Computer Science Review 40 (2021) 100379

Contents lists available at ScienceDirect

Computer Science Review

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

Review article

A survey on deep learning and its applications

Shi Dong [a,b,*], Ping Wang [a], Khushnood Abbas [a]

[a] School of Computer Science and Technology, Zhoukou Normal University, Henan, 466000, China
[b] State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications), Beijing 100876, China

ARTICLE INFO

Article history:
Received 5 July 2020
Received in revised form 27 January 2021
Accepted 3 February 2021
Available online 15 March 2021

Keywords:
Deep learning
Stacked auto encoder
Deep belief networks
Deep Boltzmann machine
Convolutional neural network

ABSTRACT

Deep learning, a branch of machine learning, is a frontier for artificial intelligence, aiming to be closer to its primary goal—artificial intelligence. This paper mainly adopts the summary and the induction methods of deep learning. Firstly, it introduces the global development and the current situation of deep learning. Secondly, it describes the structural principle, the characteristics, and some kinds of classic models of deep learning, such as stacked auto encoder, deep belief network, deep Boltzmann machine, and convolutional neural network. Thirdly, it presents the latest developments and applications of deep learning in many fields such as speech processing, computer vision, natural language processing, and medical applications. Finally, it puts forward the problems and the future research directions of deep learning.

Contents

* Corresponding author at: School of Computer Science and Technology, Zhoukou Normal University, Henan, 466000, China.
E-mail address: njbsok@163.com (S. Dong).

https://doi.org/10.1016/j.cosrev.2021.100379
1574-0137/

1. Introduction

Deep learning is nothing but many classifiers working together, which are based on linear regression followed by some activation functions. Its basis is the same as the traditional statistical linear regression $W^TX + b$ approach. The only difference is that there are many neural nodes in deep learning instead of only one node which is called linear regression in the traditional statistical learning. These neural nodes are also known as a neural network, and one classifier node is known as a neural unit or perception. Another contrasting point need to be noticed is that in deep learning there are many layers between the input and the output. A layer can have many hundreds or even thousands of neural units. The layers which are in between the input and the output known as the hidden layers and the nodes are known as the hidden nodes. The draw-back of the traditional machine learning classifiers is that we need to write a complex hypothesis by ourselves, while in the deep neural network it is generated by the network itself, which makes it a powerful tool for learning nonlinear relationships effectively. Machine learning can be divided into two development processes, including shallow learning and deep learning. In 2006, before the deep learning was again introduced into the research trend, the research direction mainly focuses on the shallow learning structure for data processing. Compared with the deep learning, the shallow learning will be limited not to exceed two layers of non-linear feature conversion layer. The most common shallow structures include Logistic Regression [1–4], Support Vector Machines [5–8], Gaussian Mixture Models [9,10], and so on. So far, shallow learning can only quickly and efficiently solve the problem with multiple restrictions, but it cannot handle the complex problem in the real world, such as the human voices, the natural pictures, the visual scenes, and so on. The shallow learning has a limitation so that it can never be handled like the human brain for information. In 2006, Hinton et al. [11] put forward a deep belief network (DBN, Deep Belief Network), which was stacked through a number of restricted Boltzmann machines (RBM, Restricted Boltzmann Machine). They put forward an unsupervised training algorithm with greedy layer-by-layer through unsupervised learning and training. Then, they put the data by learning as an initial value of supervised learning. So that the deep learning structure could solve the problem which the shallow learning could not solve. As the deep learning started its development, more and more scientific and technological personnel began to focus on the applications of the deep learning research, which significantly promoted the development of the human intelligence. The study of deep learning is mainly embodied in the convening of various world-class artificial intelligence conferences, the establishment of the world elite research group, the establishment of the enterprise research team, and the continuous applications of deep learning in artificial intelligence. Deep learning algorithms are proposed continuously, and new records are created continuously in many data sets. For example, in the test process of image classification for 1000 kinds of images, in five years, through the continuous improvement of the deep learning model, the image classification error rate dropped to 3.5%, which is higher than the accuracy of the ordinary people. In fact, that was a success of using deep learning to enable machines to learn how to successfully identify and categorize images. The development of science and technology is constantly refreshing the human cognition, and deep learning model is constantly being updated as the core technology model of the artificial intelligence in the big data environment, reflecting the latest research progress of the current science and technology.

2. History of deep neural network

The initial move towards neural Networks occurred in 1943, when Warren McCulloch, a neurophysiologist, and a youthful mathematician, Walter Pitts, composed a paper on how neurons may function. They proposed a basic neural network with electrical circuits. In 1949 Donald Hebb theorized that neural pathways are strengthened each time they are used [12]. In 1950s, Nathanial Rochester from the IBM research to simulated abstract neural network on IBM 704 computers [13]. In 1956 four scientists worked together on a summer project known as Dartmouth Summer Research project on Artificial Intelligence. The four scientists were John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. They provided insightful leap in AI research [14]. Following the Dartmouth project in 1957, John Von Neumann proposed that telegraph relays or vacuum tubes can be used to imitate the simple neuron function. In 1958, Frank Rosenblatt, a neuro-biologist of Cornell, began work on the Perceptron. He was charmed with the activity of the eye of a fly. A significant part of the preparing which advises a fly to escape is done in its eye. The Perceptron, which came about because of this research, was built in hardware and is the most established neural network still being used today. A single layer perceptron was discovered to be helpful in classifying a single valued set of inputs into one of two classes. The perceptron figures a weighted amount of the data sources, takes away a limit, and passes one of two potential qualities out as the outcome. In 1959, Bernard Widrow and Marcian Hoff of Stanford created models they called ADALINE and MADALINE [1]. These models were named for their utilization of Multiple ADAptive LINear Elements. MADALINE was the primary neural network to be applied to a real-world problem. It is a adaptive channel which eliminate with echoes on telephone lines. This neural organization is still in business use. Shockingly, these prior triumphs made individuals overstate the capability of neural networks, especially considering the restriction in the hardware then accessible. This extreme hype, which streamed out of the academic and technical fields, tainted the overall writing of the time. Disillusionment set in as guarantees were unfilled. Likewise, a dread set in as essayists contemplated what impact "figuring machines" would have on man. Asimov's arrangement on robots uncovered the impacts on man's ethics and qualities when machines where equipped for doing the entirety of humankind's work. In 1982, interest in the field was reestablished. John Hopfield of Caltech introduced a

[1] http://www2.psych.utoronto.ca/users/reingold/courses/ai/cache/neural4.html

paper to the National Academy of Sciences [2]. His methodology was to make more valuable machines by utilizing bidirectional lines. Beforehand, the associations between neurons was just a single way. Additionally, in 1982, there was a joint US–Japan Conference on Cooperative/Competitive Neural Networks. Japan declared another Fifth Generation exertion on neural networks, and US papers created stress that the US could be abandoned in the field (Fifth era processing includes computerized reasoning. Original utilized switches and wires, the second era utilized the transistor, third state utilized strong state innovation like incorporated circuits and more significant level programming dialects, and the fourth era is code generators.). Subsequently, there was additionally subsidizing and, in this manner, more exploration in the field. In 1985, American Institute of Physics started what has become a yearly gathering — Neural Networks for Computing. By 1987, the Institute of Electrical and Electronic Engineer's (IEEE) first International Conference on Neural Networks drew in excess of 1,800 participants. In 1997, A recurrent neural network structure, Long Short-Term Memory (LSTM) was proposed by Schmidhuber and Hochreiter. Long momentary memory (LSTM) is artificial recurrent neural network (RNN) architecture [1] utilized in the field of deep learning. Not at all like standard feedforward neural networks, LSTM has feedback connections. It cannot just cycle single information focuses, (for example, pictures), yet additionally whole sequence of data, (for example, speech or video). In 1998, Yann LeCun published Gradient-Based Learning Applied to Document Recognition which was a major step in learning from data [15].

