Hindawi
Computational Intelligence and Neuroscience
Volume 2018, Article ID 7068349, 13 pages
https://doi.org/10.1155/2018/7068349

Review Article

Deep Learning for Computer Vision: A Brief Review

Athanasios Voulodimos,[1,2] **Nikolaos Doulamis**,[2] **Anastasios Doulamis**,[2] **and Eftychios Protopapadakis**[2]

[1]*Department of Informatics, Technological Educational Institute of Athens, 12210 Athens, Greece*
[2]*National Technical University of Athens, 15780 Athens, Greece*

Correspondence should be addressed to Athanasios Voulodimos; thanosv@mail.ntua.gr

Received 17 June 2017; Accepted 27 November 2017; Published 1 February 2018

Academic Editor: Diego Andina

Over the last years deep learning methods have been shown to outperform previous state-of-the-art machine learning techniques in several fields, with computer vision being one of the most prominent cases. This review paper provides a brief overview of some of the most significant deep learning schemes used in computer vision problems, that is, Convolutional Neural Networks, Deep Boltzmann Machines and Deep Belief Networks, and Stacked Denoising Autoencoders. A brief account of their history, structure, advantages, and limitations is given, followed by a description of their applications in various computer vision tasks, such as object detection, face recognition, action and activity recognition, and human pose estimation. Finally, a brief overview is given of future directions in designing deep learning schemes for computer vision problems and the challenges involved therein.

1. Introduction

Deep learning allows computational models of multiple processing layers to learn and represent data with multiple levels of abstraction mimicking how the brain perceives and understands multimodal information, thus implicitly capturing intricate structures of large-scale data. Deep learning is a rich family of methods, encompassing neural networks, hierarchical probabilistic models, and a variety of unsupervised and supervised feature learning algorithms. The recent surge of interest in deep learning methods is due to the fact that they have been shown to outperform previous state-of-the-art techniques in several tasks, as well as the abundance of complex data from different sources (e.g., visual, audio, medical, social, and sensor).

The ambition to create a system that simulates the human brain fueled the initial development of neural networks. In 1943, McCulloch and Pitts [1] tried to understand how the brain could produce highly complex patterns by using interconnected basic cells, called neurons. The McCulloch and Pitts model of a neuron, called a MCP model, has made an important contribution to the development of artificial neural networks. A series of major contributions in the field is presented in Table 1, including LeNet [2] and Long Short-Term Memory [3], leading up to today's "era of deep learning." One of the most substantial breakthroughs in deep learning came in 2006, when Hinton et al. [4] introduced the Deep Belief Network, with multiple layers of Restricted Boltzmann Machines, greedily training one layer at a time in an unsupervised way. Guiding the training of intermediate levels of representation using unsupervised learning, performed locally at each level, was the main principle behind a series of developments that brought about the last decade's surge in deep architectures and deep learning algorithms.

Among the most prominent factors that contributed to the huge boost of deep learning are the appearance of large, high-quality, publicly available labelled datasets, along with the empowerment of parallel GPU computing, which enabled the transition from CPU-based to GPU-based training thus allowing for significant acceleration in deep models' training. Additional factors may have played a lesser role as well, such as the alleviation of the vanishing gradient problem owing to the disengagement from saturating activation functions (such as hyperbolic tangent and the logistic function), the proposal

Table 1: Important milestones in the history of neural networks and machine learning, leading up to the era of deep learning.

Milestone/contribution	Contributor, year
MCP model, regarded as the ancestor of the Artificial Neural Network	McCulloch & Pitts, 1943
Hebbian learning rule	Hebb, 1949
First perceptron	Rosenblatt, 1958
Backpropagation	Werbos, 1974
Neocognitron, regarded as the ancestor of the Convolutional Neural Network	Fukushima, 1980
Boltzmann Machine	Ackley, Hinton & Sejnowski, 1985
Restricted Boltzmann Machine (initially known as Harmonium)	Smolensky, 1986
Recurrent Neural Network	Jordan, 1986
Autoencoders	Rumelhart, Hinton & Williams, 1986 Ballard, 1987
LeNet, starting the era of Convolutional Neural Networks	LeCun, 1990
LSTM	Hochreiter & Schmidhuber, 1997
Deep Belief Network, ushering the "age of deep learning"	Hinton, 2006
Deep Boltzmann Machine	Salakhutdinov & Hinton, 2009
AlexNet, starting the age of CNN used for ImageNet classification	Krizhevsky, Sutskever, & Hinton, 2012

of new regularization techniques (e.g., dropout, batch normalization, and data augmentation), and the appearance of powerful frameworks like TensorFlow [5], theano [6], and mxnet [7], which allow for faster prototyping.

Deep learning has fueled great strides in a variety of computer vision problems, such as object detection (e.g., [8, 9]), motion tracking (e.g., [10, 11]), action recognition (e.g., [12, 13]), human pose estimation (e.g., [14, 15]), and semantic segmentation (e.g., [16, 17]). In this overview, we will concisely review the main developments in deep learning architectures and algorithms for computer vision applications. In this context, we will focus on three of the most important types of deep learning models with respect to their applicability in visual understanding, that is, Convolutional Neural Networks (CNNs), the "Boltzmann family" including Deep Belief Networks (DBNs) and Deep Boltzmann Machines (DBMs) and Stacked (Denoising) Autoencoders. Needless to say, the current coverage is by no means exhaustive; for example, Long Short-Term Memory (LSTM), in the category of Recurrent Neural Networks, although of great significance as a deep learning scheme, is not presented in this review, since it is predominantly applied in problems such as language modeling, text classification, handwriting recognition, machine translation, speech/music recognition, and less so in computer vision problems. The overview is intended to be useful to computer vision and multimedia analysis researchers, as well as to general machine learning researchers, who are interested in the state of the art in deep learning for computer vision tasks, such as object detection and recognition, face recognition, action/activity recognition, and human pose estimation.

The remainder of this paper is organized as follows. In Section 2, the three aforementioned groups of deep learning model are reviewed: Convolutional Neural Networks, Deep Belief Networks and Deep Boltzmann Machines, and Stacked Autoencoders. The basic architectures, training processes, recent developments, advantages, and limitations of each group are presented. In Section 3, we describe the contribution of deep learning algorithms to key computer vision tasks, such as object detection and recognition, face recognition, action/activity recognition, and human pose estimation; we also provide a list of important datasets and resources for benchmarking and validation of deep learning algorithms. Finally, Section 4 concludes the paper with a summary of findings.

2. Deep Learning Methods and Developments

2.1. Convolutional Neural Networks. Convolutional Neural Networks (CNNs) were inspired by the visual system's structure, and in particular by the models of it proposed in [18]. The first computational models based on these local connectivities between neurons and on hierarchically organized transformations of the image are found in Neocognitron [19], which describes that when neurons with the same parameters are applied on patches of the previous layer at different locations, a form of translational invariance is acquired. Yann LeCun and his collaborators later designed Convolutional Neural Networks employing the error gradient and attaining very good results in a variety of pattern recognition tasks [20–22].

