Deep Learning in Medical Image Analysis

Dinggang Shen,[1,2] Guorong Wu,[1] and Heung-Il Suk[2]

[1]Department of Radiology, University of North Carolina, Chapel Hill, North Carolina 27599; email: dgshen@med.unc.edu

[2]Department of Brain and Cognitive Engineering, Korea University, Seoul 02841, Republic of Korea; email: hisuk@korea.ac.kr

Annu. Rev. Biomed. Eng. 2017. 19:221–48

First published as a Review in Advance on March 9, 2017

The *Annual Review of Biomedical Engineering* is online at bioeng.annualreviews.org

https://doi.org/10.1146/annurev-bioeng-071516-044442

Keywords

medical image analysis, deep learning, unsupervised feature learning

Abstract

This review covers computer-assisted analysis of images in the field of medical imaging. Recent advances in machine learning, especially with regard to deep learning, are helping to identify, classify, and quantify patterns in medical images. At the core of these advances is the ability to exploit hierarchical feature representations learned solely from data, instead of features designed by hand according to domain-specific knowledge. Deep learning is rapidly becoming the state of the art, leading to enhanced performance in various medical applications. We introduce the fundamentals of deep learning methods and review their successes in image registration, detection of anatomical and cellular structures, tissue segmentation, computer-aided disease diagnosis and prognosis, and so on. We conclude by discussing research issues and suggesting future directions for further improvement.

Contents

1. INTRODUCTION

Over the past few decades, medical imaging techniques, such as computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), mammography, ultrasound, and X-ray, have been used for the early detection, diagnosis, and treatment of diseases (1). In the clinic, medical image interpretation has been performed mostly by human experts such as radiologists and physicians. However, given wide variations in pathology and the potential fatigue of human experts, researchers and doctors have begun to benefit from computer-assisted interventions. Although the rate of progress in computational medical image analysis has not been as rapid as that in medical imaging technologies, the situation is improving with the introduction of machine learning techniques.

In applying machine learning, finding or learning informative features that well describe the regularities or patterns inherent in data plays a pivotal role in various tasks in medical image analysis. Conventionally, meaningful or task-related features were designed mostly by human experts on the basis of their knowledge about the target domains, making it challenging for nonexperts to exploit machine learning techniques for their own studies. In the meantime, there have been efforts to learn sparse representations based on predefined dictionaries, possibly learned from training samples. Sparse representation is motivated by the principle of parsimony in many areas of science; that is, the simplest explanation of a given observation should be preferred over more complicated ones. Sparsity-inducing penalization and dictionary learning have demonstrated the validity of this approach for feature representation and feature selection in medical image analysis (2–6). It should be noted that sparse representation or dictionary learning methods described in the literature still find informative patterns or regularities inherent in data with a shallow architecture, thus limiting their representational power. However, deep learning (7) has overcome this obstacle by incorporating the feature engineering step into a learning step. That is, instead of extracting features manually, deep learning requires only a set of data with minor preprocessing, if necessary, and then discovers the informative representations in a self-taught manner (8, 9). Therefore, the burden of feature engineering has shifted from humans to computers, allowing nonexperts in machine learning to effectively use deep learning for their own research and/or applications, especially in medical image analysis.

The unprecedented success of deep learning is due mostly to the following factors: (*a*) advances in high-tech central processing units (CPUs) and graphics processing units (GPUs), (*b*) the availability of a huge amount of data (i.e., big data), and (*c*) developments in learning algorithms (10–14). Technically, deep learning can be regarded as an improvement over conventional artificial neural networks (15) in that it enables the construction of networks with multiple (more than two) layers. Deep neural networks can discover hierarchical feature representations such that higher-level features can be derived from lower-level features (9). Because these techniques enable hierarchical feature representations to be learned solely from data, deep learning has achieved record-breaking performance in a variety of artificial intelligence applications (16–23) and grand challenges (24, 25; see **https://grand-challenge.org**). In particular, improvements in computer vision prompted the use of deep learning in medical image analysis, such as image segmentation (26, 27), image registration (28), image fusion (29), image annotation (30), computer-aided diagnosis (CADx) and prognosis (31–33), lesion/landmark detection (34–36), and microscopic image analysis (37, 38).

Deep learning methods are highly effective when the number of available samples during the training stage is large. For example, in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), more than one million annotated images were available (24). However, in most medical applications there are far fewer images (i.e., <1,000). Therefore, a primary challenge in applying deep learning to medical images is the limited number of training samples available to build deep models without suffering from overfitting. To overcome this challenge, research groups have devised various strategies, such as (*a*) taking either two-dimensional (2D) or three-dimensional (3D) image patches, rather than the full-sized images, as input (29, 39–45) in order to reduce input dimensionality and thus the number of model parameters; (*b*) expanding the data set by artificially generating samples via affine transformation (i.e., data augmentation), and then training their network from scratch with the augmented data set (39–42); (*c*) using deep models trained on a huge number of natural images in computer vision as "off-the-shelf" feature extractors, and then training the final classifier or output layer with the target-task samples (43, 45); (*d*) initializing model parameters with those of pretrained models from nonmedical or natural images, then fine-tuning the network parameters with the task-related samples (46, 47); and (*e*) using models trained with small-sized inputs for arbitrarily sized inputs by transforming weights in the fully connected layers into convolutional kernels (36, 48).

In terms of input types, we can categorize deep models as typical multilayer neural networks that take vector-format (i.e., nonstructured) values as input and convolutional networks that take 2D or 3D (i.e., structured) values as input. Because of the structural characteristics of images (the structural or configural information contained in neighboring pixels or voxels is another important source of information), convolutional neural networks (CNNs) have attracted great interest in the field of medical image analysis (26, 35–37, 48–50). However, networks with vectorized inputs have also been successfully used in different medical applications (28, 29, 31, 33, 51–54). Along with deep neural networks, deep generative models (55)—such as deep belief networks (DBNs) and deep Boltzmann machines (DBMs), which are probabilistic graphical models with multiple layers of hidden variables—have been successfully applied to brain disease diagnosis (29, 33, 47, 56), lesion segmentation (36, 49, 57, 58), cell segmentation (37, 38, 59, 60), image parsing (61–63), and tissue classification (26, 35, 48, 50).

This review is organized as follows. In Section 2, we explain the computational theories of neural networks and deep models [e.g., stacked auto-encoders (SAEs), DBNs, DBMs, CNNs] and discuss how they extract high-level representations from data. In Section 3, we introduce recent studies

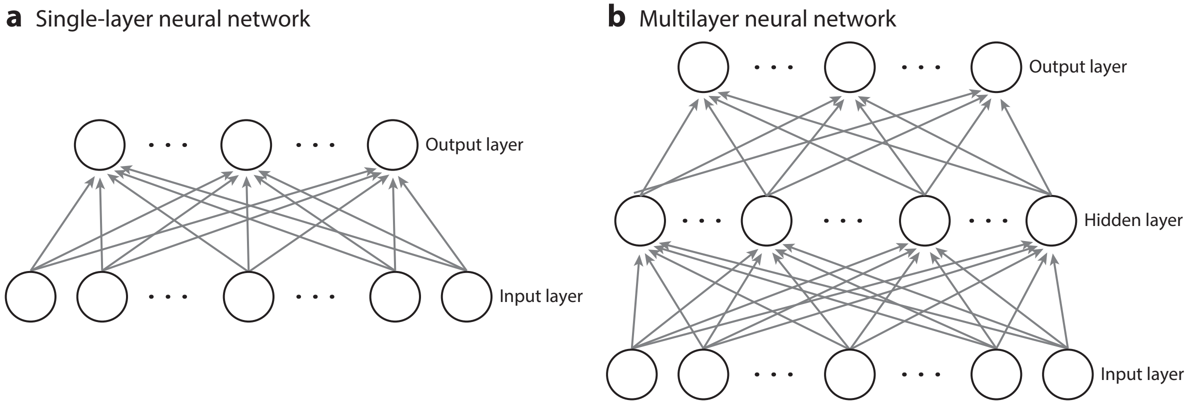

Figure 1

Architectures of two feed-forward neural networks.

using deep models for different applications in medical imaging, including image registration, anatomy localization, lesion segmentation, detection of objects and cells, tissue segmentation, and computer-aided detection (CADe) and CADx. Finally, in Section 4 we conclude by summarizing research trends and suggesting directions for further improvements.

2. DEEP LEARNING

In this section, we explain the fundamental concepts of feed-forward neural networks and basic deep models in the literature. We focus on learning hierarchical feature representations from data. We also discuss how to efficiently learn parameters of deep architecture by reducing overfitting.

2.1. Feed-Forward Neural Networks

In machine learning, artificial neural networks are a family of models that mimic the structural elegance of the neural system and learn patterns inherent in observations. The perceptron (64) is the earliest trainable neural network with a single-layer architecture,[1] composed of an input layer and an output layer. A perceptron, or a modified perceptron with multiple output units (**Figure 1*a***), is regarded as a linear model, prohibiting its application in tasks involving complicated data patterns, despite the use of nonlinear activation functions in the output layer.

This limitation can be overcome by introducing a so-called hidden layer between the input layer and the output layer. Note that in neural networks the units of the neighboring layers are fully connected to one another, but there are no connections among units in the same layer. For a two-layer neural network (**Figure 1*b***), also known as a multilayer perceptron, given an input vector $\mathbf{v} = [v_i] \in \mathbb{R}^D$, we can write the estimation function of an output unit y_k as a composition function as follows:

$$y_k(\mathbf{v}; \Theta) = f^{(2)}\left(\sum_{j=1}^{M} W_{kj}^{(2)} f^{(1)}\left(\sum_{i=1}^{D} W_{ji}^{(1)} v_i + b_j^{(1)}\right) + b_k^{(2)}\right), \tag{1}$$

[1] In general, the input layer is not counted.

where the superscript denotes a layer index, $f^{(1)}(\cdot)$ and $f^{(2)}(\cdot)$ denote nonlinear activation functions of units at the specified layers, M is the number of hidden units, and $\Theta = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}\}$ is a parameter set.[2] Conventionally, the hidden units' activation function, $f^{(1)}(\cdot)$, is commonly defined with a sigmoidal function such as a logistic sigmoid function or a hyperbolic tangent function, whereas the output units' activation function $f^{(2)}(\cdot)$ is dependent on the target task. Because the estimation proceeds in a forward direction, this type of network is also referred to as a feed-forward neural network.

When the hidden layer in Equation 1 is regarded as a feature extractor, $\boldsymbol{\phi}(\mathbf{v}) = [\phi_j(\mathbf{v})] \in \mathbb{R}^M$ from an input $\mathbf{v}$, the output layer is only a simple linear model,

$$y_k(\mathbf{v}; \Theta) = f^{(2)}\left(\sum_{j=1}^{M} W_{kj}^{(2)} \phi_j(\mathbf{v}) + b_k^{(2)}\right), \tag{2}$$

where $\phi_j(\mathbf{v}) \equiv f^{(1)}\left(\sum_{i=1}^{D} W_{ji}^{(1)} v_i + b_j^{(1)}\right)$. The same interpretation holds when there is a higher number of hidden layers. Thus, it is intuitive that the role of hidden layers is to find features that are informative for the target task.

The practical use of neural networks requires that the model parameters Θ be learned from data. The problem of parameter learning can be formulated as the minimization of the error function. From an optimization perspective, the error function E for neural networks is highly nonlinear and nonconvex. Thus, there is no analytic solution of the parameter set Θ. Instead, one can use a gradient descent algorithm by updating the parameters iteratively. In order to utilize a gradient descent algorithm, there must be a way to compute a gradient $\nabla E(\Theta)$ evaluated at the parameter set Θ.