3. Activation functions

Another important factor in a neural network is the activation functions which are inspired by the human neural firing, i.e., it either fire or not. The activation functions are used to generate nonlinear relationships between the input and the output. This nonlinearity, combined with many neural nodes and many layers, mimics the human brain like structure, which is why it is called a neural network. There are many activation functions (some of them presented in Fig. 1(B)). In Fig. 1, we have plotted different activation functions, which are commonly used, such as Sigmoid, Hyperbolic tangent, and Relu. The role of the activation function is to transform and abstract the data into a more classifiable plane. Generally, the data is very tightly clustered; it is the job of the activation function which transforms the data into a different plane which helps in observing the effects of different dimensions in the given problem. The best and classic example of the activation function is sigmoid activation, which is used in logistic regression. In fact, the logistic regression can be considered as one neural unit (See Fig. 1(A)). The job of the sigmoid function is to take any input and give an output between 0 and 1, which is used for the classification problems. In Fig. 1(C), we have plotted one hidden layer neural network, which has three hidden neural units in the hidden layer and one in the output layer. This hidden unit is similar to the logistic regression model. The difference is that in the next layer, the input comes from the layer just before it. In Fig. 1(D), we have plotted a description of more than one hidden layer and more than one neural unit in every layer. From Fig. 1, it can be easily noticed that the neural network can consist of many layers, and every layer can have any number of neural units.

[2] https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history2.html

4. Parameter learning

As the traditional machine learning classifiers, the deep learning classifiers also need to learn parameters with the help of some mathematical tools such as the gradient descent. The gradient descent algorithm is very useful in learning parameters for convex functions. A function is convex if it has one absolute minima/maxima. If the function is convex, then the parameter learning is easy; otherwise, it needs some mathematical tricks to change a non-convex function into convex function. This problem is also known as a convex optimization problem. However, technicality, neural network optimization is a non-convex optimization. It means that it has many optimum (minima/maxima) points. The learning is done by minimizing the error between the predicted value and the actual value.

5. Deep learning performance

This figure (Fig. 2) shows how the size of the neural network affects the prediction accuracy. For small data with small size, the neural network can perform as Regression/Logistic Regression and SVM (SVM, Support Vector Machines) classifiers. But, for big data a small neural network is better than classical classifiers. However, bigger neural network improves its performance if it is trained on big data. The performance of bigger size neural network grows with the data as it is compared to the classical models and the medium and smaller neural networks. Wange et al. [16] found that deep neural networks can be better perceived by using the knowledge received by the visualization of the output images acquired at each layer. Studies has been done to improve the visualization idea in the neural networks by methods for a strategy of obscuring and de-obscuring technique which may be useful to its (neural network) performance.

5.1. Limitation and key issues

Deep learning are estimation of large number of parameters that goes from input space to output vector space. They are learning continuous geometric morphing from an input vector space to output vector space. They are training point by point so a deep neural network can best interpolate to points which are near to training points. Which means to best learn we need point by point training for all possible outcomes, which can be expensive for real world complex problems such as autonomous driving etc. Training with huge number of possible outcomes can reduce the chance of testing error. Deep learning fails to abstract information. For example an algorithm which is based on logic can be applied on verity of unknown data. But deep learning can only be applied on which data the model was trained. Although it is known that deep neural networks can approximate any degree of polynomial function in a given time. The main challenge is the optimizing errors. The error optimization is difficult because there is no one or generic function to achieve this task.

5.2. Optimization

It is understood that many real world problems with a natural non-convex optimizations are NP-hard [17]. Optimization for high dimensional data is an open challenge for researchers [18]. Nouiehed and Razaviyayn [17] proposed the solution to the non-convex optimization problem in neural network. Yun et al. [19] found that even if the input data are random and the labels are created according to a planted model the loss surface of nonlinear networks contains spurious local minima. Furthermore, Yun et al. [19] observed that intrusion of minor nonlinearities in activation functions caused bad local minima in loss surface of neural networks. Haeffele and Vidal came up with solution to find optimum minima in Deep neural networks by studying the effect of over-parametrization on the training of neural networks [20].

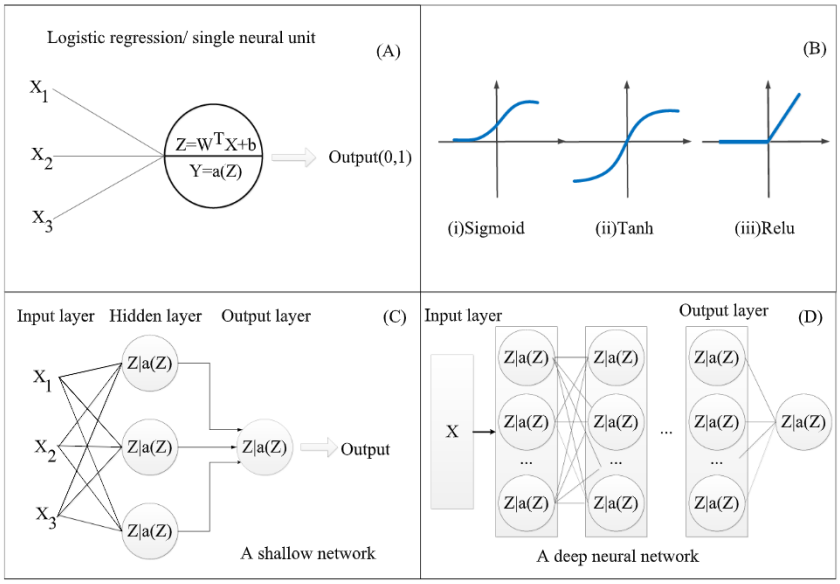

Fig. 1. Basics of (Deep) Neural Network. Sub-figure (A) shows the classic logistic regression,(B) shows different activation functions plot, (C) shows one hidden layer shallow network and (D) depicts more than two hidden layer neural network.