A CNN comprises three main types of neural layers, namely, (i) convolutional layers, (ii) pooling layers, and (iii) fully connected layers. Each type of layer plays a different role. Figure 1 shows a CNN architecture for an object detection in image task. Every layer of a CNN transforms the input volume to an output volume of neuron activation, eventually leading to the final fully connected layers, resulting in a mapping of the input data to a 1D feature vector. CNNs have been extremely successful in computer vision applications, such as face recognition, object detection, powering vision in robotics, and self-driving cars.

(i) Convolutional Layers. In the convolutional layers, a CNN utilizes various kernels to convolve the whole image as

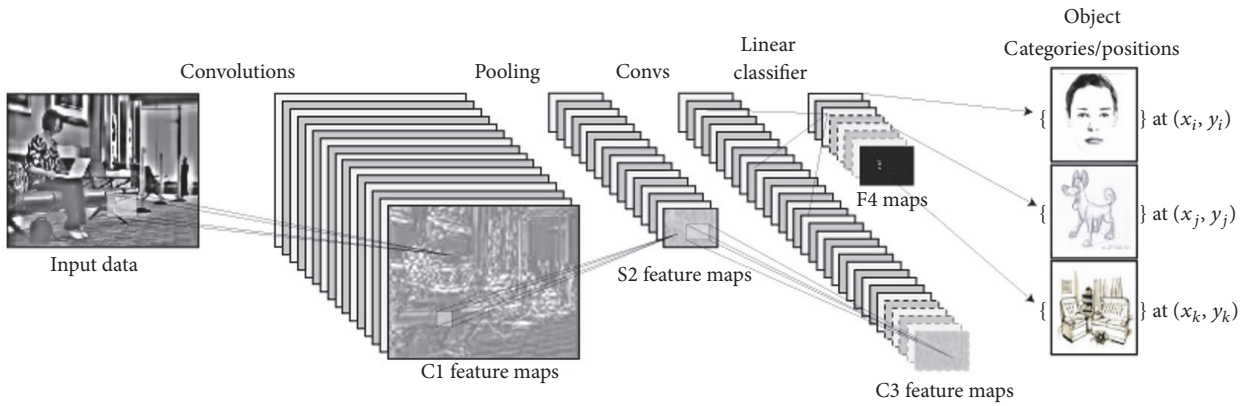


Figure 1: Example architecture of a CNN for a computer vision task (object detection).

well as the intermediate feature maps, generating various feature maps. Because of the advantages of the convolution operation, several works (e.g., [23, 24]) have proposed it as a substitute for fully connected layers with a view to attaining faster learning times.

(ii) Pooling Layers. Pooling layers are in charge of reducing the spatial dimensions (width × height) of the input volume for the next convolutional layer. The pooling layer does not affect the depth dimension of the volume. The operation performed by this layer is also called subsampling or downsampling, as the reduction of size leads to a simultaneous loss of information. However, such a loss is beneficial for the network because the decrease in size leads to less computational overhead for the upcoming layers of the network, and also it works against overfitting. Average pooling and max pooling are the most commonly used strategies. In [25] a detailed theoretical analysis of max pooling and average pooling performances is given, whereas in [26] it was shown that max pooling can lead to faster convergence, select superior invariant features, and improve generalization. Also there are a number of other variations of the pooling layer in the literature, each inspired by different motivations and serving distinct needs, for example, stochastic pooling [27], spatial pyramid pooling [28, 29], and def-pooling [30].

(iii) Fully Connected Layers. Following several convolutional and pooling layers, the high-level reasoning in the neural network is performed via fully connected layers. Neurons in a fully connected layer have full connections to all activation in the previous layer, as their name implies. Their activation can hence be computed with a matrix multiplication followed by a bias offset. Fully connected layers eventually convert the 2D feature maps into a 1D feature vector. The derived vector either could be fed forward into a certain number of categories for classification [31] or could be considered as a feature vector for further processing [32].

The architecture of CNNs employs three concrete ideas: (a) local receptive fields, (b) tied weights, and (c) spatial subsampling. Based on local receptive field, each unit in a convolutional layer receives inputs from a set of neighboring units belonging to the previous layer. This way neurons are capable of extracting elementary visual features such as edges or corners. These features are then combined by the subsequent convolutional layers in order to detect higher order features. Furthermore, the idea that elementary feature detectors, which are useful on a part of an image, are likely to be useful across the entire image is implemented by the concept of tied weights. The concept of tied weights constraints a set of units to have identical weights. Concretely, the units of a convolutional layer are organized in planes. All units of a plane share the same set of weights. Thus, each plane is responsible for constructing a specific feature. The outputs of planes are called feature maps. Each convolutional layer consists of several planes, so that multiple feature maps can be constructed at each location.

During the construction of a feature map, the entire image is scanned by a unit whose states are stored at corresponding locations in the feature map. This construction is equivalent to a convolution operation, followed by an additive bias term and sigmoid function:

$$\mathbf{y}^{(d)} = \sigma\left(\mathbf{W}\mathbf{y}^{(d-1)} + \mathbf{b}\right), \tag{1}$$

where d stands for the depth of the convolutional layer, $\mathbf{W}$ is the weight matrix, and $\mathbf{b}$ is the bias term. For fully connected neural networks, the weight matrix is full, that is, connects every input to every unit with different weights. For CNNs, the weight matrix $\mathbf{W}$ is very sparse due to the concept of tied weights. Thus, $\mathbf{W}$ has the form of

$$\begin{bmatrix} \mathbf{w} & 0 & \cdots & 0 \\ 0 & \mathbf{w} & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{w} \end{bmatrix}, \tag{2}$$

where $\mathbf{w}$ are matrices having the same dimensions with the units' receptive fields. Employing a sparse weight matrix reduces the number of network's tunable parameters and thus increases its generalization ability. Multiplying $\mathbf{W}$ with layer inputs is like convolving the input with $\mathbf{w}$, which can be seen as a trainable filter. If the input to $d-1$ convolutional layer is of

dimension $N \times N$ and the receptive field of units at a specific plane of convolutional layer d is of dimension $m \times m$, then the constructed feature map will be a matrix of dimensions $(N - m + 1) \times (N - m + 1)$. Specifically, the element of feature map at (i, j) location will be

$$\mathbf{y}_{ij}^{(d)} = \sigma\left(x_{ij}^{(d)} + b\right) \tag{3}$$

with

$$x_{ij}^{(d)} = \sum_{\alpha=0}^{m-1} \sum_{b=0}^{m-1} w_{\alpha b} \mathbf{y}_{(i+\alpha)(j+b)}^{(d-1)}, \tag{4}$$

where the bias term b is scalar. Using (4) and (3) sequentially for all (i, j) positions of input, the feature map for the corresponding plane is constructed.

One of the difficulties that may arise with training of CNNs has to do with the large number of parameters that have to be learned, which may lead to the problem of overfitting. To this end, techniques such as stochastic pooling, dropout, and data augmentation have been proposed. Furthermore, CNNs are often subjected to pretraining, that is, to a process that initializes the network with pretrained parameters instead of randomly set ones. Pretraining can accelerate the learning process and also enhance the generalization capability of the network.

Overall, CNNs were shown to significantly outperform traditional machine learning approaches in a wide range of computer vision and pattern recognition tasks [33], examples of which will be presented in Section 3. Their exceptional performance combined with the relative easiness in training are the main reasons that explain the great surge in their popularity over the last few years.