For a feed-forward neural network, the gradient can be efficiently evaluated by means of error back-propagation (65). Once the gradient vector of all the layers is known, the parameters $\Theta \in \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}\}$ can be updated as follows:

$$\Theta^{(\tau+1)} = \Theta^{(\tau)} - \eta \nabla E\left(\Theta^{(\tau)}\right), \tag{3}$$

where η is a learning rate and τ denotes an iteration index. The update process is repeated until convergence or until the predefined number of iterations is reached. As for the parameter update in Equation 3, the stochastic gradient descent with a small subset of training samples, termed a minibatch, is commonly used in the literature (66).

2.2. Deep Models

Under a mild assumption on the activation function, a two-layer neural network with a finite number of hidden units can approximate any continuous function (67); therefore, it is regarded as a universal approximator. However, it is also possible to approximate complex functions to the same accuracy by using a deep architecture (i.e., one with more than two layers), with a far fewer number of units (8). Thus, it is possible to reduce the number of trainable parameters, enabling training with a relatively small data set (68).

[2] $\mathbf{W}^{(1)} = [W_{ji}^{(1)}] \in \mathbb{R}^{M \times D}; \mathbf{W}^{(2)} = [W_{kj}^{(2)}] \in \mathbb{R}^{K \times M}; \mathbf{b}^{(1)} = [b_j^{(1)}] \in \mathbb{R}^M; \mathbf{b}^{(2)} = [b_k^{(2)}] \in \mathbb{R}^K$.

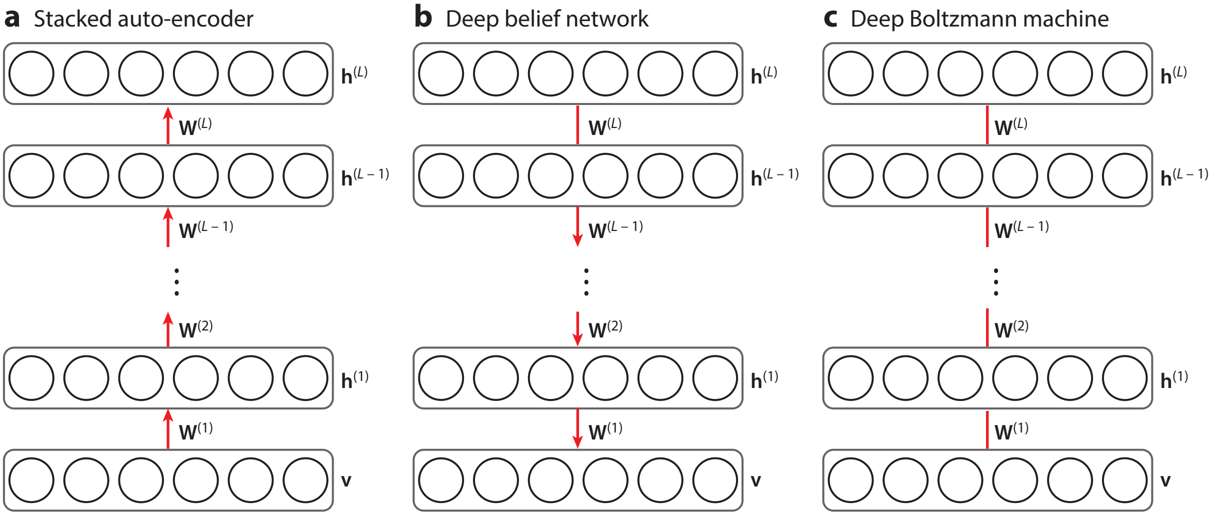

Figure 2

Three representative deep models with vectorized inputs for unsupervised feature learning. The red links, whether directed or undirected, denote the full connections of units in two consecutive layers but no connections among units in the same layer. Note the differences among models in directed/undirected connections and the directions of the connections that depict conditional relationships.

2.3. Unsupervised Feature Representation Learning

Compared with shallow architectures that require a good feature extractor designed mostly by hand on the basis of expert knowledge, deep models are useful for discovering informative features from data in a hierarchical manner (i.e., from fine to abstract). Here, we introduce three deep models that are widely used in different applications for unsupervised feature representation learning.

2.3.1. Stacked auto-encoder. An auto-encoder or auto-associator (69) is a special type of two-layer neural network that learns a latent or compressed representation of the input by minimizing the reconstruction error between the input and output values of the network, namely the reconstruction of the input from the learned representations. Because of its simple, shallow structure, a single-layer auto-encoder's representational power is very limited. But when multiple auto-encoders are stacked (**Figure 2*a***) in a configuration called an SAE, one can significantly improve the representational power by using the activation values of the hidden units of one auto-encoder as the input to the next higher auto-encoder (70). One of the most important characteristics of SAEs is their ability to learn or discover highly nonlinear and complicated patterns, such as the relations among input values. When an input vector is presented to an SAE, the different layers of the network represent different levels of information. That is, the lower the layer in the network is, the simpler the patterns are, and the higher the layer is, the more complicated or abstract the patterns inherent in the input vector are.

With regard to training parameters of the weight matrices and the biases in SAE, a straightforward approach is to apply back-propagation to the gradient-based optimization technique, beginning from random initialization by using the SAE as a conventional feed-forward neural network. Unfortunately, deep networks trained in this manner perform worse than networks with a shallow architecture, as they fall into a poor local optimum (71). To circumvent this problem, one should consider greedy layer-wise learning (10, 72). The key idea of greedy layer-wise learning is to pretrain one layer at a time. That is, the user trains parameters of the first hidden layer with the training data as input, and then trains parameters of the second hidden layer with the output from the first hidden layer as input, and so on. In other words, the representation of the lth hidden layer is used as input for the $(l+1)$-th hidden layer. An important advantage of such a pretraining technique is that it is conducted in an unsupervised manner with a standard back-propagation algorithm, enabling the user to increase the size of the data set by exploiting unlabeled samples for training.

2.3.2. Deep belief network. A restricted Boltzmann machine (RBM) (73) is a single-layer undirected graphical model with a visible layer and a hidden layer. It assumes symmetric connectivities between visible and hidden layers, but no connections among units within the same layer. Because of the symmetry of the connectivities, it can generate input observations from hidden representations. Therefore, an RBM naturally becomes an auto-encoder (10, 73), and its parameters are usually trained by use of a contrastive divergence algorithm (74) so as to maximize the log likelihood of observations. Like SAEs, RBMs can be stacked in order to construct a deep architecture, resulting in a single probabilistic model called a DBN. A DBN has one visible layer $\mathbf{v}$ and a series of hidden layers $\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(L)}$ (**Figure 2*b***). Note that when multiple RBMs are stacked hierarchically, although the top two layers still form an undirected generative model (i.e., an RBM), the lower layers form directed generative models. Thus, the joint distribution of the observed units $\mathbf{v}$ and the L hidden layers $\mathbf{h}^{(l)}$ $(l = 1, \ldots, L)$ in DBN is

$$P\left(\mathbf{v}, \mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(L)}\right) = \left(\prod_{l=0}^{L-2} P(\mathbf{h}^{(l)}|\mathbf{h}^{(l+1)})\right) P\left(\mathbf{h}^{(L-1)}, \mathbf{h}^{(L)}\right), \tag{4}$$

where $P(\mathbf{h}^{(l)}|\mathbf{h}^{(l+1)})$ corresponds to a conditional distribution for the units of layer l given the units of layer $l+1$, and $P(\mathbf{h}^{(L-1)}, \mathbf{h}^{(L)})$ denotes the joint distribution of the units in layers $L-1$ and L.

Regarding the learning of parameters, the greedy layer-wise pretraining scheme (10) can be applied in the following steps.

1. Train the first layer as an RBM with $\mathbf{v} = \mathbf{h}^{(0)}$.
2. Use the first hidden layer to obtain the representation of inputs with either the mean activations of $P(\mathbf{h}^{(1)} = 1|\mathbf{h}^{(0)})$ or samples drawn according to $P(\mathbf{h}^{(1)}|\mathbf{h}^{(0)})$, which will be used as observations for the second hidden layer.
3. Train the second hidden layer as an RBM, taking the transformed data (mean activations or samples) as training examples (for the visible layer of the RBM).
4. Iterate steps 2 and 3 for the desired number of layers, each time propagating upward either mean activations $P(\mathbf{h}^{(l+1)} = 1|\mathbf{h}^{(l)})$ or samples drawn according to the conditional probability $P(\mathbf{h}^{(l+1)}|\mathbf{h}^{(l)})$.

After the greedy layer-wise training procedure is complete, one can apply the wake–sleep algorithm (75) to further increase the log likelihood of the observations. Usually, however, no further procedure is conducted to train the whole DBN jointly in practice.

2.3.3. Deep Boltzmann machine. A DBM (55) is also constructed by stacking multiple RBMs in a hierarchical manner. However, in contrast to DBNs, all the layers in DBMs form an undirected generative model following the stacking of RBMs (**Figure 2*c***). Thus, for hidden layer l, except in the case of $l = 1$ and $l = L$, the layer's probability distribution is conditioned by its two neighboring layers, $l+1$ and $l-1$ [i.e., $P(\mathbf{h}^{(l)}|\mathbf{h}^{(l+1)}, \mathbf{h}^{(l-1)})$]. The incorporation of information from both the upper and lower layers improves a DBM's representational power so that it is more robust to noisy observations.

Let us consider a three-layer DBM, namely the $L = 2$ DBM shown in **Figure 2*c***. Given the values of the units in the neighboring layer(s), the probability of either the binary visible or binary hidden units being set to one is computed as follows:

$$P\left(h_j^{(1)} = 1|\mathbf{v}, \mathbf{h}^{(2)}\right) = \sigma\left(\sum_i W_{ij}^{(1)} v_i + \sum_k W_{jk}^{(2)} h_k^{(2)}\right), \tag{5}$$

$$P\left(h_k^{(2)} = 1|\mathbf{h}^{(1)}\right) = \sigma\left(\sum_j W_{jk}^{(2)} h_j^{(1)}\right), \tag{6}$$

$$P\left(v_i = 1|\mathbf{h}^{(1)}\right) = \sigma\left(\sum_j W_{ij}^{(1)} h_j^{(1)}\right), \tag{7}$$

where $\sigma(\cdot)$ denotes a logistic sigmoid function. In order to learn the parameters $\Theta = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}\}$,[3] we maximize the log likelihood of the observations. The derivative of the log likelihood of the observations with respect to the model parameters takes the following simple form:

$$\frac{\partial}{\partial \mathbf{W}^{(l)}} \ln P(\mathbf{v}; \Theta) = \mathbb{E}_{\text{data}}\left[\mathbf{h}^{(l-1)}(\mathbf{h}^{(l)})^\top\right] - \mathbb{E}_{\text{model}}\left[\mathbf{h}^{(l-1)}(\mathbf{h}^{(l)})^\top\right], \tag{8}$$

where $\mathbb{E}_{\text{data}}[\cdot]$ denotes the data-dependent statistics obtained by sampling the model conditioned on the visible units $\mathbf{v}(= \mathbf{h}^{(0)})$ and $\mathbb{E}_{\text{model}}[\cdot]$ denotes the data-independent statistics obtained by sampling from the model. When the model approximates the data distribution well, data-dependent and data-independent statistics reach equilibrium.

2.4. Fine-Tuning Deep Models for Target Tasks

Note that, during feature representation learning for the three deep models described above, the target values (either discrete labels or continuous real values of observations) are never involved. Therefore, there is no guarantee that the representations learned by SAEs, DBNs, or DBMs are discriminative for a classification task, for example. To address this problem, the so-called fine-tuning step is generally followed after unsupervised feature representation learning.

For a certain task involving either classification or regression, it is straightforward to convert feature representation learning models into a deep neural network by stacking another output layer on top of the highest hidden layer in an SAE, DBN, or DBM with an appropriate output function. In the case of a DBM, the original input vector should first be augmented with the marginals of the approximate posterior of the second hidden layer as a by-product when converting a DBM into a deep neural network (55). The top output layer is then used to predict the target value(s) of an input. To fine-tune the parameters in a deep neural network, we first take the pretrained connection weights of the hidden layers as the initial values, randomly initialize the connection weights between the top hidden layer and the output layer, and then train the parameters jointly in a supervised (i.e., end-to-end) manner by gradient descent with a back-propagation algorithm. Initialization of the parameters via pretraining helps the supervised optimization reduce the risk of falling into poor local optima (10, 71).