5.3. Architecture

To learn the complex relationship from the data neurons/perceptron/single neural unit are stacked together either serial, parallel or both. It is found that these arrangements of neurons also affect the over all neural networks performance. Haeffele et al. [21] observed that for networks with specific structures there is a path from initialization to global minima. This observation helps to find long lasting questing for finding optimum minima for neural networks.

5.4. Generalization and regularization

Dropout is widely used to prevent over-fitting by switching off some of the neural units randomly. It uses predefined dropout probability. It is found very strong regularizer in neural networks [22]. Mianjy et al. [23] investigated the effect of dropout regularization of an linear autoencoders (LAEs). They found that induced regularizer is indeed nuclear norm. Ising-dropout [24] is another recent dropout technique. It put the graphical Ising model on top of a neural network in order to identify less useful neurons, and drop them. The Ising dropout model is energy-based dropout method which switch offs the neural units based on activation values in dense layers of neural networks.

5.5. Stability and robustness

Sengupta et al. [25] investigated that as the prediction time increases the RNN model loses its robustness. Different experiments for the same task are obtained with different dynamical behaviors. It suggests that risk-sensitive policy selection minimizes expected complexity or computational cost (Sengupta and Friston). Zheng et al. [26] propose to attach a stability term to the objective function, which powers the model to have comparable yields for tests of the training set and their perturbed forms. Further, they propose to improve the robustness of neural organizations against little perturbation to enter pictures. [27] explored the performance of neural networks on different random weights. Haber et al. [28] interpreted deep learning as a parameter estimation problem of nonlinear dynamical systems. Given this formulation, they analyze stability and well-posedness of deep neural network and use this new understanding to develop new network architectures. Further they introduce the concept of stable networks that can be arbitrarily long. Malladi et al. [29] proposed fast normalization technique which converges and cost less computation cycle. It exploits the low-rank properties of weight updates and predicts the norms without explicitly calculating them, The capacity to revert neural networks mitigates the need to store activation values for backpropagation. Further decreasing the memory impression of our calculation in those applications motivates the utilization of reversible neural networks emerging from hyperbolic systems [30]. It additionally opens up the likelihood to build various networks by utilizing distinctive discretizations of the ODE, as in the midpoint network in.

The paper is structured as follows: Section 6 gives out an introduction and an overview of the deep learning models; Section 7 discusses the applications of deep learning technology; Section 8 gives example how to use deep learning for solving our problem. The challenges of deep learning are discussed in Section 9; Section 10 concludes the paper and points out the focus of future work.

6. Deep learning models

In this section, we survey the basic of deep learning-based models and discuss their architectures and features. At present, the deep learning mainly includes stacked automatic encoder, deep belief network, deep Boltzmann machine, convolution neural network and so on. The following is a brief introduction of basic models.

6.1. Stacking automatic encoders

1. **Automatic Encoder:** Auto Encoder (AE, Auto Encoder) [31] mainly consists of the encoder, the decoder and the hidden layer. The working process is shown in Fig. 3. An automatic encoder firstly encodes the input signal and then uses the coded signal to reconstruct the initial signal. This coded signal can minimize the error between the initial signal and the reconstructed signal. In the process of the encoding and the reconstruction, the encoder maps the input data to a specific feature space. The characteristics of the encoded

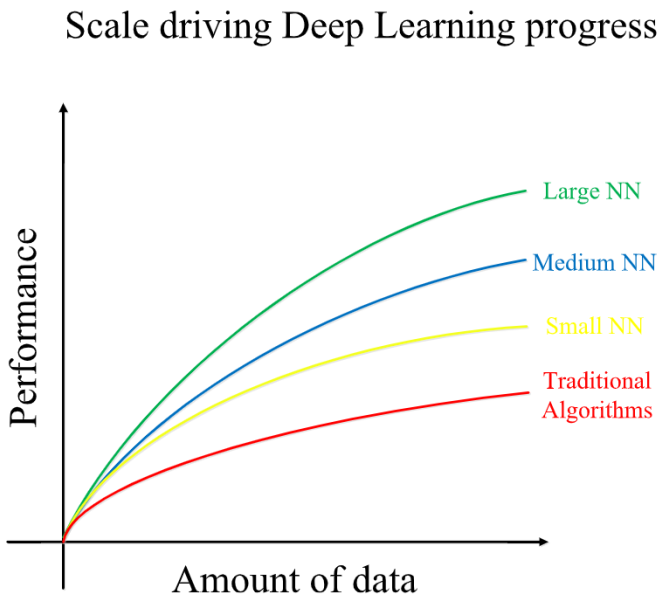

Fig. 2. This figure shows how the performance of Deep Neural Network improves with data size.

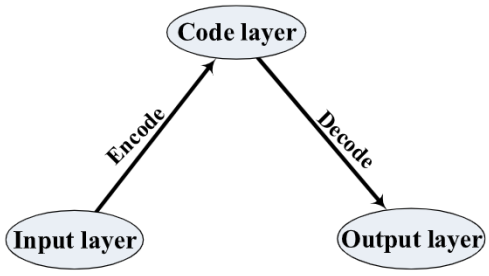

Fig. 3. Schematic diagram of the automatic encoder.

signals are mapped back to the data space by the decoder, and then the initial data is reconstructed. For the automatic encoder, the mapping is often concerned with the input that will be encoded. If there is a difference between the forced coding data and the input data, then the system can restore the initial signal in a different form. And thus, the features are extracted so they can achieve automatic learning.

2. **Stacking automatic encoders:** In 2006, Hinton et al. upgraded the structure of the encoder to improve the proposed de-noising automatic encoder (DAE, Denoising Auto Encoder [32]), after the researchers gradually put forward shrink automatic encoder, Contractive Auto Encoder, Spare Auto Encoder, Convolutional Auto Encoder, and so on. The above automatic encoders are stacked automatic encoders. The stacked automatic encoders are deep network structures formed by the n-times superposition of simple auto-coding structures. The realization of the stacked automatic encoder is presented as shown in Fig. 4. As shown in Fig. 4, let n automatic encoders which are trained from bottom to top. Firstly, the first autoencoder is trained, and the initial reconstruction error is minimized. Secondly, the output of the first autoencoder is trained as the input of the second encoder, which is operated until the last layer. Then, the output of the last layer is used as the input data of the classifier, and its parameters are re-initialized. Finally, based on the standard of the supervision, only the top is fine-tuned, or all the layers are appropriate fine-tuned.
3. **Restricted Boltzmann machine:** For a bipartite graph, if there is no link between the first layer and the second layer, then the first layer is consider as the input layer (i.e., it is the visual layer), and the second layer is consider as a hidden layer [33]. Let us suppose that all the nodes are random binary variables. Moreover, the full probability distribution $p(v, h)$ is subjected to the Boltzmann distribution which is called the Boltzmann machine [34]. The specific model is shown in Fig. 5. According to the characteristic of the restricted Boltzmann machine, the activation conditions of the hidden layers are independent for a given state of the visible layer (i.e., the input data). So, for the state of a given hidden layer, the activation conditions of the visible layers are independent. Though the distribution of the restricted Boltzmann machine cannot be calculated effectively, but a random sample can be obtained by Gibbs sampling. This random sample is subjected to the constrained Boltzmann machine. As long as the number of the hidden layers are sufficient, the Restricted Boltzmann Machine can fit any discrete distribution. In terms of the application, the restricted Boltzmann model has been successfully used to