2.2. Deep Belief Networks and Deep Boltzmann Machines. Deep Belief Networks and Deep Boltzmann Machines are deep learning models that belong in the "Boltzmann family," in the sense that they utilize the Restricted Boltzmann Machine (RBM) as learning module. The Restricted Boltzmann Machine (RBM) is a generative stochastic neural network. DBNs have undirected connections at the top two layers which form an RBM and directed connections to the lower layers. DBMs have undirected connections between all layers of the network. A graphic depiction of DBNs and DBMs can be found in Figure 2. In the following subsections, we will describe the basic characteristics of DBNs and DBMs, after presenting their basic building block, the RBM.

2.2.1. Restricted Boltzmann Machines. A Restricted Boltzmann Machine ([34, 35]) is an undirected graphical model with stochastic visible variables $\mathbf{v} \in \{0, 1\}^D$ and stochastic hidden variables $\mathbf{h} \in \{0, 1\}^F$, where each visible variable is connected to each hidden variable. An RBM is a variant of the Boltzmann Machine, with the restriction that the visible units and hidden units must form a bipartite graph. This restriction allows for more efficient training algorithms, in particular the gradient-based contrastive divergence algorithm [36].

The model defines the energy function E: $\{0, 1\}^D \times \{0, 1\}^F \to \mathbb{R}$:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i=1}^{D} \sum_{j=1}^{F} W_{ij} v_i h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} \alpha_j h_j, \tag{5}$$

where $\theta = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$ are the model parameters; that is, W_{ij} represents the symmetric interaction term between visible unit i and hidden unit j, and b_i, a_j are bias terms.

The joint distribution over the visible and hidden units is given by

$$P(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right), \qquad \mathcal{Z}(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right), \tag{6}$$

where $\mathcal{Z}(\theta)$ is the normalizing constant. The conditional distributions over hidden $\mathbf{h}$ and visible $\mathbf{v}$ vectors can be derived by (5) and (6) as

$$P(\mathbf{h} \mid \mathbf{v}; \theta) = \prod_{j=1}^{F} p\left(h_j \mid \mathbf{v}\right), \qquad P(\mathbf{v} \mid \mathbf{h}; \theta) = \prod_{i=1}^{D} p\left(v_i \mid \mathbf{h}\right). \tag{7}$$

Given a set of observations $\{\mathbf{v}_n\}_{n=1}^N$ the derivative of the log-likelihood with respect to the model parameters can be derived by (6) as

$$\frac{1}{N} \sum_{n=1}^{N} \frac{\partial \log P(\mathbf{v}_n; \theta)}{\partial W_{ij}} = \mathbb{E}_{P_{\text{data}}}\left[v_i h_j\right] - \mathbb{E}_{P_{\text{model}}}\left[v_i h_j\right], \tag{8}$$

where $\mathbb{E}_{P_{\text{data}}}$ denotes an expectation with respect to the data distribution $P_{\text{data}}(\mathbf{h}, \mathbf{v}; \theta) = P(\mathbf{h} \mid \mathbf{v}; \theta) P_{\text{data}}(\mathbf{v})$, with $P_{\text{data}}(\mathbf{v}) = (1/N) \sum_n \delta(\mathbf{v} - \mathbf{v}_\mathbf{n})$ representing the empirical distribution and $\mathbb{E}_{P\text{model}}$ is an expectation with respect to the distribution defined by the model, as in (6).

A detailed explanation along with the description of a practical way to train RBMs was given in [37], whereas [38] discusses the main difficulties of training RBMs and their underlying reasons and proposes a new algorithm with an adaptive learning rate and an enhanced gradient, so as to address the aforementioned difficulties.

2.2.2. Deep Belief Networks. Deep Belief Networks (DBNs) are probabilistic generative models which provide a joint probability distribution over observable data and labels. They are formed by stacking RBMs and training them in a greedy manner, as was proposed in [39]. A DBN initially employs an efficient layer-by-layer greedy learning strategy to initialize the deep network, and, in the sequel, fine-tunes all weights jointly with the desired outputs. DBNs are graphical models which learn to extract a deep hierarchical representation of

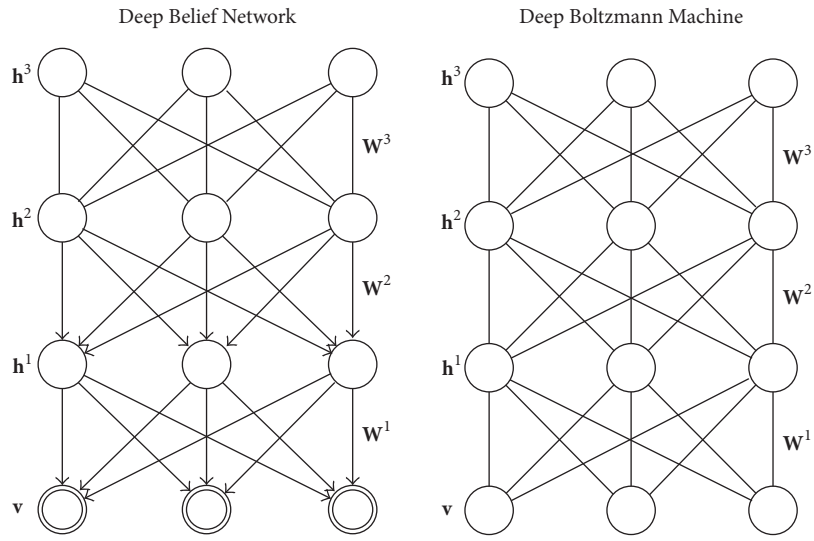


Figure 2: Deep Belief Network (DBN) and Deep Boltzmann Machine (DBM). The top two layers of a DBN form an undirected graph and the remaining layers form a belief network with directed, top-down connections. In a DBM, all connections are undirected.

the training data. They model the joint distribution between observed vector $\mathbf{x}$ and the l hidden layers $\mathbf{h}^k$ as follows:

$$P\left(\mathbf{x}, \mathbf{h}^1, \ldots, \mathbf{h}^l\right) = \left(\prod_{k=0}^{l-2} P\left(\mathbf{h}^k \mid \mathbf{h}^{k+1}\right)\right) P\left(\mathbf{h}^{l-1}, \mathbf{h}^l\right), \quad (9)$$

where $\mathbf{x} = \mathbf{h}^0$, $P(\mathbf{h}^k \mid \mathbf{h}^{k+1})$ is a conditional distribution for the visible units at level k conditioned on the hidden units of the RBM at level $k + 1$, and $P(\mathbf{h}^{l-1} \mid \mathbf{h}^l)$ is the visible-hidden joint distribution in the top-level RBM.

The principle of greedy layer-wise unsupervised training can be applied to DBNs with RBMs as the building blocks for each layer [33, 39]. A brief description of the process follows:

(1) Train the first layer as an RBM that models the raw input $\mathbf{x} = \mathbf{h}^0$ as its visible layer.

(2) Use that first layer to obtain a representation of the input that will be used as data for the second layer. Two common solutions exist. This representation can be chosen as being the mean activation $P(\mathbf{h}^1 = 1 \mid \mathbf{h}^0)$ or samples of $P(\mathbf{h}^1 \mid \mathbf{h}^0)$.