[3] For simplicity, bias parameters are omitted.

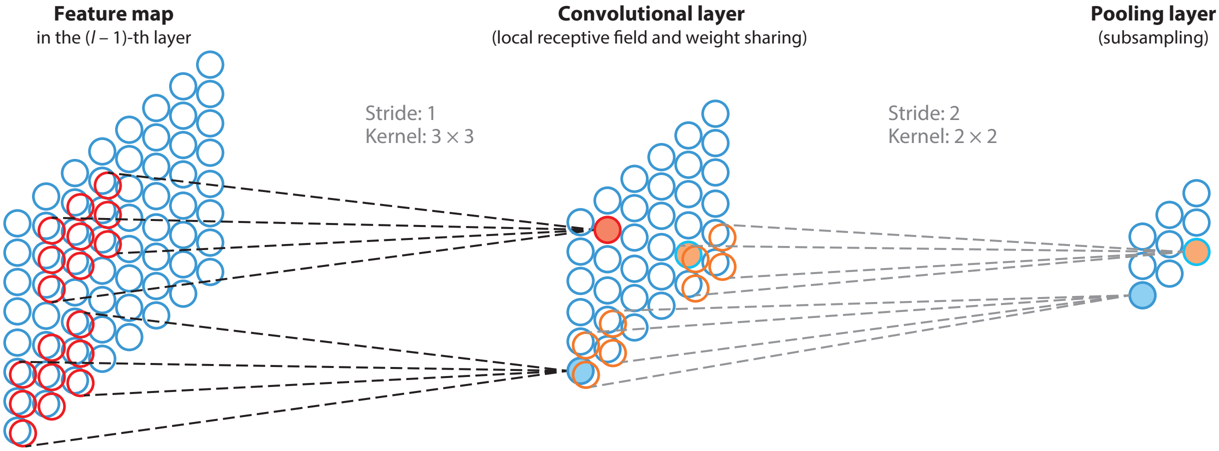

Figure 3

Three key mechanisms (i.e., local receptive field, weight sharing, and subsampling) in convolutional neural networks.

2.5. Convolutional Neural Networks

In the deep models of SAEs, DBNs, and DBMs, described above, the inputs are always in vector form. However, for (medical) images, the structural information among neighboring pixels or voxels is also important, but vectorization inevitably destroys such structural and configural information in images. CNNs (76) are designed to better utilize spatial and configural information by taking 2D or 3D images as input. Structurally, CNNs have convolutional layers interspersed with pooling layers, followed by fully connected layers as in a standard multilayer neural network. Unlike a deep neural network, a CNN exploits three mechanisms—a local receptive field, weight sharing, and subsampling (**Figure 3**)—that greatly reduce the degrees of freedom in a model.

The role of a convolutional layer is to detect local features at different positions in the input feature maps with learnable kernels $k_{ij}^{(l)}$, namely connection weights between the feature map i at layer $l-1$ and the feature map j at layer l. Specifically, the units of the convolutional layer l compute their activation $\mathbf{A}_j^{(l)}$ on the basis of only a spatially contiguous subset of units in the feature maps $\mathbf{A}_i^{(l-1)}$ of the preceding layer $l-1$ by convolving the kernels $k_{ij}^{(l)}$ as follows:

$$\mathbf{A}_j^{(l)} = f\left(\sum_{i=1}^{M^{(l-1)}} \mathbf{A}_i^{(l-1)} * k_{ij}^{(l)} + b_j^{(l)}\right), \tag{9}$$

where $M^{(l-1)}$ denotes the number of feature maps in layer $l-1$, the asterisk denotes a convolutional operator, $b_j^{(l)}$ is a bias parameter, and $f(\cdot)$ is a nonlinear activation function. Due to the mechanisms of weight sharing and local receptive field, when the input feature map is slightly shifted, the activation of the units in the feature maps is shifted by the same amount.

A pooling layer follows a convolutional layer to downsample the feature maps of the preceding convolutional layer. Specifically, each feature map in a pooling layer is linked to a feature map in the convolutional layer; each unit in a feature map of the pooling layer is computed on the basis of a subset of units within a local receptive field from the corresponding convolutional feature map. Similar to the convolutional layer, the receptive field finds a representative value (e.g., maximum or average) among the units in its field. Usually, a change in the size of the receptive field during convolution is set equal to the size of the receptive field for subsampling, helping the CNN to be translation invariant.

Theoretically, the gradient descent method combined with a back-propagation algorithm can also be applied to the learning parameters of a CNN. However, due to the special mechanisms of weight sharing, the local receptive field, and pooling, slight changes need to be made; that is, one needs to sum the gradients for a given weight over all the connections using the kernel weights in order to determine which patch in the layer's feature map corresponds to a unit in the next layer's feature map, and to upsample the feature maps of the pooling layer to recover the reduced size of the maps.

2.6. Reducing Overfitting

A critical challenge in training deep models arises from the limited number of training samples compared with the number of learnable parameters. Thus, reducing overfitting has long presented a challenge. Recent studies have devised algorithmic techniques to better train deep models. Some of the techniques are as follows.

1. Initialization/momentum (77, 78) involves the use of well-designed random initialization and a particular schedule of slowly increasing the momentum parameter as iteration passes.
2. Rectified linear unit (ReLU) (12, 79, 80) applies to nonlinear activation functions.
3. Denoising (11) involves stacking layers of denoising auto-encoders, which are trained locally to reconstruct the original "clean" inputs from the corrupted versions of them.
4. Dropout (13) and DropConnect (81) randomly deactivate a fraction (e.g., 50%) of the units or connections in a network on each training iteration.
5. Batch normalization (14) performs normalization for each minibatch and back-propagates the gradients through the normalization parameters.

See the references cited for further details.

3. APPLICATIONS IN MEDICAL IMAGING

Compared with other machine learning techniques in the literature, deep learning has witnessed significant advances. These successes have prompted researchers in the field of computational medical imaging to investigate the potential of deep learning in medical images acquired with, for example, CT, MRI, PET, and X-ray. In this section, we discuss the practical applications of deep learning in image registration and localization, detection of anatomical and cellular structures, tissue segmentation, and computer-aided disease prognosis and diagnosis.

3.1. Deep Feature Representation Learning in Medical Images

Many existing medical image processing methods rely on morphological feature representations to identify local anatomical characteristics. However, such feature representations were designed mostly by human experts, and the image features are often problem specific and not guaranteed to work for other image types. For instance, image segmentation and registration methods designed for 1.5-T T1-weighted brain MR images are not applicable to 7.0-T T1-weighted MR images (28, 52), not to mention to other modalities or different organs. Furthermore, 7.0-T MR images can reveal the brain's anatomy with a resolution equivalent to that obtained from thin slices in vitro (82). Thus, researchers can clearly observe fine brain structures at the micrometer scale, which previously was possible only with in vitro imaging. However, the lack of efficient computational tools substantially hinders the translation of new imaging techniques into the medical imaging arena.

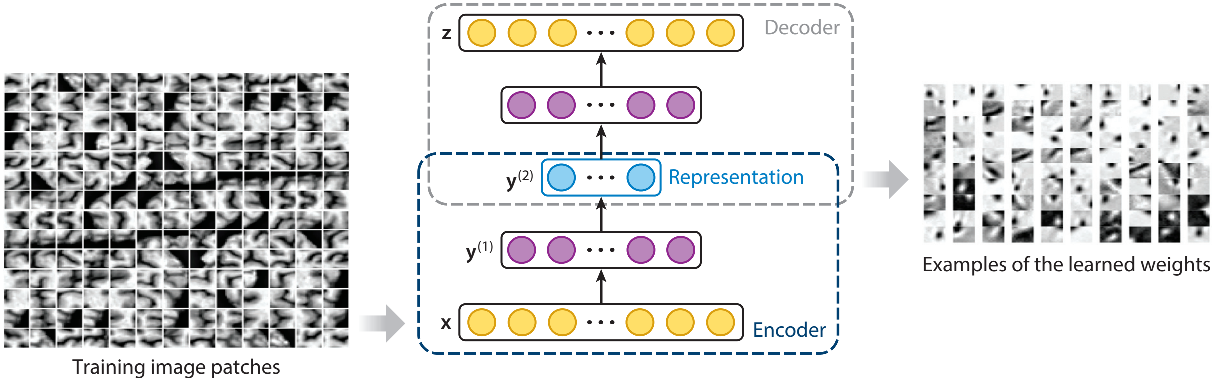

Figure 4

Construction of a deep encoder–decoder via a stacked auto-encoder and visualization of the learned feature representations. The blue circles represent high-level feature representations. The yellow and purple circles indicate the correspondence between layers in the encoder and decoder.

Although state-of-the-art methods use supervised learning to find the most relevant and essential features for target tasks, they require a significant number of manually labeled training data, and the learned features may be superficial and may misrepresent the complexity of the anatomical structures. More critically, the learning procedure is often confined to a particular template domain, with a certain number of predesigned features. Therefore, once the template or image features change, the entire training process has to start over again. To address these limitations, Wu et al. (28, 52) developed a general feature representation framework that can (*a*) capture the intrinsic characteristics of anatomical structures necessary for accurate brain region segmentation and correspondence detection and (*b*) be flexibly applied to different kinds of medical images. Specifically, these authors used an SAE with a sparsity constraint, which they therefore termed a sparse auto-encoder, to hierarchically learn feature representations in a layer-by-layer manner. Their SAE model consisted of encoding and decoding modules hierarchically (**Figure 4**). In the encoding module, given an input image patch **x**, the model mapped the input to an activation vector $\mathbf{y}^{(1)}$ through nonlinear deterministic mapping. The authors then repeated this procedure by using $\mathbf{y}^{(1)}$ as the input to train the second layer, and so forth, until they obtained high-level feature representations (**Figure 4**). The decoding module was used to validate the expressive power of the learned feature representations by minimizing the reconstruction errors between the input image patch **x** and the reconstructed patch **z** after decoding.

Figure 5 demonstrates the power of feature representations learned by deep learning methods. **Figure 5*a–c*** shows a typical image registration result for brain images of an elderly patient, and **Figure 5*d–f*** compares different feature representations for finding a correspondence of a template point. Clearly, the deformed subject image in **Figure 5*c*** is far from being well registered with the template image in **Figure 5*a***, especially for ventricles. It is very difficult to learn meaningful features given such inaccurate correspondences derived from imperfect image registration, a problem that many supervised learning methods suffer from (83–85). Moreover, the features [e.g., local patches and scale-invariant feature transform (SIFT) (86)] either detect too many noncorresponding points when using the entire intensity patch as the feature vector (**Figure 5*d***) or have too-low responses and thus miss the correspondence when using SIFT (**Figure 5*e***). Meanwhile, SAE-learned feature representations present the least confusing correspondence information for

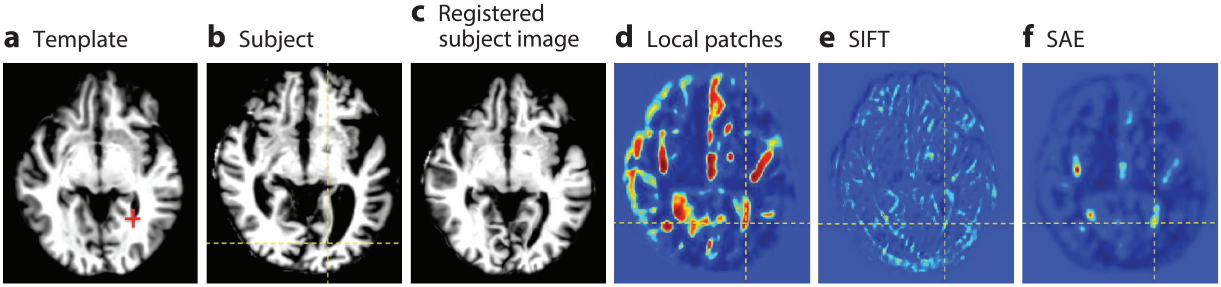

Figure 5

Similarity maps identifying the correspondence for the point indicated by the red cross in the template (*a*) with regard to the subject (*b*) by hand-designed features (*d*,*e*) and by stacked auto-encoder (SAE) features learned through unsupervised deep learning (*f*). The registered subject image is shown in panel *c*. Clearly, inaccurate registration results might undermine supervised feature representation learning, which relies strongly on the correspondences across all training images. In panels *d*–*f*, the different colors of the voxels indicate their likelihood of being selected as correspondence for their respective locations. Abbreviation: SIFT, scale-invariant feature transform.

the subject point under consideration, making it easy to locate the correspondence of the template point in the subject image domain.