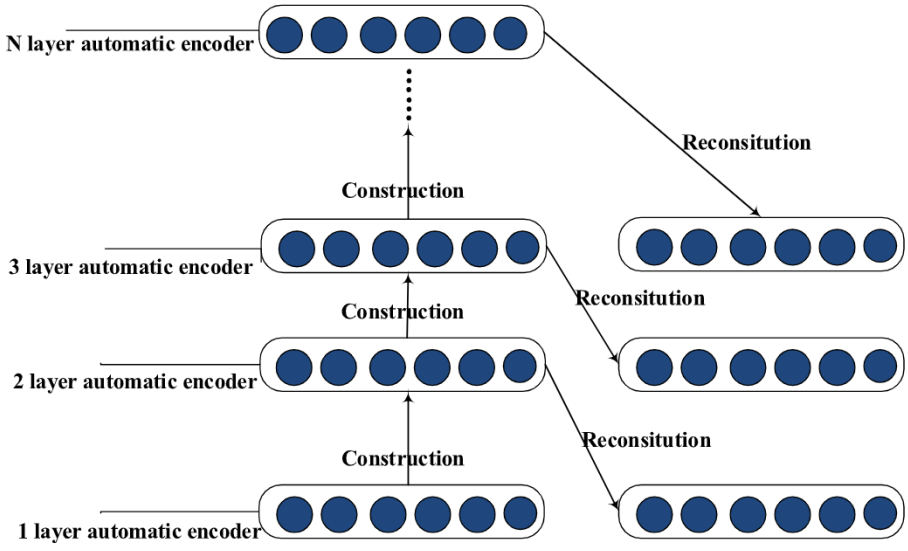

Fig. 4. Stacked automatic encoders.

solve various machine learning problems such as regression, classification, dimensionality reduction, time series modeling, image co-filtering, and feature extraction.

4. **Deep Belief Network:** The Deep Belief Network (DBN) [35, 36]is composed of a superposition of multiple constrained Boltzmann models with hidden explanatory factor neural networks of multiple layers. A typical neural network is shown in Fig. 6. The network layer remains the connection between the layers, but the connection in each layer does not exist. The data dependency exhibited in the visual layer unit is captured by the training of the hidden layer.
   As shown in Fig. 7, in the process of deep belief network training, firstly, the pre-training is carried out by an un-supervised greedy method to obtain the eigenvalues of the model layer by layer. The un-supervised greedy layer method is called Contrastive Divergence and it has been proved as a valid method. During the training process, the visual layer generates a vector v. It passes the data to the hidden layer through the vector v. Conversely, the input of the visual layer is randomly selected to attempt to reconstruct the original input signal. After that, the neuron activation unit of the new visual unit can continue forwarding the transfer input in order to reconstruct the hidden layer activation unit and then obtains the vector h. These repetitive processes are also known as Gibbs sampling. The correlation between the input of the visual layer and the activation unit of the hidden layer is the most important basis for measuring the weight update. The restricted Boltzmann machines are trained for each layer from bottom to top.
   The top-level accesses are concatenated so that the output in the bottom layer can provide an association to the top layer, which can associate its memory content and ultimately a discriminate performance obtained. After the pre-training is completed, the neural network gets a good initial data. However, this is not the optimal solution. The Deep belief Network uses the tagged data to adjust the discriminated performance by the error back propagation (BP), meanwhile, adding a label set to the top layer. Through repetitive learning, the identification weight will get the classification of the network, which will be stronger than the single error back propagation algorithm, and the training time is shorter than the feed-forward neural network. As an important turning point in the deep learning, the emergence of the deep belief networks are utilized in various areas such as voice recognition, image processing and so on.
5. **Deep Boltzmann Machine:** Deep Boltzmann Machine [37] (DBM, Deep Boltzmann Machine) is also formed by the restricted Boltzmann machine stack, which is similar to the deep belief network. The difference between the deep Boltzmann machine and the deep belief network is that the former layer and the current layer are between the non-directional connections, and there are no feedback parameters from top to bottom. The deep Boltzmann machine training method first uses unsupervised pre-training to get the desired initial authority and then it uses the field-averaging algorithm. Finally, the supervised fine-tuning is carried out. The deep Boltzmann machine is different from other models. Firstly, the deep Boltzmann machine has the ability to learn more complex intrinsic representations, which is a new way of speech recognition and object recognition. The deep learning can significantly improve the performance in the field of voice recognition. Secondly, the deep Boltzmann machine can build a higher representation in a large number of non-tagged data. To achieve the desired value, the deep Boltzmann uses the known artificial tagged data to fine-tune the model. Also, the deep Boltzmann machine can be more robust to deal with more vague input data information, and it can be spread better, which reduces the error in the process of dissemination.

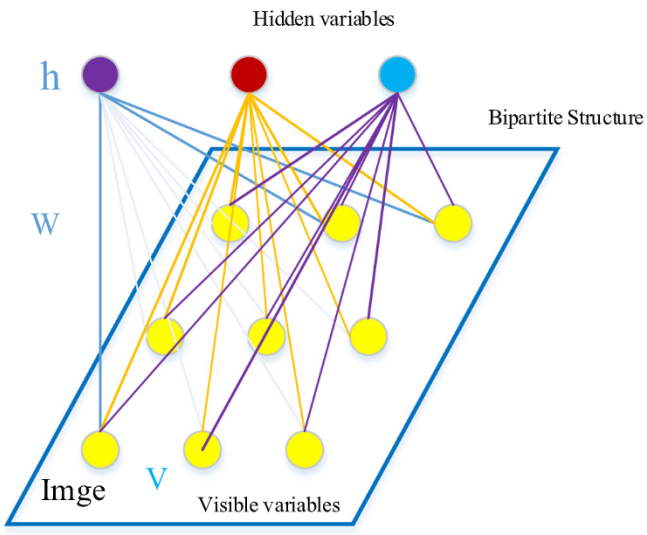

Fig. 5. Restricted Boltzmann machine binary map.