(3) Train the second layer as an RBM, taking the transformed data (samples or mean activation) as training examples (for the visible layer of that RBM).

(4) Iterate steps ((2) and (3)) for the desired number of layers, each time propagating upward either samples or mean values.

(5) Fine-tune all the parameters of this deep architecture with respect to a proxy for the DBN log- likelihood, or with respect to a supervised training criterion (after adding extra learning machinery to convert the learned representation into supervised predictions, e.g., a linear classifier).

There are two main advantages in the above-described greedy learning process of the DBNs [40]. First, it tackles the challenge of appropriate selection of parameters, which in some cases can lead to poor local optima, thereby ensuring that the network is appropriately initialized. Second, there is no requirement for labelled data since the process is unsupervised. Nevertheless, DBNs are also plagued by a number of shortcomings, such as the computational cost associated with training a DBN and the fact that the steps towards further optimization of the network based on maximum likelihood training approximation are unclear [41]. Furthermore, a significant disadvantage of DBNs is that they do not account for the two-dimensional structure of an input image, which may significantly affect their performance and applicability in computer vision and multimedia analysis problems. However, a later variation of the DBN, the Convolutional Deep Belief Network (CDBN) ([42, 43]), uses the spatial information of neighboring pixels by introducing convolutional RBMs, thus producing a translation invariant generative model that successfully scales when it comes to high dimensional images, as is evidenced in [44].

2.2.3. Deep Boltzmann Machines. Deep Boltzmann Machines (DBMs) [45] are another type of deep model using RBM as their building block. The difference in architecture of DBNs is that, in the latter, the top two layers form an undirected graphical model and the lower layers form a directed generative model, whereas in the DBM all the connections are undirected. DBMs have multiple layers of hidden units, where units in odd-numbered layers are conditionally independent of even-numbered layers, and vice versa. As a result, inference in the DBM is generally intractable. Nonetheless, an appropriate selection of interactions between visible and hidden units can lead to more tractable versions of the model. During network training, a DBM jointly trains all layers of a specific unsupervised model, and instead of maximizing the likelihood directly, the DBM uses a stochastic maximum likelihood (SML) [46] based algorithm to maximize the lower

bound on the likelihood. Such a process would seem vulnerable to falling in poor local minima [45], leaving several units effectively dead. Instead, a greedy layer-wise training strategy was proposed [47], which essentially consists in pretraining the layers of the DBM, similarly to DBN, namely, by stacking RBMs and training each layer to independently model the output of the previous layer, followed by a final joint fine-tuning.

Regarding the advantages of DBMs, they can capture many layers of complex representations of input data and they are appropriate for unsupervised learning since they can be trained on unlabeled data, but they can also be fine-tuned for a particular task in a supervised fashion. One of the attributes that sets DBMs apart from other deep models is that the approximate inference process of DBMs includes, apart from the usual bottom-up process, a top-down feedback, thus incorporating uncertainty about inputs in a more effective manner. Furthermore, in DBMs, by following the approximate gradient of a variational lower bound on the likelihood objective, one can jointly optimize the parameters of all layers, which is very beneficial especially in cases of learning models from heterogeneous data originating from different modalities [48].

As far as the drawbacks of DBMs are concerned, one of the most important ones is, as mentioned above, the high computational cost of inference, which is almost prohibitive when it comes to joint optimization in sizeable datasets. Several methods have been proposed to improve the effectiveness of DBMs. These include accelerating inference by using separate models to initialize the values of the hidden units in all layers [47, 49], or other improvements at the pretraining stage [50, 51] or at the training stage [52, 53].

2.3. Stacked (Denoising) Autoencoders. Stacked Autoencoders use the autoencoder as their main building block, similarly to the way that Deep Belief Networks use Restricted Boltzmann Machines as component. It is therefore important to briefly present the basics of the autoencoder and its denoising version, before describing the deep learning architecture of Stacked (Denoising) Autoencoders.

2.3.1. Autoencoders. An autoencoder is trained to encode the input $\mathbf{x}$ into a representation $\mathbf{r}(\mathbf{x})$ in a way that input can be reconstructed from $\mathbf{r}(\mathbf{x})$ [33]. The target output of the autoencoder is thus the autoencoder input itself. Hence, the output vectors have the same dimensionality as the input vector. In the course of this process, the reconstruction error is being minimized, and the corresponding code is the learned feature. If there is one linear hidden layer and the mean squared error criterion is used to train the network, then the k hidden units learn to project the input in the span of the first k principal components of the data [54]. If the hidden layer is nonlinear, the autoencoder behaves differently from PCA, with the ability to capture multimodal aspects of the input distribution [55]. The parameters of the model are optimized so that the average reconstruction error is minimized. There are many alternatives to measure the reconstruction error, including the traditional squared error:

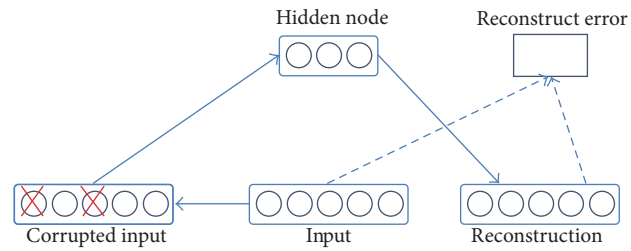


Figure 3: Denoising autoencoder [56].

$$L = \|\mathbf{x} - \mathbf{f}(\mathbf{r}(\mathbf{x}))\|^2, \tag{10}$$

where function $\mathbf{f}$ is the *decoder* and $\mathbf{f}(\mathbf{r}(\mathbf{x}))$ is the reconstruction produced by the model.

If the input is interpreted as bit vectors or vectors of bit probabilities, then the loss function of the reconstruction could be represented by cross-entropy; that is,

$$L = -\sum_i \mathbf{x}_i \log \mathbf{f}_i(\mathbf{r}(\mathbf{x})) + (1 - \mathbf{x}_i) \log(1 - \mathbf{f}_i(\mathbf{r}(\mathbf{x}))). \tag{11}$$

The goal is for the representation (or *code*) $\mathbf{r}(\mathbf{x})$ to be a distributed representation that manages to capture the coordinates along the main variations of the data, similarly to the principle of Principal Components Analysis (PCA). Given that $\mathbf{r}(\mathbf{x})$ is not lossless, it is impossible for it to constitute a successful compression for all input $\mathbf{x}$. The aforementioned optimization process results in low reconstruction error on test examples from the same distribution as the training examples but generally high reconstruction error on samples arbitrarily chosen from the input space.

2.3.2. Denoising Autoencoders. The denoising autoencoder [56] is a stochastic version of the autoencoder where the input is stochastically corrupted, but the uncorrupted input is still used as target for the reconstruction. In simple terms, there are two main aspects in the function of a denoising autoencoder: first it tries to encode the input (namely, preserve the information about the input), and second it tries to undo the effect of a corruption process stochastically applied to the input of the autoencoder (see Figure 3). The latter can only be done by capturing the statistical dependencies between the inputs. It can be shown that the denoising autoencoder maximizes a lower bound on the log-likelihood of a generative model.