In order to qualitatively evaluate the registration accuracy, Wu et al. obtained deformable image registration results over various public data sets (**Figure 6**). Compared with the state-of-the-art registration methods of intensity-based diffeomorphic Demons (87) and feature-based

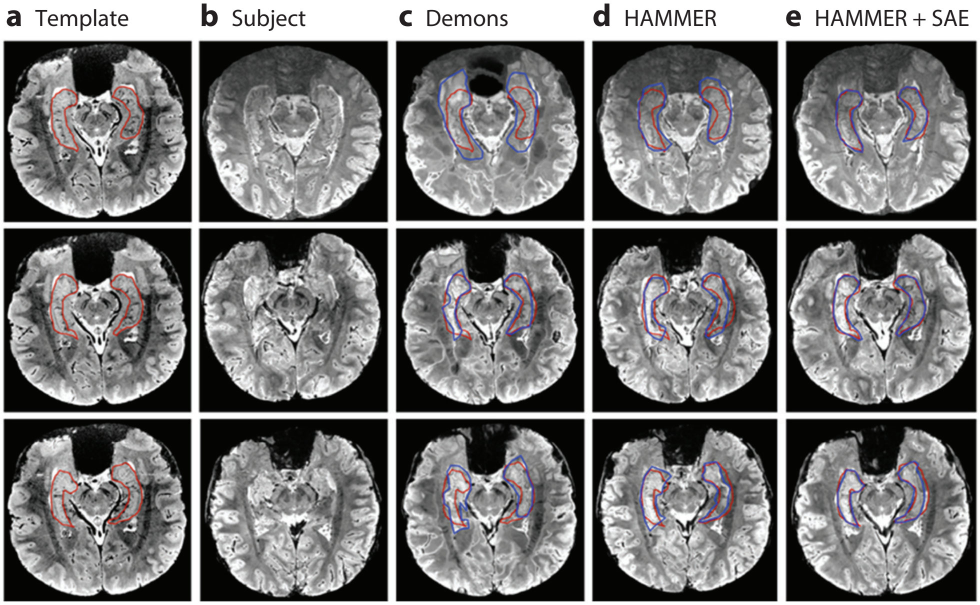

Figure 6

Typical registration results on 7.0-T magnetic resonance images of the brain by (*c*) Demons (87), (*d*) HAMMER (88), and (*e*) HAMMER combined with stacked auto-encoder (SAE)-learned feature representations. The three rows represent three different slices of the template, subject, and registered subjects. The manually labeled hippocampus on the template image and the deformed subject's hippocampus by different registration methods are marked by red and blue contours, respectively.

HAMMER (88) for 1.5- and 3.0-T MR images, the SAE-learned feature representation depicted in **Figure 6*e*** performs better.

Another successful medical application involves localizing a prostate from MR images (89, 90). Accurate prostate localization in MR images is difficult for two reasons: (*a*) The appearance patterns around the prostate boundary vary significantly between patients, and (*b*) the intensity distributions vary between patients and often do not follow a Gaussian distribution. To address these challenges, Guo et al. (90) used an SAE to learn hierarchical feature representations from MR prostate images. The resulting learned features were integrated into a sparse patch-matching framework to find the corresponding patches in the atlas images for label propagation (91). Finally, a deformable model was employed to segment the prostate by combining the shape prior with the prostate likelihood map derived from sparse patch matching. **Figure 7** shows typical prostate segmentation results from different patients, produced by three different feature representations.

The applications described above demonstrate that (*a*) the latent feature representations inferred by deep learning can successfully describe local image characteristics; (*b*) researchers can rapidly develop image analysis methods for new medical imaging modalities by using a deep learning framework to learn the intrinsic feature representations; and (*c*) the entire learning-based framework can be adapted to learn imaging feature representations and extended to various medical imaging applications, such as hippocampus segmentation (92) and prostate localization in MR images (89, 90).

3.2. Deep Learning for Detection of Anatomical Structures

Localization and interpolation of anatomical structures in medical images are key steps in the radiological workflow. Radiologists usually accomplish these tasks by identifying certain anatomical signatures, namely image features that can distinguish one anatomical structure from others. Is it possible for a computer to automatically learn such anatomical signatures? The success of such methods essentially depends on how many anatomy signatures can be extracted by computational operations. Whereas early studies often created specific image filters to extract anatomical signatures, more recent research has revealed that deep learning–based approaches have become prevalent for two reasons: (*a*) Deep learning technologies are now mature enough to solve real-world problems, and (*b*) more and more medical image data sets have become available to facilitate the exploration of big medical image data.

3.2.1. Detection of organs and body parts.

Shin et al. (51) used SAEs to separately learn both visual and temporal features in order to detect multiple organs in a time series of 3D dynamic contrast–enhanced MRI scans over data sets from two studies of liver metastases and one study of kidney metastases. Unlike conventional SAEs, the SAE in this study involved the application of a pooling operation after each layer so that features of progressively larger input regions were essentially compressed. Because different organ classes have different properties, the authors trained multiple models to separate each organ from all of the other organs in a supervised manner.

Roth et al. (93) presented a method for organ- or body part–specific anatomical classification of medical images using deep convolutional networks. Specifically, they trained their deep network by using 4,298 axial 2D CT images to learn five parts of the body: neck, lungs, liver, pelvis, and legs. Their experiments achieved an anatomy-specific classification error of 5.9% and an average AUC (area under the receiver-operating characteristic curve) value of 0.998. However, real-world applications may require more finely grained differentiation than that used for only five body parts (e.g., they may need to distinguish aortic arch from cardiac sections). To address this limitation, Yan et al. (94, 95) designed a multistate deep learning framework with a CNN to identify the body part

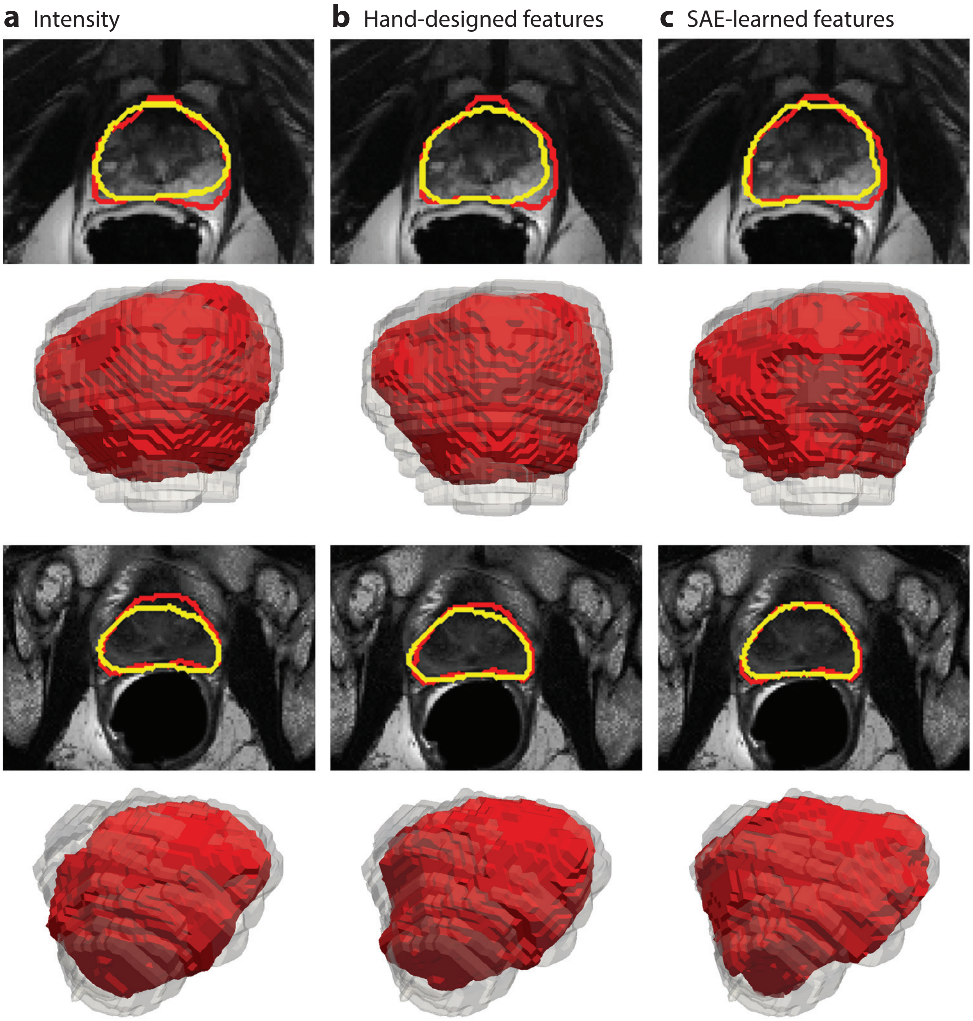

Figure 7

Typical prostate segmentation results of two different patients produced by three different feature representations. Red contours indicate manual ground-truth segmentations, and yellow contours indicate automatic segmentations. The second and fourth rows present a three-dimensional (3D) visualization of the segmentation results corresponding to the images above. For each 3D visualization, the red surfaces indicate the automatic segmentation results using different features, such as intensity, hand-designed features, and stacked auto-encoder (SAE)-learned features. The transparent gray surfaces represent ground-truth segmentations.

of a transversal slice. Because each slice may contain multiple organs (enclosed in bounding boxes), the CNN was trained in a multi-instance fashion (96) in which the objective function was adapted such that, as long as one organ was correctly labeled, the corresponding slice was considered correct. Therefore, the pretrained CNN was sensitive to the discriminative bounding boxes. On the basis of the pretrained CNN's responses, discriminative and noninformative bounding boxes were selected to further boost the representation power of the pretrained CNN. At run time, a sliding-window approach was employed to apply the boosted CNN to the subject image. Because the CNN had peaky responses only on discriminative bounding boxes, it essentially identified body parts by focusing on the most distinctive local information. Compared with global image context-based approaches, this local approach was more accurate and robust. These authors' body part recognition method was tested on 12 body parts on 7,489 CT slices, collected from scans of 675 patients varying in age from 1 to 90 years. The entire data set was divided into three groups: 2,413 (225 patients) for training, 656 (56 patients) for validation, and 4,043 (394 patients) for testing.

3.2.2. Cell detection.

Digitized tissue histopathology has recently been employed for microscopic examination and automatic disease grading. A primary challenge in microscopic image analysis involves the need to analyze all individual cells for accurate diagnosis, given that the differentiation of most disease grades depends strongly on cell-level information. To address this challenge, researchers have employed deep CNNs to robustly and accurately detect and segment cells from histopathological images (37, 38, 53, 54, 97–99), which can significantly benefit cell-level analysis for cancer diagnosis.