6.2. Convolution neural network

1. **Overview of Convolution neural network:** Convolution neural network [38] (CNN, Convolutional Neural Networks) was first proposed in the 1980s. It is inspired by the cat's cortex [39]. The LeNet-5 system was a classical model of a convolution neural network. Its error rate was only 0.9% on the MNIST data-set. It had been widely used to identify handwritten checks on banks, but it did not recognize large images. With the development of Graphics Processing Unit (GPU) technology, Krizhevsky et al. [40] used an efficient GPU supported program to solve the ImageNet problem in 2012, which made the convolution neural network again popular. In fact, one of the bottlenecks of deep neural nets was that it took a long time for training because of the many hidden nodes in its network. But as the GPUs become faster in parallel computing, this bottleneck was overcome. At present, the convolution neural network is a hot topic in the field of voice data analysis and image recognition. The convolution neural network has a network structure with share permission, which makes it closer to the biological neural network. This network structure in the convolution neural network can effectively reduce the complexity of the network model and also reduce the number of the weights. Mainly, it is more efficient to deal with high-dimensional images, which can directly consider the image as the input of the entire network and effectively avoid the complex feature extraction and reconstruction of the traditional algorithm. In the process of image recognition, the convolution neural network has a high degree of invariance in scaling, tilting, translating, and other forms of image deformation.
2. **Structure of convolution neural network:** As a multi-layer neural network, each layer in the convolution neural network structure is composed of a number of two-dimensional planes and each plane has independent neurons. The sparse connections are used between the layers.

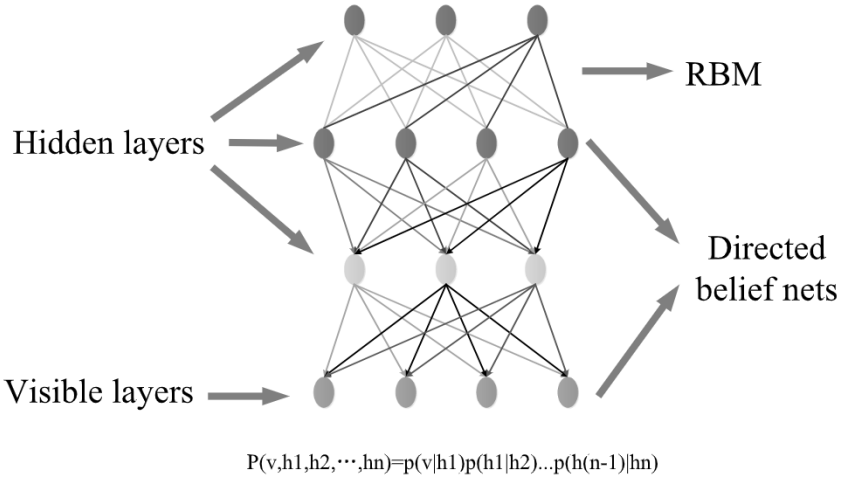

Fig. 6. Structure of deep belief network.

This means that the neuron in each feature map only connects the neurons in a small area in the upper feature map, rather than the traditional neural network. A typical model of the convolution neural network is shown in Fig. 8. The convolution neural network structure mainly depends on the shared weight, the local experience field, and the sub-collector to ensure the invariance of the input data. These factors can be explained as follows: Local experience field: The first hidden layer contains six feature maps (see Fig. 8). Each feature map corresponding to a small box in the input layer is a local experience field, or it is called the sliding window.

Convolution: The activation value a_j^l of the j^{th} feature map in the convolution layer l, which is expressed in the following equation:

$$a_j^l = f(b_j^l + \sum_{i \in M_{jl}} a_i^{(l-1)} * k_{ij}^l), \tag{1}$$

Where f is a non-linear function, $*$ is a two-dimensional convolution operator, b_j^l is the jth offset in the lth layer, and is the weight, which is a cumulative input in the feature map i of $l-1$th layer. The index vector of i is M_{jl} in the feature map of $l-1$th layer. The feature map needs to be accumulated in the lth layer. Weight share: Each convolution layer is usually composed of several feature maps and the weight of the same features in Fig. 8 is same, which can reduce the number of its own parameters.

Sub-sampling: The translation of the convolution layer will also translate its output at the same time. However, its characteristics remain unchanged, and as long as a feature is detected, its exact position will no longer be considered, as only the relative position of the other features can be preserved. So, each convolution layer has a sub-sampled layer that performs local averaging to reduce the sensitivity associated with the deformation and the translation of the output. The feature mapping of the subsampling layer is denoted by Eq. (2) as follows:

$$a_j^l = down(a_i^{(l-1)}, N^l), \tag{2}$$

Where N^l is the boundary size of the subsurface required for the l sub-sampling layer and $down$ is the down sampling function in the factor. The above formula 2 is the mean operation of the localized non-overlapping parts of size. If the neuron output layer is C-dimensional, then the C class can be identified, and the output layer which is the output characterization for front connection feature mapping is expressed in Eq. (2) as follows:

$$output = f(b^0, W^0 f_v) \tag{3}$$

Where b^0 is the partial value vector, W^0 is the weight matrix, f_v is the eigenvector, and k_{ij}^l, b_j^l, b^0, W^0 are the model parameters. As the convolution neural network structure is mainly alternately composed of convoluted layer and sub-sampling layer. With the reduction of the spatial resolution, the number of the feature maps is also increasing. The training process of the convolution neural network is as follows: The first stage is the forward training phase. It consists of the following three steps:

- Select the samples according to the given sample set randomly.
- Put the samples as initial data into the network.
- Calculate the corresponding output data.

The second stage is the backward propagation phase, It consists of the following two steps:

- Calculate the difference between the ideal data information and the output data information.
- Adjust the weight matrix according to the minimization of the error method for the reverse transmission.

6.3. Deep learning on graphs

Deep learning or traditional machine learning takes data into the form of vectors and considers it into the Euclidian plane. The graph data set is dissimilar from other data sets such as image, audio, etc. The graphs have the following characteristics, which explain the failure of the traditional machine learning approach [41].

- **Irregular Domains:** As previously mentioned, the graphs represents irregular domains or non-Euclidean data, while other data sets such as image and audio can be easily represented in the Euclidean plane or grid like structure. This explained the reason why many mathematical operations cannot be directly applied on the graph data [42].
- **Non-static structure:** Graphs are tools to represent complex systems. Therefore, they might have different shapes and structures such as homogeneous, non-homogeneous, signed, unsigned graphs, and so on. The graphs may also be different such as node centric (i.e., link prediction, node ranking,

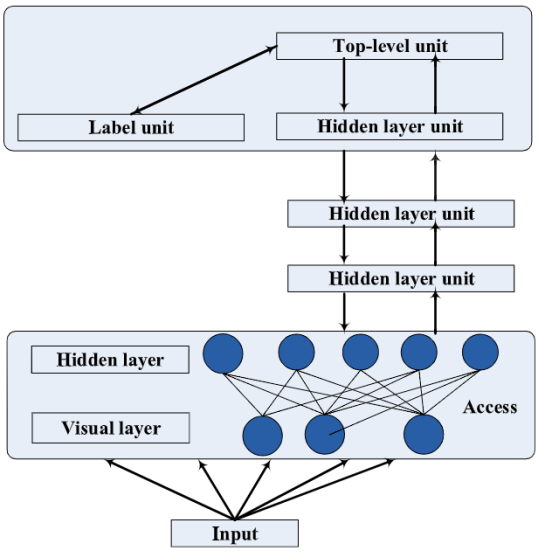

Fig. 7. Training process of deep belief network.