In [56], the stochastic corruption process arbitrarily sets a number of inputs to zero. Then the denoising autoencoder is trying to predict the corrupted values from the uncorrupted ones, for randomly selected subsets of missing patterns. In essence, the ability to predict any subset of variables from the remaining ones is a sufficient condition for completely capturing the joint distribution between a set of variables. It should be mentioned that using autoencoders for denoising was introduced in earlier works (e.g., [57]), but the substantial contribution of [56] lies in the demonstration of the successful use of the method for unsupervised pretraining of a deep architecture and in linking the denoising autoencoder to a generative model.

2.3.3. Stacked (Denoising) Autoencoders. It is possible to stack denoising autoencoders in order to form a deep network by feeding the latent representation (output code) of the denoising autoencoder of the layer below as input to the current layer. The unsupervised pretraining of such an architecture is done one layer at a time. Each layer is trained as a denoising autoencoder by minimizing the error in reconstructing its input (which is the output code of the previous layer). When the first k layers are trained, we can train the $(k + 1)$th layer since it will then be possible compute the latent representation from the layer underneath.

When pretraining of all layers is completed, the network goes through a second stage of training called fine-tuning. Here supervised fine-tuning is considered when the goal is to optimize prediction error on a supervised task. To this end, a logistic regression layer is added on the output code of the output layer of the network. The derived network is then trained like a multilayer perceptron, considering only the encoding parts of each autoencoder at this point. This stage is supervised, since the target class is taken into account during training.

As is easily seen, the principle for training stacked autoencoders is the same as the one previously described for Deep Belief Networks, but using autoencoders instead of Restricted Boltzmann Machines. A number of comparative experimental studies show that Deep Belief Networks tend to outperform stacked autoencoders ([58, 59]), but this is not always the case, especially when DBNs are compared to Stacked Denoising Autoencoders [56].

One strength of autoencoders as the basic unsupervised component of a deep architecture is that, unlike with RBMs, they allow almost any parametrization of the layers, on condition that the training criterion is continuous in the parameters. In contrast, one of the shortcomings of SAs is that they do not correspond to a generative model, when with generative models like RBMs and DBNs, samples can be drawn to check the outputs of the learning process.

2.4. Discussion. Some of the strengths and limitations of the presented deep learning models were already discussed in the respective subsections. In an attempt to compare these models (for a summary see Table 2), we can say that CNNs have generally performed better than DBNs in current literature on benchmark computer vision datasets such as MNIST. In cases where the input is nonvisual, DBNs often outperform other models, but the difficulty in accurately estimating joint probabilities as well as the computational cost in creating a DBN constitutes drawbacks. A major positive aspect of CNNs is "feature learning," that is, the bypassing of handcrafted features, which are necessary for other types of networks; however, in CNNs features are automatically learned. On the other hand, CNNs rely on the availability of ground truth, that is, labelled training data, whereas DBNs/DBMs and SAs do not have this limitation and can work in an unsupervised manner. On a different note, one of the disadvantages of autoencoders lies in the fact that they could become ineffective if errors are present in the first layers. Such errors may cause the network to learn to reconstruct the average of the training data. Denoising autoencoders [56], however, can retrieve the correct input from a corrupted version, thus leading the network to grasp the structure of the input distribution. In terms of the efficiency of the training process, only in the case of SAs is real-time training possible, whereas CNNs and DBNs/DBMs training processes are time-consuming. Finally, one of the strengths of CNNs is the fact that they can be invariant to transformations such as translation, scale, and rotation. Invariance to translation, rotation, and scale is one of the most important assets of CNNs, especially in computer vision problems, such as object detection, because it allows abstracting an object's identity or category from the specifics of the visual input (e.g., relative positions/orientation of the camera and the object), thus enabling the network to effectively recognize a given object in cases where the actual pixel values on the image can significantly differ.

Table 2: Comparison of CNNs, DBNs/DBMs, and SdAs with respect to a number of properties. + denotes a good performance in the property and − denotes bad performance or complete lack thereof.

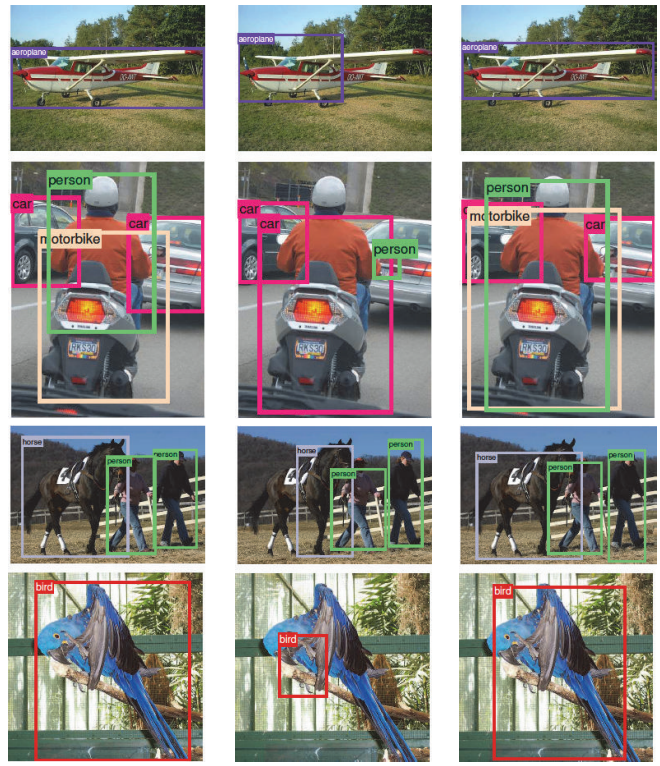
Model properties	CNNs	DBNs/DBMs	SdAs
Unsupervised learning	−	+	+
Training efficiency	−	−	+
Feature learning	+	−	−
Scale/rotation/translation invariance	+	−	−
Generalization	+	+	+

3. Applications in Computer Vision

In this section, we survey works that have leveraged deep learning methods to address key tasks in computer vision, such as object detection, face recognition, action and activity recognition, and human pose estimation.

3.1. Object Detection. Object detection is the process of detecting instances of semantic objects of a certain class (such as humans, airplanes, or birds) in digital images and video (Figure 4). A common approach for object detection frameworks includes the creation of a large set of candidate windows that are in the sequel classified using CNN features. For example, the method described in [32] employs selective search [60] to derive object proposals, extracts CNN features for each proposal, and then feeds the features to an SVM classifier to decide whether the windows include the object or not. A large number of works is based on the concept of Regions with CNN features proposed in [32]. Approaches following the Regions with CNN paradigm usually have good detection accuracies (e.g., [61, 62]); however, there is a significant number of methods trying to further improve the performance of Regions with CNN approaches, some of which succeed in finding approximate object positions but often cannot precisely determine the exact position of the object [63]. To this end, such methods often follow a joint object detection—semantic segmentation approach [64–66], usually attaining good results.

A vast majority of works on object detection using deep learning apply a variation of CNNs, for example, [8, 67, 68]



(a) (b) (c)

Figure 4: Object detection results comparison from [66]. (a) Ground truth; (b) bounding boxes obtained with [32]; (c) bounding boxes obtained with [66].