In a pioneering study, Cireşan et al. (37) used a deep CNN to detect mitosis in breast cancer histology images. Their networks were trained to classify each pixel in the images from a patch centered on the pixel. Their method won the 2012 International Conference on Pattern Recognition (ICPR) Mitosis Detection Contest,[4] outperforming other contestants by a significant margin.

Since then, different groups have used different deep learning methods for detection in histology images. For example, Xu et al. (54) used an SAE to detect cells on breast cancer histological images. To train their deep model, they utilized a denoising auto-encoder to improve robustness to outliers and noises. Su et al. (53) also used an SAE as well as sparse representation to detect and segment cells from microscopic images. Sirinukunwattana et al. (100) proposed a spatially constrained CNN (SC-CNN) to detect and classify nuclei in histopathology images. Specifically, they used an SC-CNN to estimate the likelihood of a pixel being the center of a nucleus, where pixels with high probability values were spatially constrained to locate in the vicinity of the center of nuclei. They also developed a neighboring ensemble predictor coupled with a CNN to more accurately predict the class label of the detected cell nuclei. Chen et al. (38) designed a deep cascaded CNN by exploiting the technique of the full CNN, which replaces the fully connected layers with all-convolutional kernels (101). They first trained a coarse retrieval model to identify and locate mitosis candidates while maintaining high sensitivity. On the basis of the retrieved candidates, they then created a fine discrimination model by transferring deep and rich feature hierarchies learned on a large natural image data set to distinguish mitoses from hard mimics. Their cascaded CNN achieved the best detection accuracy in the 2014 ICPR MITOS-ATYPIA challenge.[5]

3.3. Deep Learning for Segmentation

Automatic segmentation of brain images is a prerequisite for quantitative assessment of the brain in patients of all ages. An important step in brain image preprocessing involves removing nonbrain regions such as the skull. Although current methods demonstrate good results on nonenhanced T1-weighted images, they still struggle when applied to other modalities and pathologically altered tissues. To circumvent such limitations, Kleesiek et al. (27) used 3D convolutional deep learning architecture for skull extraction, a technique that was not limited to nonenhanced T1-weighted MR images. While training their 3D CNN, they constructed minibatches of multiple cubes that

[4] For details, refer to **http://ludo17.free.fr/mitos_2012/index.html**.

[5] For details, refer to **http://mitos-atypia-14.grand-challenge.org/**.

were larger than the actual input to their 3D CNN for computational efficiency. Specifically, their deep model could take an arbitrary-sized 3D patch as input by building a fully convolutional network (101); thus, the output could be a block of predictions per input, rather than a single prediction as in a conventional CNN. Over four different data sets, their method achieved the highest average specificity measures in comparison to six commonly used tools (i.e., BET, BEaST, BSE, ROBEX, HWA, and 3dSkullStrip), whereas its sensitivity was about average.

Moeskops et al. (102) devised a multiscale CNN to enhance robustness in neonatal image segmentation and spatial consistency. Their network used multiple patch sizes and multiple convolution kernel sizes to acquire multiscale information about each voxel. Using this method, the authors obtained promising segmentation results for eight tissue types, with a Dice ratio[6] averaging 0.82 to 0.91 over five different data sets.

The first year of life is the most dynamic phase of postnatal human brain development, characterized by rapid tissue growth and development of a wide range of cognitive and motor functions. Accurate tissue segmentation of infant brain MR images into white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF) in this phase is crucial in studies of normal and abnormal early brain development. Segmentation of infants' brain MR images is considerably more difficult to perform than for adults because of the reduced tissue contrast (103), increased noise, severe partial volume effect (104), and ongoing WM myelination (103, 105). Specifically, the WM and GM exhibit almost the same intensity level (especially in the cortical regions), resulting in low image contrast. Although many methods have been proposed for infant brain image segmentation, most focus on segmentation of images of either neonates (∼3 months) or infants (>12 months) using a single T1-weighted or T2-weighted image (106–110). Few studies have addressed the challenges posed by segmentation of isointense-phase images (around 6 months old).

To overcome these difficulties, Zhang et al. (26) designed four CNN architectures to segment infant brain tissues on the basis of multimodal MR images. Specifically, each CNN contained three input feature maps corresponding to T1-weighted, T2-weighted, and fractional anisotropy (FA) image patches measuring 13×13 voxels. The authors applied to each CNN three convolutional layers and one fully connected layer, followed by an output layer with a softmax function for tissue classification. On a set of manually segmented isointense-phase brain images, these CNNs significantly outperformed competing methods.

More recently, Nie et al. (48) proposed the use of multiple fully convolutional networks (mFCNs) (**Figure 8**) to segment isointense-phase brain images with T1-weighted, T2-weighted, and FA modality information. Instead of simply combining three-modality data from the original (low-level) feature maps, they employed a deep architecture to effectively fuse high-level information from all three modalities. They assumed that high-level representations from different modalities were complementary to one another. First, the authors trained one network for each modality in order to effectively employ information from multiple modalities; second, they fused multiple-modality features from the high layer of each network (**Figure 8**). In these experiments, the mFCNs achieved average Dice ratios of 0.852 for CSF, 0.873 for GM, and 0.887 for WM from eight subjects, outperforming fully convolutional networks and other competing methods.

3.4. Deep Learning for Computer-Aided Detection

The goal of CADe is to find or localize abnormal or suspicious regions in structural images, and thus to alert clinicians. CADe aims to increase the detection rate of diseased regions while

[6] $D(A, B) = 2(A \cap B)/(A + B)$, where $\cap$ is the intersection.

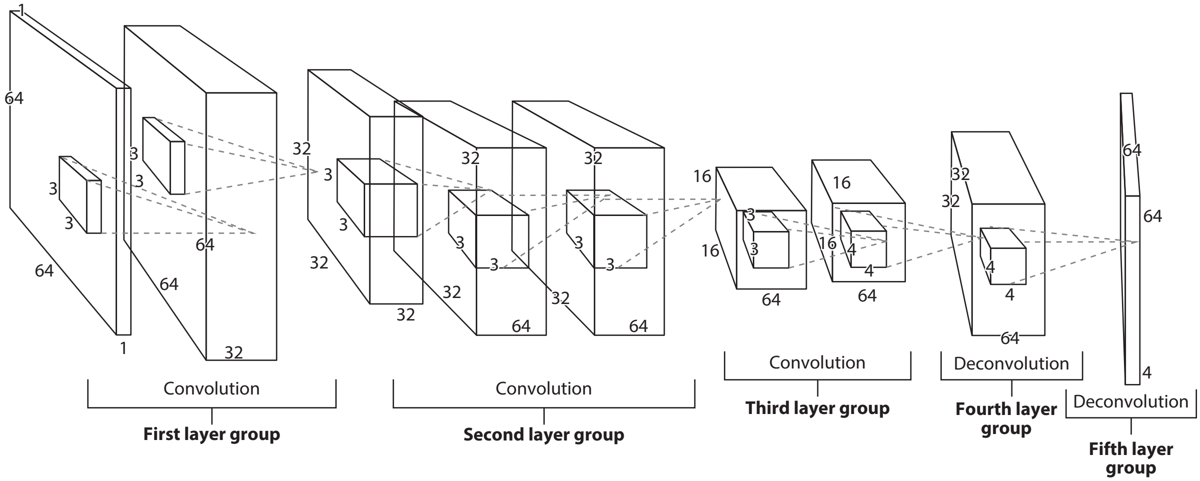

Figure 8

The architecture of the fully convolutional network used for tissue segmentation in Reference 48.

reducing the false-negative rate, which may be due to error or fatigue on the part of the observers. Although CADe is well established in medical imaging, deep learning methods have improved its performance in different clinical applications.

Typically, CADe occurs as follows: (*a*) The candidate regions are detected by means of image processing techniques; (*b*) the candidate regions are represented by a set of features, such as morphological or statistical information; and (*c*) the features are fed into a classifier, such as a support vector machine (SVM), to output a probability or make a decision as to whether disease is present. As explained in Section 1, human-designed feature representations can be incorporated into deep learning. Many groups have successfully used their own deep models in applications such as detection of pulmonary nodules, detection of lymph nodes, classification of interstitial lung disease in CT images, detection of cerebral microbleeds, and detection of multiple sclerosis lesions in MR images. Notably, most of the methods described in the literature exploited deep convolutional models to maximally utilize structural information in two, two-and-a-half, or three dimensions.

Ciompi et al. (43) used a pretrained OverFeat (111) out of the box as a feature extractor and empirically showed that a CNN learned from a completely different domain of natural images can provide useful feature descriptions for classification of pulmonary perifissural nodules. Roth et al. (40) focused on training deep models from scratch. To confront the problem of data insufficiency in training deep CNNs, they expanded their data set by scaling, translation, and rotation in random overtraining samples. They augmented the test samples in a similar way; obtained CNN outputs for every augmented test sample; and took the average of the outputs of the randomly transformed, scaled, and rotated patches for detection of lymph nodes and colonic polyps. To better utilize volumetric information in images, both Ciompi et al. (43) and Roth et al. (40) considered two-and-a-half-dimensional (2.5D) information with 2D patches of three orthogonal views (axial, sagittal, and coronal). Setio et al. (42) considered three sets of orthogonal views for a total of nine views from a 3D patch and used ensemble methods to fuse information from different views for detection of pulmonary nodules.

Gao et al. (112) focused on the holistic classification of CT patterns for interstitial lung disease by using a deep CNN. They borrowed the network architecture from Reference 113, with six units at the output layer, to classify patches into normal, emphysema, ground glass, fibrosis, micronodules, and consolidation. To overcome the overfitting problem, they utilized a data augmentation strategy by generating images by randomly jittering and cropping 10 subimages per original CT slice. At the testing stage, they generated 10 jittered images and fed them into the trained CNN. Finally, they predicted the input slice by aggregation, similar to the research by Roth et al. (40).

Shin et al. (45) conducted experiments on data sets of thoraco-abdominal lymph node detection and interstitial lung disease classification to explore how the performance of a CNN changes according to architecture, data set characteristics, and transfer learning. They considered five deep CNNs, namely CifarNet (114), AlexNet (113), OverFeat (111), VGG-16 (115), and GoogLeNet (116), which achieved state-of-the-art performance in various computer vision applications. From their extensive experiments, these authors drew some interesting conclusions: (*a*) It was consistently beneficial for CADe problems to transfer features learned from the large-scale annotated natural image data sets (ImageNet), and (*b*) applications of off-the-shelf deep CNN features to CADe problems could be improved by exploring the performance-complementary properties of human-designed features.

Unlike the studies above, which used deterministic deep architectures, van Tulder & de Bruijne (35) exploited a deep generative model with a convolutional RBM as the basic building block for classification of interstitial lung disease. Specifically, they used a discriminative RBM with an additional label layer along with input and hidden layers to improve the discriminative power of learned feature representations. These experiments demonstrated the advantages of combining generative and discriminative learning objectives by achieving higher performance than that of purely generative or discriminative learning methods.

Pereira et al. (34) investigated brain tumor segmentation by using CNNs in MR images. They explored small-sized kernels in order to have fewer parameters but deeper architectures. They trained different CNN architectures for low- and high-grade tumors and validated their method in the 2013 Brain Tumor Segmentation (BRATS) Challenge,[7] where their technique ranked at the top for the complete, core, and enhancing regions for the challenge data set. Brosch et al. (49) applied deep learning for multiple sclerosis lesion segmentation on MR images. Their model was a 3D CNN composed of two interconnected pathways, namely a convolutional pathway that learned hierarchical feature representations similar to those of other CNNs and a deconvolutional pathway consisting of deconvolutional and unpooling layers with shortcut connections to the corresponding convolutional layers. The deconvolutional layers were designed to calculate abstract segmentation features from the features represented by each convolutional layer and the activation of the previous deconvolutional layer, if applicable. In comparison to five publicly available methods for multiple sclerosis lesion segmentation, this method achieved the best performance in terms of Dice similarity coefficient, absolution volume difference, and lesion false-positive rate.