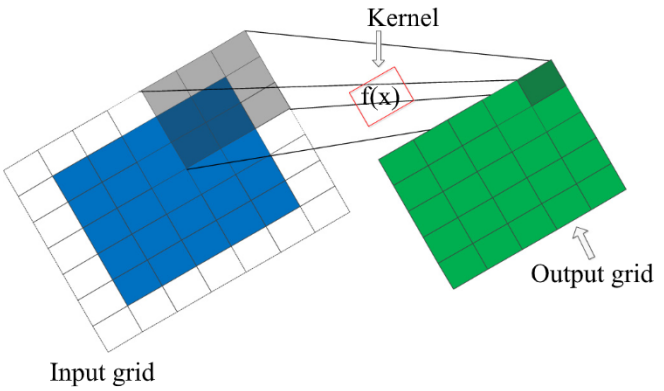

Fig. 9. The picture shows how convolutional operator works on grid like structure and generates output grid.

etc.), graph centric (e.g., graph generation, graph classification, etc.) and so on. The most utilised graph representation method is using adjacency matrices. Which changes its shape after addition or deletion of nodes. That's the reason ML models cannot take adjacency matrices directly.

- **Scalability and Parallelization:** In the computational tools abundance era, the first problem we have is the big data problem. In consequence, the generated graphs might have millions of nodes and billions of edges. For example, Google web page link data graph causes hindrance for training machine learning models especially with many nodes and hidden layers. The second problem is that how to parallelize the algorithms since every node in the graph carries some information about the other nodes in the graph, i.e., the nodes have some relations with other nodes, which should not be missed since losing such information might lose vital information.
- **Domain specific knowledge:** Learning on graphs might also need to be aware about domain specific knowledge such as drug–target interaction prediction task where the drug chemical molecular structure may help for better prediction. The other extra information might be helpful in the prediction about drug–drug side effect as a feature.

1. **Graph Neural Networks:** Graph Neural Network is a kind of neural network that takes the input as a graph data, not as a vector. It learns to represent the features for every node i. The further generated features can be used in any graph-related problem such as node classification, graph classification, clustering, and so on. In the node classification problem, every node is characterized by its feature x_i and it has some associated label l_v. In the graph classification problem, there is a set of nodes which have associated label l_i. By learning the features of the nodes i, the graph neural network has to predict the label for the unknown nodes i. It learns to represent each node in d dimensional vector v_i. The vector v_i contains information about the neighbor nodes of the node i as presented in the following equation (4) [43]:

$$V_i = f(X_i, X_{co[i]}, V_{ne[i]}, X_{ne[i]}) \tag{4}$$

Where, $X_{co[i]}$ represents the features of the edges adjacent to the node i, is the embedded neighboring node of the node of i, and f is a transition function (feed-forward neural network), which output d dimensional vector. The above formula can be solved using the neighborhood aggregation theorem method as it can be rewritten in an iterative form as follows:

$$V^{t+1} = F(V^t, X) \tag{5}$$

The further output transition function O_i is applied to get the final low dimensional vector as follows:

$$O_i = g(V_i, X_i) \tag{6}$$

The further hidden parameters are learned by applying the loss function between the predicted output O_i and the true labels l_i. Based on the graph neural network, many derived deep learning models are developed, such as Graph Convolutional Neural Network (GCN) [44] and GaphSage [45] etc. These models are the state of the art models and based on the graph neural networks.

2. **Graph Convolutional Networks:** A graph convolutional neural network operates in three steps.

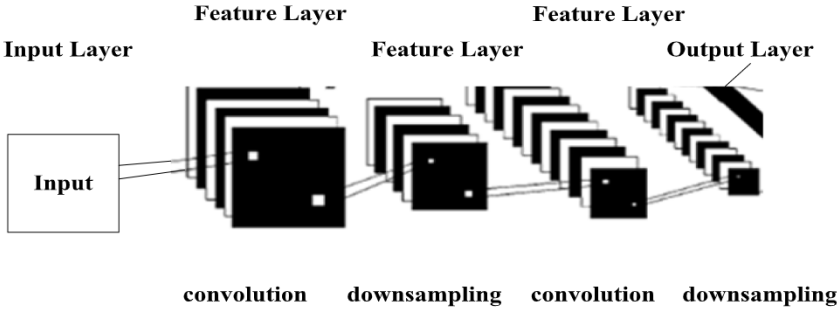

Fig. 8. Convolutional Neural Network model.

- **Kernel/filters:** A kernel/filter is a function that acts like a scanner that has a limitation on the number of pixels or cells (of adjacency matrices) needs to be considered at a time, i.e., at one-time, the scanner function works on a small part of the grid. Fig. 9. shows how the kernel function is applied on an input grid.
- **Pooling:** Similar to the kernel scanner function, pooling is a function that gives an output for all the values scanned by the scanner at a time. This output value can be based on max pooling (i.e., an important element or average value). It is also known as mean pooling (as it can be seen from Fig. 9). It can be noticed from many values in the input grid that only one output cell is generated after applying the kernel followed by the pooling function.
- **Flattening:** Flattening function truncates the grid structure into a lower-dimensional vector, which can be used to feed the forward neural networks.

The above three methods are common to all graph convolutional neural networks. The big difference comes only in the different kernel functions across the graph neural networks. However, the researchers have classified the graph convolutional neural networks into two types as follows:

- **Spatial Methods:** These kinds of convolutional operations do not need eigenvalues of the graphs.
- **Spectral Methods:** These kinds of methods are based on eigenvalues and they are more popular than the spatial methods as they consider both aspects, i.e., the whole graph structure as well as the individual graph components. There is a subfield known as Graph Signal Processing (GPS) which is based on signal processing techniques such as Fourier transformation etc. Some of the state-of-the-art works in this field are ChebNets [46], and Kipf and Welling [44].

6.4. Deep probabilistic neural networks

Although probabilistic neural networks (PNN) are there for quite long time [47]. But these were based on shallow neural network architecture. The PNN network has four basic layers, the input layer: that grabs and distributes the input vector; the pattern layer: that applies the kernel to the input; the summation layer: that gets the average of the output of the pattern units for each class; and the decision layer: that declares the class assigned to input vector based on the unit with the maximum output from the summation layer [47]. Very recently deep probabilistic neural network(DPNN) has been introduced [48]. The advantage of PNNs over neural networks is that they can perform better even with less training data [47]. Therefore they are found useful in many cases where we usually lack of lots of training data such as medical imaging, signal processing etc. Further PNNs are better under adversarial attack which makes it promising choice as DNNs fails even under minor random errors. Gast et al. [48] provide a deep probabilistic neural network by altering a little to the current neural network. The applied following two approaches to achieve this:

- The first and simplest consists of replacing the output layer of well-proven networks with a probabilistic one see Fig. 10(b).
- The second alteration goes by considering activation uncertainties also within the network by means of deep uncertainty propagation see Fig. 10(c).