(in which a new def-pooling layer and new learning strategy are proposed), [9] (weakly supervised cascaded CNNs), and [69] (subcategory-aware CNNs). However, there does exist a relatively small number of object detection attempts using other deep models. For example, [70] proposes a coarse object locating method based on a saliency mechanism in conjunction with a DBN for object detection in remote sensing images; [71] presents a new DBN for 3D object recognition, in which the top-level model is a third-order Boltzmann machine, trained using a hybrid algorithm that combines both generative and discriminative gradients; [72] employs a fused deep learning approach, while [73] explores the representation capabilities of a deep model in a semisupervised paradigm. Finally, [74] leverages stacked autoencoders for multiple organ detection in medical images, while [75] exploits saliency-guided stacked autoencoders for video-based salient object detection.

3.2. Face Recognition. Face recognition is one of the hottest computer vision applications with great commercial interest as well. A variety of face recognition systems based on the extraction of handcrafted features have been proposed [76–79]; in such cases, a feature extractor extracts features from an aligned face to obtain a low-dimensional representation, based on which a classifier makes predictions. CNNs brought about a change in the face recognition field, thanks to their feature learning and transformation invariance properties. The first work employing CNNs for face recognition was [80]; today light CNNs [81] and VGG Face Descriptor [82] are among the state of the art. In [44] a Convolutional DBN achieved a great performance in face verification.

Moreover, Google's FaceNet [83] and Facebook's DeepFace [84] are both based on CNNs. DeepFace [84] 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, followed by three locally connected layers and two fully connected layers used to make final predictions. Although DeepFace attains great performance rates, its representation is not easy to interpret because the faces of the same person are not necessarily clustered during the training process. On the other hand, FaceNet defines a triplet loss function on the representation, which makes the training process learn to cluster the face representation of the same person. Furthermore, CNNs constitute the core of OpenFace [85], an open-source face recognition tool, which is of comparable (albeit a little lower) accuracy, is open-source, and is suitable for mobile computing, because of its smaller size and fast execution time.

3.3. Action and Activity Recognition. Human action and activity recognition is a research issue that has received a lot of attention from researchers [86, 87]. Many works on human activity recognition based on deep learning techniques have been proposed in the literature in the last few years [88]. In [89] deep learning was used for complex event detection and recognition in video sequences: first, saliency maps were used

for detecting and localizing events, and then deep learning was applied to the pretrained features for identifying the most important frames that correspond to the underlying event. In [90] the authors successfully employ a CNN-based approach for activity recognition in beach volleyball, similarly to the approach of [91] for event classification from large-scale video datasets; in [92], a CNN model is used for activity recognition based on smartphone sensor data. The authors of [12] incorporate a radius–margin bound as a regularization term into the deep CNN model, which effectively improves the generalization performance of the CNN for activity classification. In [13], the authors scrutinize the applicability of CNN as joint feature extraction and classification model for fine-grained activities; they find that due to the challenges of large intraclass variances, small interclass variances, and limited training samples per activity, an approach that directly uses deep features learned from ImageNet in an SVM classifier is preferable.

Driven by the adaptability of the models and by the availability of a variety of different sensors, an increasingly popular strategy for human activity recognition consists in fusing multimodal features and/or data. In [93], the authors mixed appearance and motion features for recognizing group activities in crowded scenes collected from the web. For the combination of the different modalities, the authors applied multitask deep learning. The work of [94] explores combination of heterogeneous features for complex event recognition. The problem is viewed as two different tasks: first, the most informative features for recognizing events are estimated, and then the different features are combined using an AND/OR graph structure. There is also a number of works combining more than one type of model, apart from several data modalities. In [95], the authors propose a multimodal multistream deep learning framework to tackle the egocentric activity recognition problem, using both the video and sensor data and employing a dual CNNs and Long Short-Term Memory architecture. Multimodal fusion with a combined CNN and LSTM architecture is also proposed in [96]. Finally, [97] uses DBNs for activity recognition using input video sequences that also include depth information.

3.4. Human Pose Estimation. The goal of human pose estimation is to determine the position of human joints from images, image sequences, depth images, or skeleton data as provided by motion capturing hardware [98]. Human pose estimation is a very challenging task owing to the vast range of human silhouettes and appearances, difficult illumination, and cluttered background. Before the era of deep learning, pose estimation was based on detection of body parts, for example, through pictorial structures [99].

Moving on to deep learning methods in human pose estimation, we can group them into holistic and part-based methods, depending on the way the input images are processed. The holistic processing methods tend to accomplish their task in a global fashion and do not explicitly define a model for each individual part and their spatial relationships. DeepPose [14] is a holistic model that formulates the human pose estimation method as a joint regression problem and does not explicitly define the graphical model or part detectors for the human pose estimation. Nevertheless, holistic-based methods tend to be plagued by inaccuracy in the high-precision region due to the difficulty in learning direct regression of complex pose vectors from images.

On the other hand, the part-based processing methods focus on detecting the human body parts individually, followed by a graphic model to incorporate the spatial information. In [15], the authors, instead of training the network using the whole image, use the local part patches and background patches to train a CNN, in order to learn conditional probabilities of the part presence and spatial relationships. In [100] the approach trains multiple smaller CNNs to perform independent binary body-part classification, followed with a higher-level weak spatial model to remove strong outliers and to enforce global pose consistency. Finally, in [101], a multi-resolution CNN is designed to perform heat-map likelihood regression for each body part, followed with an implicit graphic model to further promote joint consistency.

3.5. Datasets. The applicability of deep learning approaches has been evaluated on numerous datasets, whose content varied greatly, according the application scenario. Regardless of the investigated case, the main application domain is (natural) images. A brief description of utilized datasets (traditional and new ones) for benchmarking purposes is provided below.

(1) Grayscale Images. The most used grayscale images dataset is MNIST [20] and its variations, that is, NIST and perturbed NIST. The application scenario is the recognition of handwritten digits.

(2) RGB Natural Images. Caltech RGB image datasets [102], for example, Caltech 101/Caltech 256 and the Caltech Silhouettes, contain pictures of objects belonging to 101/256 categories. CIFAR datasets [103] consist of thousands of 32×32 color images in various classes. COIL datasets [104] consist of different objects imaged at every angle in a 360 rotation.

(3) Hyperspectral Images. SCIEN hyperspectral image data [105] and AVIRIS sensor based datasets [106], for example, contain hyperspectral images.

(4) Facial Characteristics Images. Adience benchmark dataset [107] can be used for facial attributes identification, that is, age and gender, from images of faces. Face recognition in unconstrained environments [108] is another commonly used dataset.

(5) Medical Images. Chest X-ray dataset [109] comprises 112120 frontal-view X-ray images of 30805 unique patients with the text-mined fourteen disease image labels (where each image can have multilabels). Lymph Node Detection and Segmentation datasets [110] consist of Computed Tomography images of the mediastinum and abdomen.

(6) Video Streams. The WR datasets [111, 112] can be used for video-based activity recognition in assembly lines [113], containing sequences of 7 categories of industrial tasks. YouTube-8M [114] is a dataset of 8 million YouTube video URLs, along with video-level labels from a diverse set of 4800 Knowledge Graph entities.