An important limitation of typical deep CNNs arises from the fixed architecture of the models themselves. When an input observation is larger than the unit in the input layer, the straightforward solution is to apply a sliding-window strategy. However, it is computationally very expensive and time/memory consuming to do so. Because of this scalability issue in CNNs, Dou et al. (36) devised a 3D fully connected network by transforming units in the fully connected layers into a 3D ($1 \times 1 \times 1$) convolutionable kernel that enabled an arbitrary-sized input to be processed efficiently (101). The output of this 3D fully connected network could be remapped back onto the original input, making it possible to interpret the network output more intuitively. For detection of cerebral microbleeds in MR images, these authors designed a cascade framework. They first screened the input with

[7] For details, refer to **http://martinos.org/qtim/miccai2013/**.

the proposed 3D fully connected network to retrieve candidates with high probabilities of being cerebral microbleeds, and then applied a 3D CNN discrimination model for final detection. These experiments validated the effectiveness of the method by removing massive redundant computations and dramatically speeding up the detection process.

3.5. Deep Learning for Computer-Aided Diagnosis

CADx provides a second objective opinion regarding assessment of a disease from image-based information. The major applications of CADx involve the discrimination of malignant from benign lesions and the identification of certain diseases from one or more images. Conventionally, most CADx systems were developed to use human-designed features engineered by domain experts. Recently, deep learning methods have been successfully applied to CADx systems.

Cheng et al. (39) used an SAE with a denoising technique (SDAE) to differentiate breast ultrasound lesions and lung CT nodules. Specifically, the image regions of interest (ROIs) were first resized to 28 × 28, where all of the pixels in each patch were treated as the input to the SDAE. During the pretraining step, the authors corrupted the input patches with random noise to enhance the noise tolerance of their model. Later, during the fine-tuning step, they included the resized scale factors of the two ROI dimensions and the aspect ratios of the original ROIs to preserve the original information. Shen et al. (41) created a hierarchical learning framework with a multiscale CNN to capture various sizes of lung nodules. In this CNN architecture, three CNNs that took nodule patches from different scales as input were assembled in parallel. To reduce overfitting, the authors set the parameters of the three CNNs to be shared during training. The activations of the top hidden layer in three CNNs, one for each scale, were concatenated to form a feature vector. For classification, the authors used an SVM with a radial basis function kernel and random forest, which was trained to minimize so-called companion objectives, defined as the combination of overall hinge loss function and sum of the companion hinge loss functions (117).

Suk et al. (31) used an SAE to identify Alzheimer disease or mild cognitive impairment by fusing neuroimaging and biological features. They extracted GM volume features from MR images, regional mean intensity values from PET images, and three biological features ($A\beta_{42}$, p-tau, and t-tau) from CSF. After training modality-specific SAEs, for each modality they constructed an augmented feature vector by concatenating the original features with the outputs of the top hidden layer of the respective SAEs. A multikernel SVM (118) was trained for clinical decision making. The same authors extended their research to find hierarchical feature representations by combining heterogeneous modalities during feature representation learning, rather than in the classifier learning step (29). Specifically, they used a DBM to find a latent hierarchical feature representation from a 3D patch, and then devised a systematic method for a joint feature representation (**Figure 9*a***) from the paired patches of MRI and PET with a multimodal DBM. To enhance diagnostic performance, they also used a discriminative DBM by adding a discriminative RBM (119) on top of the highest hidden layer. That is, the top hidden layer was connected to both the lower hidden layer and the additional label layer that indicated the label of the input patches (**Figure 9*a***). Using this method, the authors trained a multimodal DBM to discover hierarchical and discriminative feature representations by integrating the process of discovering features of inputs with their use in classification. **Figure 9*b*,*c*** shows the learned connection weights from the MRI pathway and the PET pathway.

Plis et al. (120) applied a DBN to MR images and validated the feasibility of the application by investigating whether a building block of deep generative models was competitive with independent component analysis, the most widely used method for functional MRI (fMRI) analysis. They also examined the effect of the depth of deep models in the analysis of structural MR images of a

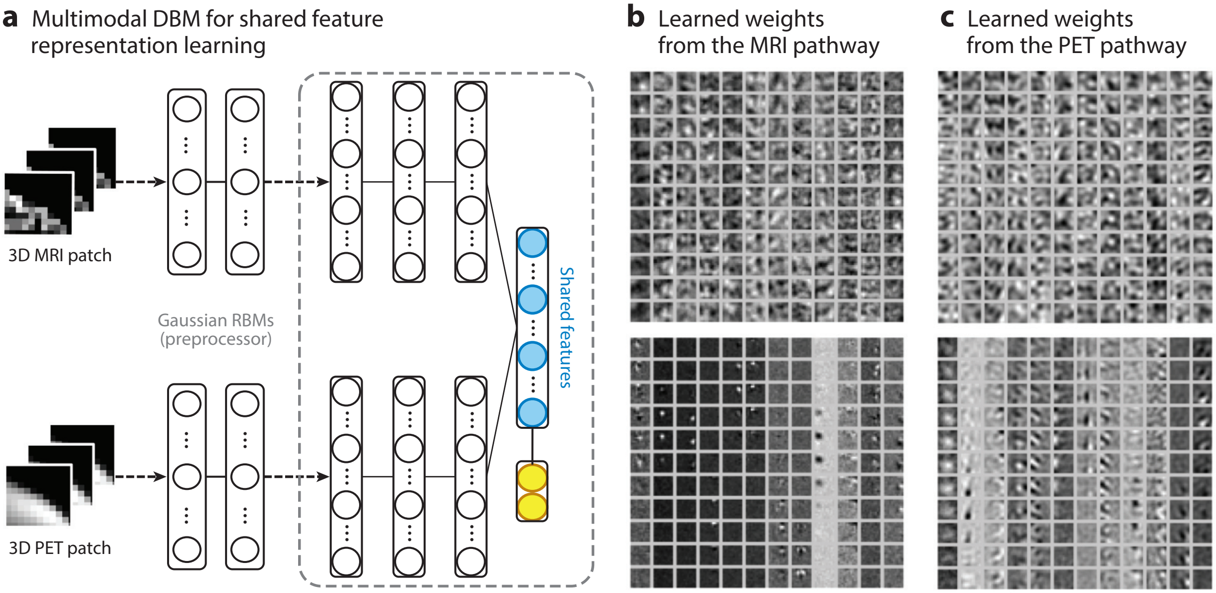
a Multimodal DBM for shared feature representation learning
3D MRI patch
3D PET patch
Gaussian RBMs (preprocessor)
Shared features
b Learned weights from the MRI pathway
c Learned weights from the PET pathway

Figure 9

(*a*) Shared feature learning from patches of different modalities, such as magnetic resonance imaging (MRI) and positron emission tomography (PET), with a discriminative multimodal deep Boltzmann machine (DBM). The yellow circles represent the input patches, and the blue circles show joint feature representation. (*b*,*c*) Visualization of the learned weights in Gaussian restricted Boltzmann machines (RBMs) (*bottom*) and those of the first hidden layer (*top*) from MRI and PET pathways in a multimodal DBM (29). Each column, with 11 patches in the upper block and the lower block, composes a three-dimensional patch.

schizophrenia data set and a Huntington disease data set. Inspired by the work of Plis et al., Kim et al. (121) and Suk et al. (33) independently studied applications of deep learning for fMRI-based brain disease diagnosis. Kim et al. used an SAE for whole-brain resting-state functional connectivity pattern representation for the diagnosis of schizophrenia and the identification of aberrant functional connectivity patterns associated with schizophrenia. They first computed Pearson's correlation coefficients between pairs of 116 regions on the basis of their regional mean blood oxygenation level–dependent (BOLD) signals. After performing Fisher's *r*-to-*z* transformation on the coefficients and Gaussian normalization sequentially, they fed the pseudo-*z*-scored levels into their SAE. More recently, Suk et al. (33) proposed a novel framework of fusing deep learning with a hidden Markov model (HMM) for functional dynamics estimation in resting-state fMRI and successfully used this framework for the diagnosis of mild cognitive impairment (MCI). Specifically, they devised a deep auto-encoder (DAE) by stacking multiple RBMs in order to discover hierarchical nonlinear functional relations among brain regions. **Figure 10** shows examples of the learned connection weights in the form of functional networks. This DAE was used to transform the regional mean BOLD signals into an embedding space, whose bases were understood as complex functional networks. After embedding functional signals, Suk et al. then used the HMM to estimate the dynamic characteristics of functional networks inherent in resting-state fMRI via internal states, which could be inferred statistically from observations. By building a generative model with an HMM, they estimated the likelihood of the input features of resting-state fMRI as belonging to the corresponding status (i.e., MCI or normal healthy control), then used this information to determine the clinical label of a test subject.

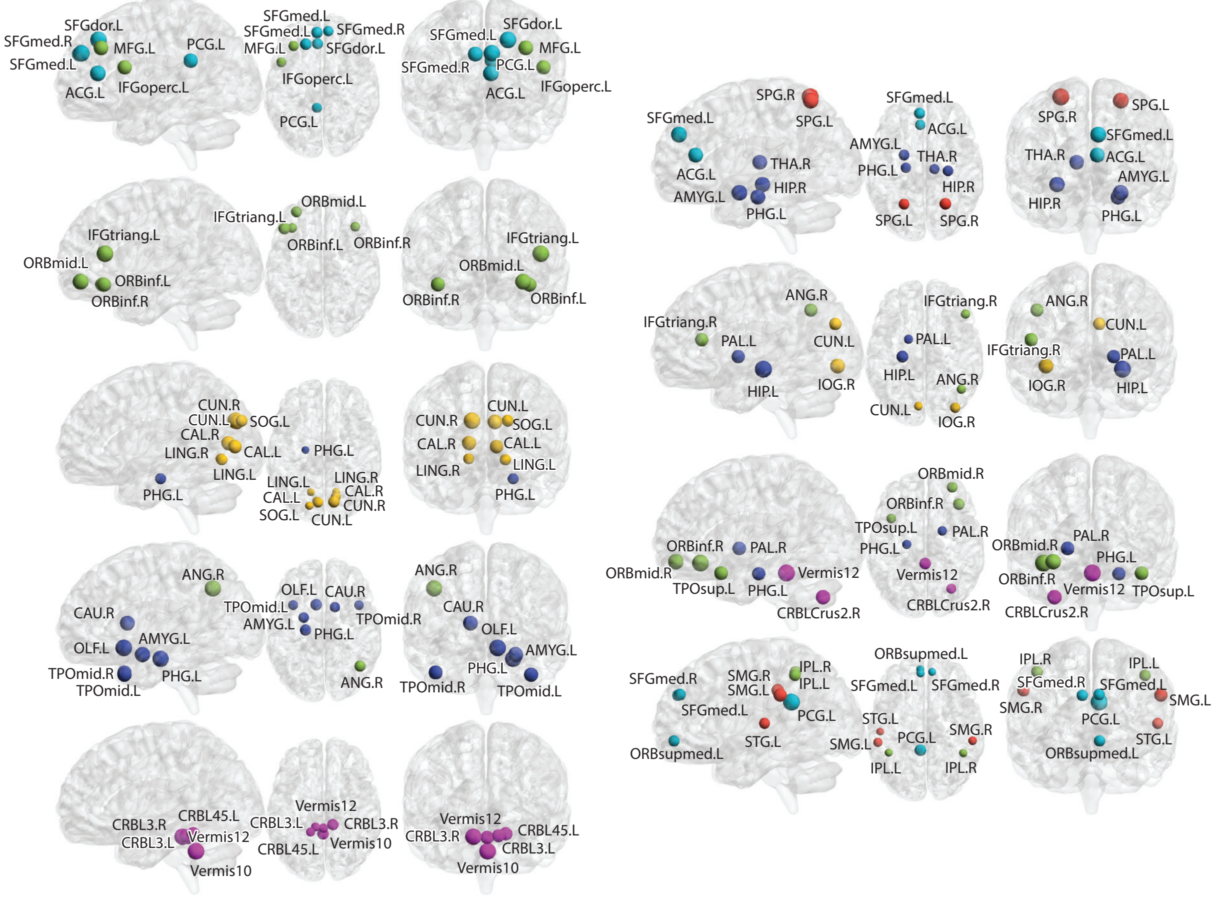

Figure 10

Functional networks learned from the first hidden layer of the deep auto-encoder from Reference 33. The functional networks in the left column correspond to (*from top to bottom*) the default-mode network, executive attention network, visual network, subcortical regions, and cerebellum. The functional networks in the right column show the relations among regions of different networks, cortices, and cerebellum.