6.5. Deep fuzzy neural networks

Although current neural networks outperforms even human as a benchmark for some problems such as image recognition but still they lack many aspects. One of the aspect is interpretability and expansibility of the models. One cannot know or explain what is going on inside the neural network parameter estimations. Therefore we use it as a black box machines. Some networks such as Deep Convolution networks stack upto hundreds and thousands of layers together to solve image recognition problem. It consist of millions or billions of internal parameters. Although their performance is very good but we cannot explain all those layers or parameters by our logic. Which is the reason even slight adversarial attack on image it predicts completely different response. Even random label can cause deep neural networks to over-fit and affect the test performance very bad [49]. Neural networks lack logical reasoning therefore it is prone to adversarial attack. For example Alpha Go lost one game to South Korean Go player because its 78th move was not decisive. Further this move led to sequence of moves. Neural networks gives unpredictable and un-interpretable results. In an example despite the advances sensors and cameras the Neural network failed to detect the pedestrian at shadowed street [50]. Considering the logical reasoning aspect of Fuzzy logic some research proposed Deep fuzzy neural networks [51] by fuzzing the two systems together. They proposed hierarchical approaches to fuse the fuzzy logic and neural network that simultaneously leaned feature representations altogether for robust data classification. Further Zhou et al. first transform the input vector into latent vector using neural network then fuzzifies the representation at the output layer for pattern classification [52].

6.6. Generative Adversarial Networks (GANs)

These are the class of generative models based on game theory. Which do not explicitly model the data distribution but rather models the sample from it. Sampling is performed using a deep neural network. The neural network takes as input random noise and transform it into model distribution. Suppose we have examples of sample data $P_{data}(x) \rightarrow \{x_i\}, i = 1...N$. We need to find model which approximate the given data i.e. $P_{\text{model}}(x) \sim P_{data}(x)$. $P_{\text{model}}(x)$ is not parametric model but it is accomplished using deep neural network [53,54]. Generative Adversarial Network consists of two neural networks. One is called Generator and another one is called Discriminator. This model is called adversarial because the generator is constantly trying to fool the discriminator into believing that input is from training data(real data). While discriminator always distinguishes between the two.

1. **Generator:** A neural network that takes as input, a random noise vector and transform it into a model distribution.
2. **Discriminator:** It is a neural network that distinguishes between output data point (Fake) and training data samples (Real). It acts like a classifier as if the input is real or fake. These two neural networks trying to work against each other. In these setting the weights of generator learns that converts a random noise vector into a model distribution. From Fig. 11 generator G takes a random noise vector from latent space and out puts some samples. Now discriminator D takes input from training data (real) and checks against the generated fake sample from generator (G). The training data should have images from the similar kinds of tasks say paintings or faces etc. Upon taking both the input the error function outputs probability that particular sample is real or fake. This output is used to train the weights of the generator as well as the discriminator. The another important part is formulation of error function or cost function in GANs. This problem is formulated as MiniMax zero sum game.

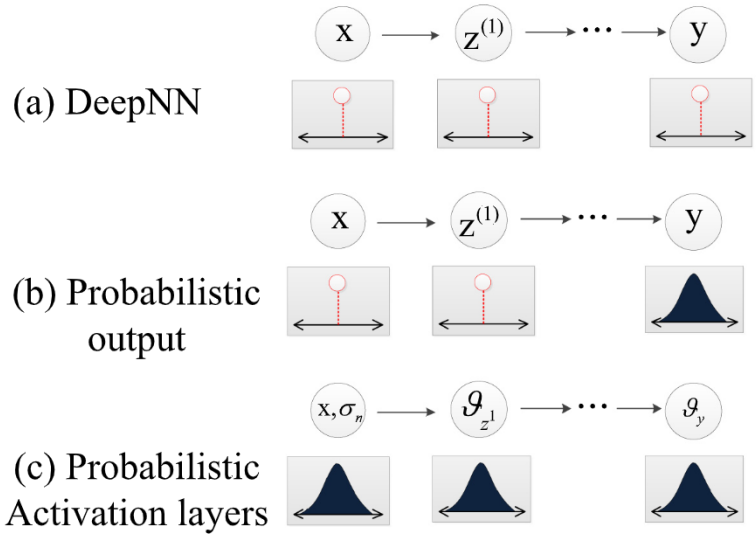

Fig. 10. (a) Shows the deterministic neural network. (b) Shows how final layer changes to probabilistic out put. (c) Shows how intermediate layers activation value changes to distributions.
Source: Image from [48]).

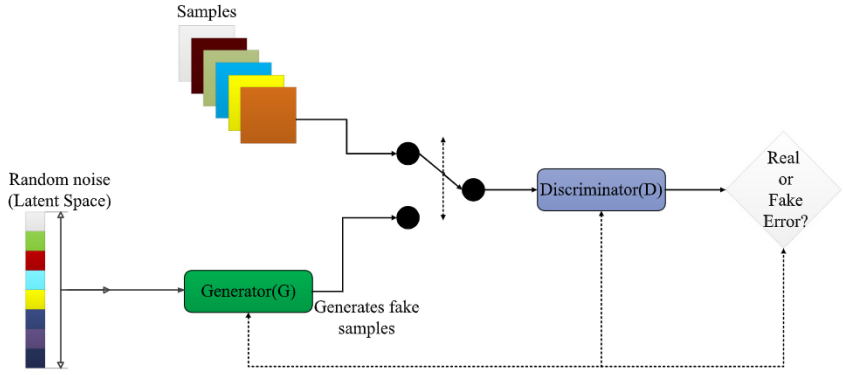

Fig. 11. How Generative Adversarial Networks works.

7. Applications of deep learning

In this section we covered applications of deep learning in various areas. Further we summarized it in one Table 1. Following are the various applications of Deep learning.

7.1. Natural language processing:

In natural language, the deep learning is applied in many areas such as voice translation, machine translation, computer semantic understanding, and so on. In fact, the success of deep learning is only in two fields, i.e., the image processing and the natural language processing. In 2012, Schwenk et al. [55]proposed a phrase-based statistical machine translation system based on Deep Neural Network (DNN). It was able to learn meaningful translation probabilities for unseen phrases which were not presented in the training set. In 2014, Dong et al. [56] proposed a novel Adaptive Multi-Compositionality (AdaMC) layer in the recursive neural network. This model introduced more than one composition function, which was adaptively selected based on the input features. In 2014, Tang et al. [57] presented a DNN on Twitter data for sentiment analysis. In 2015, Google introduced Word Lens recognition technology based on deep learning, which used word lenses in real-time call translation and video translation. This technology not only could read the words in real-time, but also those words could be translated into the desired target language. Also, the translation work could be done through the phone without networking. The current technology could be applied in more than a visual translation of 20 languages. In addition, Google proposed an automatic mail reply function in Gmail, which used a deep learning model for extracting the e-mail content and analyzing it semantically. Finally, a reply is generated based on the analysis of the semantic. This technique is fundamentally different from the traditional e-mail auto-respond functionality.