4. Conclusions

The surge of deep learning over the last years is to a great extent due to the strides it has enabled in the field of computer vision. The three key categories of deep learning for computer vision that have been reviewed in this paper, namely, CNNs, the "Boltzmann family" including DBNs and DBMs, and SdAs, have been employed to achieve significant performance rates in a variety of visual understanding tasks, such as object detection, face recognition, action and activity recognition, human pose estimation, image retrieval, and semantic segmentation. However, each category has distinct advantages and disadvantages. CNNs have the unique capability of feature learning, that is, of automatically learning features based on the given dataset. CNNs are also invariant to transformations, which is a great asset for certain computer vision applications. On the other hand, they heavily rely on the existence of labelled data, in contrast to DBNs/DBMs and SdAs, which can work in an unsupervised fashion. Of the models investigated, both CNNs and DBNs/DBMs are computationally demanding when it comes to training, whereas SdAs can be trained in real time under certain circumstances.

As a closing note, in spite of the promising—in some cases impressive—results that have been documented in the literature, significant challenges do remain, especially as far as the theoretical groundwork that would clearly explain the ways to define the optimal selection of model type and structure for a given task or to profoundly comprehend the reasons for which a specific architecture or algorithm is effective in a given task or not. These are among the most important issues that will continue to attract the interest of the machine learning research community in the years to come.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is implemented through IKY scholarships programme and cofinanced by the European Union (European Social Fund—ESF) and Greek national funds through the action titled "Reinforcement of Postdoctoral Researchers," in the framework of the Operational Programme "Human Resources Development Program, Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) 2014–2020.

References

[1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology*, vol. 5, no. 4, pp. 115–133, 1943.

[2] Y. LeCun, B. Boser, J. Denker et al., "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2 (NIPS*89)*, D. Touretzky, Ed., Denver, CO, USA, 1990.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] 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.

[5] TensorFlow, Available online: https://www.tensorflow.org.

[6] B. Frederic, P. Lamblin, R. Pascanu et al., "Theano: new features and speed improvements," in *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*, 2012, http://deeplearning.net/software/theano/.

[7] Mxnet, Available online: http://mxnet.io.

[8] W. Ouyang, X. Zeng, X. Wang et al., "DeepID-Net: Object Detection with Deformable Part Based Convolutional Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1320–1334, 2017.

[9] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. V. Gool, "Weakly Supervised Cascaded Convolutional Networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5131–5139, Honolulu, HI, July 2017.

[10] N. Doulamis and A. Voulodimos, "FAST-MDL: Fast Adaptive Supervised Training of multi-layered deep learning models for consistent object tracking and classification," in *Proceedings of the 2016 IEEE International Conference on Imaging Systems and Techniques, IST 2016*, pp. 318–323, October 2016.

[11] N. Doulamis, "Adaptable deep learning structures for object labeling/tracking under dynamic visual environments," *Multimedia Tools and Applications*, pp. 1–39, 2017.

[12] 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.

[13] S. Cao and R. Nevatia, "Exploring deep learning based solutions in fine grained activity recognition in the wild," in *Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 384–389, Cancun, December 2016.

[14] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 1653–1660, USA, June 2014.

[15] X. Chen and A. L. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," in *Proceedings of the NIPS*, 2014.

[16] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 1520–1528, Santiago, Chile, December 2015.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 3431–3440, IEEE, Boston, Mass, USA, June 2015.

[18] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, pp. 106–154, 1962.

[19] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[21] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[22] M. Tygert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino, and A. Szlam, "A mathematical motivation for complex-valued convolutional networks," *Neural Computation*, vol. 28, no. 5, pp. 815–825, 2016.

[23] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free? - Weakly-supervised learning with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 685–694, June 2015.

[24] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, Boston, Mass, USA, June 2015.

[25] Y. L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the ICML*, 2010.

[26] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6354, no. 3, pp. 92–101, 2010.

[27] H. Wu and X. Gu, "Max-Pooling Dropout for Regularization of Convolutional Neural Networks," in *Neural Information Processing*, vol. 9489 of *Lecture Notes in Computer Science*, pp. 46–54, Springer International Publishing, Cham, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *Computer Vision – ECCV 2014*, vol. 8691 of *Lecture Notes in Computer Science*, pp. 346–361, Springer International Publishing, Cham, 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[30] W. Ouyang, X. Wang, X. Zeng et al., "DeepID-Net: Deformable deep convolutional neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 2403–2412, USA, June 2015.

[31] 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, Lake Tahoe, Nev, USA, December 2012.

[32] 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, Columbus, Ohio, USA, June 2014.

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

[34] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *In Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, pp. 194–281, MIT Press, Cambridge, MA, USA, 1986.

[35] G. E. Hinton and T. J. Sejnowski, "Learning and Relearning in Boltzmann Machines," vol. 1, p. 4.2, MIT Press, Cambridge, MA, 1986.

[36] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proceedings of the tenth international workshop on artificial intelligence and statistics., NP: Society for Artificial Intelligence and Statistics*, pp. 33–40, 2005.

[37] G. Hinton, "A practical guide to training restricted Boltzmann machines," *Momentum*, vol. 9, p. 926, 2010.

[38] K. Cho, T. Raiko, and A. Ilin, "Enhanced gradient for training restricted Boltzmann machines," *Neural Computation*, vol. 25, no. 3, pp. 805–831, 2013.

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

[40] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning—a new frontier in artificial intelligence research," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[41] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[42] 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 (ICML '09)*, pp. 609–616, ACM, Montreal, Canada, June 2009.

[43] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks," *Communications of the ACM*, vol. 54, no. 10, pp. 95–103, 2011.

[44] G. B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2518–2525, June 2012.

[45] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, vol. 24, pp. 448–455, 2009.

[46] L. Younes, "On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates," *Stochastics and Stochastics Reports*, vol. 65, no. 3-4, pp. 177–228, 1999.

[47] R. Salakhutdinov and H. Larochelle, "Efficient learning of deep Boltzmann machines," in *Proceedings of the AISTATS*, 2010.

[48] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," *Journal of Machine Learning Research*, vol. 15, pp. 2949–2980, 2014.

[49] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep Boltzmann machines," *Neural Computation*, vol. 24, no. 8, pp. 1967–2006, 2012.

[50] R. Salakhutdinov and G. Hinton, "A better way to pretrain Deep Boltzmann Machines," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems 2012, NIPS 2012*, pp. 2447–2455, usa, December 2012.

[51] K. Cho, T. Raiko, A. Ilin, and J. Karhunen, "A two-stage pretraining algorithm for deep boltzmann machines," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8131, pp. 106–113, 2013.

[52] G. Montavon and K. Müller, "Deep Boltzmann Machines and the Centering Trick," in *Neural Networks: Tricks of the Trade*, vol. 7700 of *Lecture Notes in Computer Science*, pp. 621–637, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[53] I. Goodfellow, M. Mirza, A. Courville et al., "Multi-prediction deep Boltzmann machines," in *Proceedings of the NIPS*, 2013.

[54] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, vol. 59, no. 4-5, pp. 291–294, 1988.

[55] N. Japkowicz, S. J. Hanson, and M. A. Gluck, "Nonlinear autoassociation is not equivalent to PCA," *Neural Computation*, vol. 12, no. 3, pp. 531–545, 2000.

[56] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *in Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)*, W. W. Cohen, A. McCallum, and S. T. Roweis, Eds., pp. 1096–1103, ACM, 2008.