Other studies have used CNNs to diagnose brain disease. Brosch et al. (47) performed manifold learning from downsampled MR images by using a deep generative model composed of three convolutional RBMs and two RBM layers. To speed up the calculation of convolutions, the computational bottleneck of the training algorithm, they performed training in the frequency domain. By generating volume samples from their deep generative model, they validated the effectiveness of deep learning for manifold embedding with no explicitly defined similarity measure or proximity graph. Li et al. (44) constructed a three-layer CNN with two convolutional layers and one fully connected layer. They proposed to use CNNs to integrate multimodal neuroimaging data by designing a 3D CNN architecture that received one volumetric MRI patch as input and another volumetric PET patch as output. When trained end to end on subjects with both data modalities, the network captured the nonlinear relationship between two modalities. These experiments demonstrated that PET data could be predicted and estimated, given the input MRI data, and the authors quantitatively evaluated the proposed data completion method by comparing the classification results according to the predicted and actual PET images.

4. CONCLUSION

Computational modeling for medical image analysis has had a significant impact on both clinical applications and scientific research. Recent progress in deep learning has shed new light on medical image analysis by enabling the discovery of morphological and/or textural patterns in images solely from data. Deep learning methods have achieved state-of-the-art performance across different medical applications; however, there is still room for improvement. First, as witnessed in computer vision, in which breakthrough improvements were achieved by use of large numbers of training data [e.g., more than one million annotated images in ImageNet (24)], a large, publicly available data set of medical images from which deep models can find more generalized features would lead to improved performance. Second, although data-driven feature representations, especially in an unsupervised manner, have helped enhance accuracy, it would be desirable to devise a new methodological architecture involving domain-specific knowledge. Third, it is necessary to develop algorithmic techniques to efficiently handle images acquired with different scanning protocols so that it would not be necessary to train modality-specific deep models. Finally, when using deep learning to investigate underlying patterns in images such as fMRI, because of the black box–like characteristics of deep models, it remains challenging to understand and interpret the learned models intuitively.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

D.S. and H.-I.S. contributed equally to the writing of this review, and both are first and corresponding authors. The writing of this review was supported by an Institute for Information and Communications Technology Promotion grant funded by the Korean government [B0101-16-0307, Basic Software Research in Human-Level Lifelong Machine Learning (Machine Learning Center)]; by the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2015R1C1A1A01052216); and by the US National Institutes of Health (grants EB006733, EB008374, EB009634, MH100217, MH108914, AG041721, AG049371, AG042599, and DE022676).

LITERATURE CITED

1. Brody H. 2013. Medical imaging. *Nature* 502:S81
2. Shao Y, Gao Y, Guo Y, Shi Y, Yang X, Shen D. 2014. Hierarchical lung field segmentation with joint shape and appearance sparse learning. *IEEE Trans. Med. Imaging* 33:1761–80
3. Wang L, Chen KC, Gao Y, Shi F, Liao S, et al. 2014. Automated bone segmentation from dental CBCT images using patch-based sparse representation and convex optimization. *Med. Phys.* 41:043503
4. Yap PH, Zhang Y, Shen D. 2016. Multi-tissue decomposition of diffusion MRI signals via L0 sparse-group estimation. *IEEE Trans. Image Process.* 25:4340–53
5. Suk HI, Lee SW, Shen D. 2016. Deep sparse multi-task learning for feature selection in Alzheimer's disease diagnosis. *Brain Struct. Funct.* 221:2569–87

6. Chen Y, Juttukonda M, Su Y, Benzinger T, Rubin BG, et al. 2015. Probabilistic air segmentation and sparse regression estimated pseudo CT for PET/MR attenuation correction. *Radiology* 275:562–69
7. Schmidhuber J. 2015. Deep learning in neural networks: an overview. *Neural Netw.* 61:85–117
8. Bengio Y. 2009. *Learning Deep Architectures for AI: Foundations and Trends in Machine Learning*. Boston: Now. 127 pp.
9. LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. *Nature* 521:436–44
10. Hinton GE, Salakhutdinov RR. 2006. Reducing the dimensionality of data with neural networks. *Science* 313:504–7
11. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* 11:3371–408
12. Nair V, Hinton GE. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 807–14. New York: ACM
13. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15:1929–58
14. Ioffe S, Szegedy C. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–56. New York: ACM
15. Bishop CM. 1995. *Neural Networks for Pattern Recognition*. Oxford, UK: Oxford Univ. Press
16. Collobert R, Weston J. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–67. New York: ACM
17. Sutskever I, Martens J, Hinton GE. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 1017–24. New York: ACM
18. Hinton GE, Deng L, Yu D, Dahl GE, Mohamed A, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Proc. Mag.* 29:82–97
19. Szegedy C, Toshev A, Erhan D. 2013. Deep neural networks for object detection. In *Proceedings of the 26th Neural Information Processing Systems Conference* (*NIPS 2013*), ed. CJC Burges, L Bottou, M Welling, Z Ghahramani, KQ Weinberger, pp. 2553–61. **https://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection**
20. Taigman Y, Yang M, Ranzato M, Wolf L. 2014. DeepFace: closing the gap to human-level performance in face verification. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–8. Washington, DC: IEEE
21. Zhang J, Zong C. 2015. Deep neural networks in machine translation: an overview. *IEEE Intell. Syst.* 30:16–25
22. Karpathy A, Li F. 2015. Deep visual–semantic alignments for generating image descriptions. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–37. Washington, DC: IEEE
23. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529:484–89
24. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, et al. 2015. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115:211–52
25. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. 2012. *The PASCAL Visual Object Classes Challenge 2012* (*VOC2012*) *results*. **http://host.robots.ox.ac.uk/pascal/VOC/voc2012/**
26. Zhang W, Li R, Deng H, Wang L, Lin W, et al. 2015. Deep convolutional neural networks for multimodality isointense infant brain image segmentation. *NeuroImage* 108:214–24
27. Kleesiek J, Urban G, Hubert A, Schwarz D, Maier-Hein K, et al. 2016. Deep MRI brain extraction: a 3D convolutional neural network for skull stripping. *NeuroImage* 129:460–69
28. Wu G, Kim M, Wang Q, Munsell BC, Shen D. 2016. Scalable high-performance image registration framework by unsupervised deep feature representations learning. *IEEE Trans. Biomed. Eng.* 63:1505–16
29. Suk HI, Lee SW, Shen D. 2014. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage* 101:569–82

30. Shin H, Roberts K, Lu L, Demner-Fushman D, Yao J, Summers RM. 2016. Learning to read chest X-rays: recurrent neural cascade model for automated image annotation. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2497–506. Washington, DC: IEEE
31. Suk HI, Lee SW, Shen D. 2015. Latent feature representation with stacked auto-encoder for AD/MCI diagnosis. *Brain Struct. Funct.* 220:841–59
32. Suk HI, Shen D. 2015. Deep learning in diagnosis of brain disorders. In *Recent Progress in Brain and Cognitive Engineering*, ed. SW Lee, HH Bülthoff, KR Müller, pp. 203–13. Berlin: Springer
33. Suk HI, Wee CY, Lee SW, Shen D. 2016. State-space model with deep learning for functional dynamics estimation in resting-state fMRI. *NeuroImage* 129:292–307
34. Pereira S, Pinto A, Alves V, Silva CA. 2016. Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Med. Imaging* 35:1240–51
35. van Tulder G, de Bruijne M. 2016. Combining generative and discriminative representation learning for lung CT analysis with convolutional restricted Boltzmann machines. *IEEE Trans. Med. Imaging* 35:1262–72
36. Dou Q, Chen H, Yu L, Zhao L, Qin J, et al. 2016. Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. *IEEE Trans. Med. Imaging* 35:1182–95
37. Cireşan DC, Giusti A, Gambardella LM, Schmidhuber J. 2013. Mitosis detection in breast cancer histological images with deep neural networks. In *Proceedings of the 2013 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 411–18. Berlin: Springer
38. Chen H, Dou Q, Wang X, Qin J, Heng PA. 2016. Mitosis detection in breast cancer histology images via deep cascaded networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 1167–73. Palo Alto, CA: AAAI
39. Cheng JZ, Ni D, Chou YH, Qin J, Tiu CM, et al. 2016. Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans. *Sci. Rep.* 6:24454
40. Roth HR, Lu L, Liu J, Yao J, Seff A, et al. 2016. Improving computer-aided detection using convolutional neural networks and random view aggregation. *IEEE Trans. Med. Imaging* 35:1170–81
41. Shen W, Zhou M, Yang F, Yang C, Tian J. 2015. Multi-scale convolutional neural networks for lung nodule classification. In *Lecture Notes in Computer Science*, vol. 9123: *Information Processing in Medical Imaging*, pp. 588–99. Berlin: Springer
42. Setio AAA, Ciompi F, Litjens G, Gerke P, Jacobs C, et al. 2016. Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. *IEEE Trans. Med. Imaging* 35:1160–69
43. Ciompi F, de Hoop B, van Riel SJ, Chung K, Scholten ET, et al. 2015. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. *Med. Image Anal.* 26:195–202
44. Li R, Zhang W, Suk HI, Wang L, Li J, et al. 2014. Deep learning based imaging data completion for improved brain disease diagnosis. In *Proceedings of the 2014 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 305–12. Berlin: Springer
45. Shin HC, Roth HR, Gao M, Lu L, Xu Z, et al. 2016. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* 35:1285–98
46. Gupta A, Ayhan M, Maida A. 2013. Natural image bases to represent neuroimaging data. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 987–94. New York: ACM
47. Brosch T, Tam R. 2013. Manifold learning of brain MRIs by deep learning. In *Proceedings of the 2013 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 633–40. Berlin: Springer
48. Nie D, Wang L, Gao Y, Shen D. 2016. Fully convolutional networks for multi-modality isointense infant brain image segmentation. In *Proceedings of the 13th IEEE International Symposium on Biomedical Imaging*, pp. 1342–45. Washington, DC: IEEE
49. Brosch T, Tang LYW, Yoo Y, Li DKB, Traboulsee A, Tam R. 2016. Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation. *IEEE Trans. Med. Imaging* 35:1229–39

