[53] L. Lin, K. Wang, W. Zuo, M. Wang, J. Luo, and L. Zhang, "A deep structured model with radius-margin bound for 3D human

activity recognition," *International Journal of Computer Vision*, vol. 118, no. 2, pp. 256–273, 2016.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1026–1034, IEEE, Santiago, Chile, December 2015.

[55] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[56] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[57] W. Lu, M. Li, and L. Zhang, "Palm vein recognition using directional features derived from local binary patterns," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 5, pp. 87–98, 2016.

[58] D. Xie, Z. Huang, S. Wang, and H. Liu, "Moving objects segmentation from compressed surveillance video based on motion estimation," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, pp. 3132–3135, IEEE, Tsukuba, Japan, November 2012.

[59] S. Wang, H. Liu, D. Xie, and B. Zeng, "A novel scheme to code object flags for video synopsis," in *Proceedings of the IEEE Visual Communications and Image Processing (VCIP '12)*, pp. 1–5, November 2012.

[60] T. Kanade, *Picture processing system by computer complex and recognition of human faces [Ph.D. thesis]*, Kyoto University, 3952, 1973.

[61] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3025–3032, June 2013.

[62] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A practical transfer learning algorithm for face verification," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 3208–3215, December 2013.

[63] T. Berg and P. N. Belhumeur, "Tom-vs-Pete classifiers and identity-preserving alignment for face verification," in *Proceedings of the 23rd British Machine Vision Conference (BMVC '12)*, BMVA Press, September 2012.

[64] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: a joint formulation," in *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part III*, vol. 7574 of *Lecture Notes in Computer Science*, pp. 566–579, Springer, Berlin, Germany, 2012.

[65] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[66] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: closing the gap to human-level performance in face verification," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1701–1708, June 2014.

[67] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 815–823, IEEE, Boston, Mass, USA, June 2015.

[68] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," https://arxiv.org/abs/1511.02683.

[69] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference*, vol. 1, p. 6, 2015.

[70] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: a general-purpose face recognition library with mobile applications," Tech. Rep. CMU-CS-16-118, CMU School of Computer Science, 2016.

[71] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proceedings of the 17th IEEE Symposium on Computers and Communication (ISCC '12)*, pp. 59–66, July 2012.

[72] H.-J. Hsu and K.-T. Chen, "Face recognition on drones: issues and limitations," in *Proceedings of the 1st Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use (DroNet '15)*, pp. 39–44, ACM, Florence, Italy, 2015.

[73] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, Mass, USA, 2007.

[74] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 365–372, IEEE, Kyoto, Japan, October 2009.

[75] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993.

[76] R. Eslamloueyan, "Designing a hierarchical neural network based on fuzzy clustering for fault diagnosis of the Tennessee-Eastman process," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 1407–1415, 2011.

[77] V. Venkatasubramanian and K. Chan, "A neural network methodology for process fault diagnosis," *AIChE Journal*, vol. 35, no. 12, pp. 1993–2002, 1989.

[78] K. Watanabe, I. Matsuura, M. Abe, M. Kubota, and D. M. Himmelblau, "Incipient fault diagnosis of chemical processes via artificial neural networks," *AIChE Journal*, vol. 35, no. 11, pp. 1803–1812, 1989.

[79] J. Y. Fan, M. Nikolaou, and R. E. White, "An approach to fault diagnosis of chemical processes via neural networks," *AIChE Journal*, vol. 39, no. 1, pp. 82–88, 1993.

[80] K. Watanabe, S. Hirota, L. Hou, and D. M. Himmelblau, "Diagnosis of multiple simultaneous fault via hierarchical artificial neural networks," *AIChE Journal*, vol. 40, no. 5, pp. 839–848, 1994.

[81] R. Eslamloueyan, M. Shahrokhi, and R. Bozorgmehri, "Multiple simultaneous fault diagnosis via hierarchical and single artificial neural networks," *Scientia Iranica*, vol. 10, no. 3, pp. 300–310, 2003.

[82] L. H. Chiang, M. E. Kotanchek, and A. K. Kordon, "Fault diagnosis based on Fisher discriminant analysis and support vector machines," *Computers and Chemical Engineering*, vol. 28, no. 8, pp. 1389–1401, 2004.

[83] M. Ge, R. Du, G. Zhang, and Y. Xu, "Fault diagnosis using support vector machine with an application in sheet metal stamping operations," *Mechanical Systems and Signal Processing*, vol. 18, no. 1, pp. 143–159, 2004.

[84] M. Grbovic, W. Li, P. Xu, A. K. Usadi, L. Song, and S. Vucetic, "Decentralized fault detection and diagnosis via sparse PCA

based decomposition and maximum entropy decision fusion," *Journal of Process Control*, vol. 22, no. 4, pp. 738–750, 2012.

[85] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, 2016.

[86] M. Gan, C. Wang, and C. Zhu, "Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 92–104, 2016.

[87] P. Jiang, Z. Hu, J. Liu, S. Yu, and F. Wu, "Fault diagnosis based on chemical sensor data with an active deep neural network," *Sensors*, vol. 16, no. 10, p. 1695, 2016.

[88] H. J. Steinhauer, A. Karlsson, G. Mathiason, and T. Helldin, "Root-cause localization using restricted Boltzmann machines," in *Proceedings of the 19th International Conference on Information Fusion (FUSION '16)*, pp. 248–255, ISIF, 2016.

[89] B. Jokanovic, M. Amin, and F. Ahmad, "Radar fall motion detection using deep learning," in *Proceedings of the IEEE Radar Conference (RadarConf '16)*, IEEE, Philadelphia, Pa, USA, May 2016.

[90] L. M. Frazier, "MDR for law enforcement," *IEEE Potentials*, vol. 16, no. 5, pp. 23–26, 1997.

[91] E. F. Greneker, "Radar flashlight for through the wall detection of humans," in *SPIE Proceedings of the Targets and Backgrounds: Characterization and Representation IV*, pp. 280–285, SPIE, Orlando, Fla, USA, April 1998.

[92] J. Park, R. Javier, T. Moon, and Y. Kim, "Micro-doppler based classification of human aquatic activities via transfer learning of convolutional neural networks," *Sensors*, vol. 16, no. 12, p. 1990, 2016.

[93] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RGBD images," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 842–849, St Paul, Minn, USA, May 2012.

[94] J. R. Smith, K. P. Fishkin, B. Jiang et al., "RFID-based techniques for human-activity detection," *Communications of the ACM*, vol. 48, no. 9, pp. 39–44, 2005.

[95] P. Van Dorp and F. C. A. Groen, "Human walking estimation with radar," *IEE Proceedings: Radar, Sonar and Navigation*, vol. 150, no. 5, pp. 356–366, 2003.

[96] R. J. Javier and Y. Kim, "Application of linear predictive coding for human activity classification based on micro-doppler signatures," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1831–1834, 2014.

[97] Y. Kim and H. Ling, "Human activity classification based on micro-doppler signatures using a support vector machine," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1328–1337, 2009.

[98] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering Online*, vol. 12, no. 1, article 66, 2013.

[99] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 579–590, 2013.

[100] S. S. Khan and B. Taati, "Detecting unseen falls from wearable devices using channel-wise ensemble of autoencoders," https://arxiv.org/abs/1610.03761.

[101] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell, "Weakly-supervised discovery of visual pattern configurations," in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS '14)*, pp. 1637–1645, Québec, Canada, December 2014.

[102] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 5353–5360, Boston, Mass, USA, June 2015.

[103] H. Li, R. Zhao, and X. Wang, "Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification," https://arxiv.org/abs/1412.4526.