[57] P. Gallinari, Y. LeCun, S. Thiria, and F. Fogelman-Soulie, "Memoires associatives distribuees," in *Proceedings of the in Proceedings of COGNITIVA 87*, Paris, La Villette, 1987.

[58] 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, Corvallis, Ore, UA, June 2007.

[59] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems (NIPS06)*, B. Sch, J. Platt, and., T. Hoffman, and B. Schölkopf, Eds., vol. 19, pp. 153–160, MIT Press, 2007.

[60] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

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

[62] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[63] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?" in *Proceedings of the 25th British Machine Vision Conference, BMVC 2014*, gbr, September 2014.

[64] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Computer Vision—ECCV 2014*, vol. 8695 of *Lecture Notes in Computer Science*, pp. 297–312, Springer, 2014.

[65] J. Dong, Q. Chen, S. Yan, and A. Yuille, "Towards unified object detection and semantic segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8693, no. 5, pp. 299–314, 2014.

[66] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler, "SegDeepM: Exploiting segmentation and context in deep neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 4703–4711, USA, June 2015.

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

[68] 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 2016 Computing in Cardiology Conference*.

[69] T. Chen, S. Lu, and J. Fan, "S-CNN: Subcategory-aware convolutional networks for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[70] W. Diao, X. Sun, X. Zheng, F. Dou, H. Wang, and K. Fu, "Efficient Saliency-Based Object Detection in Remote Sensing Images Using Deep Belief Networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 2, pp. 137–141, 2016.

[71] V. Nair and G. E. Hinton, "3D object recognition with deep belief nets," in *Proceedings of the NIPS*, 2009.

[72] N. Doulamis and A. Doulamis, "Fast and adaptive deep fusion learning for detecting visual objects," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7585, no. 3, pp. 345–354, 2012.

[73] N. Doulamis and A. Doulamis, "Semi-supervised deep learning for object tracking and classification," pp. 848–852.

[74] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1930–1943, 2013.

[75] J. Li, C. Xia, and X. Chen, "A benchmark dataset and saliency-guided stacked autoencoders for video-based salient object detection," *IEEE Transactions on Image Processing*, 2017.

[76] 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.

[77] 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.

[78] 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)*, pp. 1–11, September 2012.

[79] 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.

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

[81] 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.

[82] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in *Proceedings of the British Machine Vision Conference 2015*, pp. 41.1-41.12, Swansea.

[83] 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.

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

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

[86] A. S. Voulodimos, D. I. Kosmopoulos, N. D. Doulamis, and T. A. Varvarigou, "A top-down event-driven approach for concurrent activity recognition," *Multimedia Tools and Applications*, vol. 69, no. 2, pp. 293–311, 2014.

[87] A. S. Voulodimos, N. D. Doulamis, D. I. Kosmopoulos, and T. A. Varvarigou, "Improving multi-camera activity recognition by employing neural network based readjustment," *Applied Artificial Intelligence*, vol. 26, no. 1-2, pp. 97–118, 2012.

[88] K. Makantasis, A. Doulamis, N. Doulamis, and K. Psychas, "Deep learning based human behavior recognition in industrial workflows," in *Proceedings of the 23rd IEEE International Conference on Image Processing, ICIP 2016*, pp. 1609–1613, September 2016.

[89] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann, "DevNet: A Deep Event Network for multimedia event detection and evidence recounting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 2568–2577, USA, June 2015.

[90] T. Kautz, B. H. Groh, J. Hannink, U. Jensen, H. Strubberg, and B. M. Eskofier, "Activity recognition in beach volleyball using a DEEp Convolutional Neural NETwork: leveraging the potential of DEEp Learning in sports," *Data Mining and Knowledge Discovery*, vol. 31, no. 6, pp. 1678–1705, 2017.

[91] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F.-F. Li, "Large-scale video classification with convolutional neural networks," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR '14)*, pp. 1725–1732, Columbus, OH, USA, June 2014.

[92] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235–244, 2016.

[93] J. Shao, C. C. Loy, K. Kang, and X. Wang, "Crowded Scene Understanding by Deeply Learned Volumetric Slices," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 613–623, 2017.

[94] K. Tang, B. Yao, L. Fei-Fei, and D. Koller, "Combining the right features for complex event recognition," in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, ICCV 2013*, pp. 2696–2703, Australia, December 2013.

[95] S. Song, V. Chandrasekhar, B. Mandal et al., "Multimodal Multi-Stream Deep Learning for Egocentric Activity Recognition," in *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2016*, pp. 378–385, USA, July 2016.

[96] R. Kavi, V. Kulathumani, F. Rohit, and V. Kecojevic, "Multiview fusion for activity recognition using deep neural networks," *Journal of Electronic Imaging*, vol. 25, no. 4, Article ID 043010, 2016.

[97] H. Yalcin, "Human activity recognition using deep belief networks," in *Proceedings of the 24th Signal Processing and Communication Application Conference, SIU 2016*, pp. 1649–1652, tur, May 2016.

[98] A. Kitsikidis, K. Dimitropoulos, S. Douka, and N. Grammalidis, "Dance analysis using multiple kinect sensors," in *Proceedings of the 9th International Conference on Computer Vision Theory and Applications, VISAPP 2014*, pp. 789–795, prt, January 2014.

[99] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[100] A. Jain, J. Tompson, and M. Andriluka, "Learning human pose estimation features with convolutional networks," in *Proceedings of the ICLR*, 2014.

[101] J. J. Tompson, A. Jain, Y. LeCun et al., "Joint training of a convolutional network and a graphical model for human pose estimation," in *Proceedings of the NIPS*, 2014.

[102] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[103] A. Krizhevsky and G. Hinton, Learning multiple layers of features from tiny images, 2009.

[104] S. A. Nene, S. K. Nayar, and H. Murase, Columbia object image library (coil-20), 1996.

[105] T. Skauli and J. Farrell, "A collection of hyperspectral images for imaging systems research," in *Proceedings of the Digital Photography IX*, USA, February 2013.

[106] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3," *Datasets*, 2015.

[107] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2170–2179, 2014.

[108] 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., University of Massachusetts, Amherst, 2007.

[109] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-Ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3462–3471, Honolulu, HI, May 2017.

[110] A. Seff, L. Lu, A. Barbu, H. Roth, H.-C. Shin, and R. M. Summers, "Leveraging mid-level semantic boundary cues for automated lymph node detection," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9350, pp. 53–61, 2015.

[111] A. Voulodimos, D. Kosmopoulos, G. Vasileiou et al., "A dataset for workflow recognition in industrial scenes," in *Proceedings of the 2011 18th IEEE International Conference on Image Processing, ICIP 2011*, pp. 3249–3252, Belgium, September 2011.

[112] A. Voulodimos, D. Kosmopoulos, G. Vasileiou et al., "A threefold dataset for activity and workflow recognition in complex industrial environments," *IEEE MultiMedia*, vol. 19, no. 3, pp. 42–52, 2012.

[113] D. I. Kosmopoulos, A. S. Voulodimos, and A. D. Doulamis, "A system for multicamera task recognition and summarization for structured environments," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 161–171, 2013.

[114] S. Abu-El-Haija et al., "YouTube-8M: A large-scale video classification benchmark," Tech. Rep., 2016, https://arxiv.org/abs/1609.08675.