50. Chen H, Dou Q, Wang X, Qin J, Heng P. 2016. Mitosis detection in breast cancer histological images via deep cascaded networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 1160–66. Palo Alto, CA: AAAI
51. Shin HC, Orton MR, Collins DJ, Doran SJ, Leach MO. 2013. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data. *IEEE Trans. Pattern Anal. Mach. Intell.* 35:1930–43
52. Wu G, Kim M, Wang Q, Gao Y, Liao S, Shen D. 2013. Unsupervised deep feature learning for deformable registration of MR brain images. In *Proceedings of the 2013 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 649–56. Berlin: Springer
53. Su H, Xing F, Kong X, Xie Y, Zhang S, Yang L. 2015. Robust cell detection and segmentation in histopathological images using sparse reconstruction and stacked denoising autoencoders. In *Proceedings of the 2015 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 383–90. Berlin: Springer
54. Xu J, Xiang L, Liu Q, Gilmore H, Wu J, et al. 2016. Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. *IEEE Trans. Med. Imaging* 35:119–30
55. Salakhutdinov R. 2015. Learning deep generative models. *Annu. Rev. Stat. Appl.* 2:361–85
56. Munsell BC, Wee CY, Keller SS, Weber B, Elger C, et al. 2015. Evaluation of machine learning algorithms for treatment outcome prediction in patients with epilepsy based on structural connectome data. *NeuroImage* 118:219–30
57. Maier O, Schrder C, Forkert ND, Martinetz T, Handels H. 2015. Classifiers for ischemic stroke lesion segmentation: a comparison study. *PLOS ONE* 10:1–16
58. Havaei M, Davy A, Warde-Farley D, Biard A, Courville A, et al. 2017. Brain tumor segmentation with deep neural networks. *Med. Image Anal.* 35:18–31
59. Ronneberger O, Fischer P, Brox T. 2015. U-net: convolutional networks for biomedical image segmentation. In *Proceedings of the 2015 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 234–41. Berlin: Springer
60. Fakhry A, Peng H, Ji S. 2016. Deep models for brain EM image segmentation: novel insights and improved performance. *Bioinformatics* 32:2352–58
61. Farag A, Lu L, Roth HR, Liu J, Turkbey E, Summers RM. 2015. A bottom-up approach for pancreas segmentation using cascaded superpixels and (deep) image patch labeling. arXiv:1505.06236 [cs.CV]
62. Ghesu FC, Krubasik E, Georgescu B, Singh V, Zheng Y, et al. 2016. Marginal space deep learning: efficient architecture for volumetric image parsing. *IEEE Trans. Med. Imaging* 35:1217–28
63. Wang CW, Huang CT, Lee JH, Li CH, Chang SW, et al. 2016. A benchmark for comparison of dental radiography analysis algorithms. *Med. Image Anal.* 31:63–76
64. Rosenblatt F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 1958:65–386
65. Rumelhart DE, Hinton GE, Williams RJ. 1986. Learning representations by back-propagating errors. *Nature* 323:533–36
66. Le QV, Ngiam J, Coates A, Lahiri A, Prochnow B, Ng AY. 2011. On optimization methods for deep learning. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 265–72. New York: ACM
67. Hornik K. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* 4:251–57
68. Schwarz G. 1978. Estimating the dimension of a model. *Ann. Stat.* 6:461–64
69. Bourlard H, Kamp Y. 1988. Auto-association by multilayer perceptrons and singular value decomposition. *Biol. Cybern.* 59:291–94
70. Bengio Y, Lamblin P, Popovici D, Larochelle H. 2007. Greedy layer-wise training of deep networks. In *Proceedings of the 19th Conference on Neural Information Processing Systems* (*NIPS 2006*), ed. B Schölkopf, JC Platt, T Hoffmann, pp. 153–60. **https://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks**
71. Larochelle H, Bengio Y, Louradour J, Lamblin P. 2009. Exploring strategies for training deep neural networks. *J. Mach. Learn. Res.* 10:1–40
72. Hinton GE, Osindero S, Teh YW. 2006. A fast learning algorithm for deep belief nets. *Neural Comput.* 18:1527–54

73. Smolensky P. 1986. Information processing in dynamical systems: foundations of harmony theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. pp. 194–281. Cambridge, MA: MIT Press
74. Hinton GE. 2002. Training products of experts by minimizing contrastive divergence. *Neural Comput.* 14:1771–800
75. Hinton G, Dayan P, Frey B, Neal R. 1995. The "wake–sleep" algorithm for unsupervised neural networks. *Science* 268:1158–61
76. Lecun Y, Bottou L, Bengio Y, Haffner P. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86:2278–324
77. Glorot X, Bengio Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–56. Brookline, MA: Microtome
78. Sutskever I, Martens J, Dahl GE, Hinton GE. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 1139–47. New York: ACM
79. Glorot X, Bordes A, Bengio Y. 2011. Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, ed. G Gordon, D Dunson, M Dudik, pp. 315–23. Brookline, MA: Microtome
80. Maas AL, Hannun AY, Ng AY. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning, Workshop on Deep Learning for Audio, Speech, and Language Processing*, p. 192. New York: ACM
81. Wan L, Zeiler MD, Zhang S, LeCun Y, Fergus R. 2013. Regularization of neural networks using DropConnect. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1056–66. New York: ACM
82. Cho ZH, Kim YB, Han JY, Min HK, Kim KN, et al. 2008. New brain atlas—mapping the human brain in vivo with 7.0 T MRI and comparison with postmortem histology: Will these images change modern medicine? *Int. J. Imaging Syst. Technol.* 18:2–8
83. Wu G, Qi F, Shen D. 2006. Learning-based deformable registration of MR brain images. *IEEE Trans. Med. Imaging* 25:1145–57
84. Ou Y, Sotiras A, Paragios N, Davatzikos C. 2011. DRAMMS: deformable registration via attribute matching and mutual-saliency weighting. *Med. Image Anal.* 15:622–39
85. Sotiras A, Davatzikos C, Paragios N. 2013. Deformable medical image registration: a survey. *IEEE Trans. Med. Imaging* 32:1153–90
86. Lowe DG. 1999. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*. 8 pp. **http://www.cs.ubc.ca/~lowe/papers/iccv99.pdf**
87. Vercauteren T, Pennec X, Perchant A, Ayache N. 2009. Diffeomorphic demons: efficient non-parametric image registration. *NeuroImage* 45:S61–72
88. Wu G, Kim M, Wang Q, Shen D. 2014. S-HAMMER: hierarchical attribute-guided, symmetric diffeomorphic registration for MR brain images. *Hum. Brain Mapp.* 35:1044–60
89. Liao S, Gao Y, Oto A, Shen D. 2013. Representation learning: a unified deep learning framework for automatic prostate MR segmentation. In *Proceedings of the 2013 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 254–61. Berlin: Springer
90. Guo Y, Gao Y, Shen D. 2016. Deformable MR prostate segmentation via deep feature learning and sparse patch matching. *IEEE Trans. Med. Imaging* 35:1077–89
91. Liao S, Gao Y, Shi Y, Yousuf A, Karademir I, et al. 2013. Automatic prostate MR image segmentation with sparse label propagation and domain-specific manifold regularization. *Inf. Proc. Med. Imaging* 23:511–23
92. Kim M, Wu G, Shen D. 2013. Unsupervised deep learning for hippocampus segmentation in 7.0 Tesla MR images. In *Lecture Notes in Computer Science*, vol. 8184: *Machine Learning in Medical Imaging*, pp. 1–8. Berlin: Springer
93. Roth HR, Lee CT, Shin HC, Seff A, Kim L, et al. 2015. Anatomy-specific classification of medical images using deep convolutional nets. In *Proceedings of the IEEE 12th International Symposium on Biomedical Imaging*, pp. 293–303. Washington, DC: IEEE

94. Yan Z, Zhan Y, Peng Z, Liao S, Shinagawa Y, et al. 2015. Bodypart recognition using multi-stage deep learning. In *Proceedings of the 24th Conference on Information Processing in Medical Imaging*, pp. 449–61. New York: ACM

95. Yan Z, Zhan Y, Peng Z, Liao S, Shinagawa Y, et al. 2016. Multi-instance deep learning: Discover discriminative local anatomies for bodypart recognition. *IEEE Trans. Med. Imaging* 35:1332–43

96. Maron O, Lozano-Pérez T. 1998. A framework for multiple-instance learning. In *Proceedings of Neural Information Processing Systems* (*NIPS 1998*), pp. 570–76. **https://papers.nips.cc/paper/1346-a-framework-for-multiple-instance-learning**

97. Liu F, Yang L. 2015. A novel cell detection method using deep convolutional neural network and maximum-weight independent set. In *Proceedings of the 2015 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 349–57. Berlin: Springer

98. Xie Y, Xing F, Kong X, Su H, Yang L. 2015. Beyond classification: structured regression for robust cell detection using convolutional neural network. In *Proceedings of the 2015 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 358–65. Berlin: Springer

99. Xie Y, Kong X, Xing F, Liu F, Su H, Yang L. 2015. Deep voting: a robust approach toward nucleus localization in microscopy images. In *Proceedings of the 2015 Medical Image Computing and Computer-Assisted Intervention Conference*, pp. 374–82. Berlin: Springer

100. Sirinukunwattana K, Raza SEA, Tsang YW, Snead DRJ, Cree IA, Rajpoot NM. 2016. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Trans. Med. Imaging* 35:1196–206

101. Long J, Shelhamer E, Darrell T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 371–80. Washington, DC: IEEE

102. Moeskops P, Viergever MA, Mendrik AM, de Vries LS, Benders MJNL, Išgum I. 2016. Automatic segmentation of MR brain images with a convolutional neural network. *IEEE Trans. Med. Imaging* 35:1252–61

103. Weisenfeld NI, Warfield SK. 2009. Automatic segmentation of newborn brain MRI. *NeuroImage* 47:564–72

104. Xue H, Srinivasan L, Jiang S, Rutherford M, Edwards AD, et al. 2007. Automatic segmentation and reconstruction of the cortex from neonatal MRI. *NeuroImage* 38:461–77

105. Gui L, Lisowski R, Faundez T, Hüppi PS, Lazeyras F, Kocher M. 2012. Morphology-driven automatic segmentation of MR images of the neonatal brain. *Med. Image Anal.* 16:1565–79

106. Warfield S, Kaus M, Jolesz FA, Kikinis R. 2000. Adaptive, template moderated, spatially varying statistical classification. *Med. Image Anal.* 4:43–55

107. Prastawa M, Gilmore JH, Lin W, Gerig G. 2005. Automatic segmentation of MR images of the developing newborn brain. *Med. Image Anal.* 9:457–66

108. Wang L, Shi F, Lin W, Gilmore JH, Shen D. 2011. Automatic segmentation of neonatal images using convex optimization and coupled level sets. *NeuroImage* 58:805–17

109. Wang L, Shi F, Li G, Gao Y, Lin W, et al. 2014. Segmentation of neonatal brain MR images using patch-driven level sets. *NeuroImage* 84:141–58

110. Wang L, Gao Y, Shi F, Li G, Gilmore JH, et al. 2015. Links: learning-based multi-source integration framework for segmentation of infant brain images. *NeuroImage* 108:160–72

111. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. 2013. OverFeat: integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229 [cs.CV]

112. Gao M, Bagci U, Lu L, Wu A, Buty M, et al. 2016. Holistic classification of CT attenuation patterns for interstitial lung diseases via deep convolutional neural networks. *Comput. Methods Biomech. Biomed. Eng.* 2016:1–6

113. Krizhevsky A, Sutskever I, Hinton GE. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of Neural Information Processing Systems* (*NIPS 2012*), pp. 1097–105. **https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf**

114. Krizhevsky A. 2009. *Learning multiple layers of features from tiny images*. Tech. rep., Dep. Comput. Sci., Univ. Toronto, Can.

115. Simonyan K, Zisserman A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs.CV]

116. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, et al. 2015. Going deeper with convolutions. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9. Washington, DC: IEEE

117. Lee CY, Xie S, Gallagher PW, Zhang Z, Tu Z. 2015. Deeply-supervised nets. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pp. 562–70. Brookline, MA: Microtome

118. Gönen M, Alpaydın E. 2011. Multiple kernel learning algorithms. *J. Mach. Learn. Res.* 12:2211–68

119. Larochelle H, Bengio Y. 2008. Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 536–43. New York: ACM

120. Plis SM, Hjelm D, Salakhutdinov R, Allen EA, Bockholt HJ, et al. 2014. Deep learning for neuroimaging: a validation study. *Front. Neurosci.* 8:229

121. Kim J, Calhoun VD, Shim E, Lee JH. 2016. Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: evidence from whole-brain resting-state functional connectivity patterns of schizophrenia. *NeuroImage* 124:127–46