7.2. Speech recognition:

In order to realize the Human–Computer Interaction, the researchers made great efforts. In 1952, Bell Institute's Davis and others successfully developed the world's first experimental system which can identify 10 English digital pronunciations. The research on speech recognition technology has few decades of history, and voice recognition was the dictator used in certain areas as it was mentioned by the US press as one of the top ten events in computer development. In the last two decades, speech recognition technology has made significant progress. With the continuous improvement of the deep learning model, a large number of speech recognition devices or applications have begun to move from the laboratory to the market. In 2014, Baidu launched Deep Speech, a voice recognition system with deep learning technology, which can achieve 8% accuracy in noisy

Table 1
Applications of deep learning.

Application field	Reference	Methods	Task
Natural language processing	Schwenk et al. 2012 [55]	DNN-based	Phrase-based translation
	Dong et al. 2014 [56]	AdamMC +RNN based	Sentiment analysis and semantic composition
	Tang et al. 2014 et al. [57]	COOOL: DNN-based	Sentiment classification
Speech recognition	You et al. 2015 [58]	DNN-based	Speech recognition
	Maas et al. 2017 [59]	DNN-based	Speech recognition
Medical applications	Li et al. 2014 [60]	CNN-based	Lung disease identification
	Li et al. 2015 [61]	DNN-based	Alzheimer's disease classification
	Sirinukunwattana et al. 2016 [62]	SC-CNN and NEP	Cancer disease classification
	Dou et al. 2016 [63]	3-D CNN	Cerebral microbleeds identification
Semantic segmentation & Scene labeling & Face recognition	Krizhevsky et al. 2012 [40]	CNN-based	Image detection
	Malik et al. 2012 [64]	Bayesian network	Ontology learning
	Behnke et al. 2014 [65]	RGB-D sensor based on the deep learning technology	Semantic scene segmentation
	Sun et al. 2014 [66]	DeepID	Face recognition
	Pinheiro P et al. 2014 [67]	CNN-based	Scene labeling
	Taigman et al. 2014 [68]	DeepFace	Face recognition
	Long et al. 2015 [69]	FCN	Semantic segmentation
	Schroff et al. 2015 [70]	FaceNet	Face recognition and clustering
	Wang et al. 2015, [71]	CNN-based	Pixel-wise semantic segmentation
	Zheng, 2015 [72]	CNN-based	Semantic segmentation
	Ronneberger et al. 2015 [73]	U-Net (CNN-based)	Biomedical Image Segmentation
	Badrinarayana et al. 2015[74]	SegNet (Convolutional Encoder–Decoder)	Semantic Pixel-Wise Labeling
	Liu et al. 2015 [75]	DPN(CNN)- and MRF-based	Semantic segmentation
	Byeon W et al. 2015 [76]	2D	pixel-level segmentation and scene labeling
	Lin et al. 2016 [77]	CRF- and CNN-based	Semantic segmentation
	Shen et al. 2016 [78]	CRF- and CNN-based	Semantic image segmentation
	Chandra et al. 2016 [79]	GCRF based	Contextual relations between parts of the image.
	Luc et al. 2016 [80]	GAN- and CNN-based	Semantic segmentation
	Hoffman J et al. 2016 [81]	FCN-based	Semantic segmentation
	Shuai B et al. 2016 [82]	DAG-RNNs	Scene labeling
	He et al. 2016; [83]	ResNet	Image recognition
	Chen et al. 2017 [84]	CNN-based	Semantic segmentation
	Koziski M et al. 2017 [85]	GAN based	Semantic segmentation
	Chen et al. 2017 [86]	CNN-based	Semantic segmentation
	Souly et al. 2017 [87]	GAN based	Semantic segmentation
	Yu et al.2018 [88]	CNN-based	Semantic segmentation
	Marvin et al. 2018 [89]	CRF- and CNN-based	Semantic segmentation
Object Detection	Karen et al. 2015 [90]	CNN-based	Oject detection
	Pierre et al. 2014 [91]	CNN-based	Object localization and detection
	Russakovsky et al. 2015 [92]	CNN-based	Object detection
	Chatfield et al. 2015 [93]	CNN-based	Object detection
	Pi et al. 2020 [94]	CNN-based	Object detection in aerial imagery.
	Gu et al. 2020 [95]	CNN-based	Object detection in X-ray images.
Video object segmentation	Caellas et al. 2017 [96]	OSVOS: CNN-based	Moving object detection in videos
	Shin et al. 2017 [97]	CNN-based	Moving object detection in videos
	Jang et al. 2017 [98]	convolutional trident network	Moving object detection in videos
	Hu et al. 2017 [99]	MaskRNN	Instance level video object segmentation
	Sasikumar et al. 2018 [100]	Mask R-CNN	Moving object detection in videos
	Li et al. 2018 [101]	CNN-based	Video foreground target extraction
	Xiao et al. 2018 [102]	MoNet	Moving object detection in videos
	Goel et al. 2018 [103]	MOREL	Moving object detection in videos
Background/foreground separation	Schofield et al. 1996[104]	DNN-based	Object detection in videos
	Tavakkoli et al. 2005 [105]	DNN	Foreground and background separation in videos
	Culibrk et al. 2006 [106]	DNN-based	Background modeling
	Maddalena et al. 2007 [107]	Self organization+DNN	Background modeling
	Ramiraz et al. 2013 [108]	RESOM	Background separation in videos
	Guo et al. 2013 [109]	PS-RBM	Background modeling
	Xu et al. 2014 [110]	Auto-encoder Networks	Background modeling
	Xu et al. 2014 [111]	Auto-encoder Networks	Background modeling
	Ramirez et al. 2015 [112]	Self-organizing Maps (SOMs) and Cellular Neural Networks (CNNs), CNN-based	Dynamic object detection
	Qu et al. 2016 [113]	Context-encoder: CNN-based	Background modeling
	Minematsu et al. 2018 [114]	DNN-based	Background modeling
	Ammar et al. 2019 [115]	DeepSphere	Foreground modeling
	Sultana et al. 2020 [116]	GAN-based	Background modeling

(*continued on next page*)

environments. In February 2016, Baidu's Deep Speech 2's error rate of phrase recognition had been reduced to 3.7%. In 2015, You et al. [58] proposed a node pruning method to reconstruct the DNN which gave a novel bottleneck feature. Further, in 2017,

Table 1 (*continued*).

Application field	Reference	Methods	Task
Graphs based applications	Duvenaud et al. 2015 [117]	GCNN-based	Molecular property generation
	Kearness et al. 2016 [118]	Molecular Graph Convolution: GCNN-based	Molecular property prediction
	Berg et al. 2017 [119]	Graph Auto Encoder-based	Link prediction
	Monti et al. 2017 [120]	Multi-Graph CNN	Matrix completion
	Gilmer et al. 2017 [121]	MPNN	Molecular property prediction
	Coley et al. 2017 [122]	GCN-based	Molecular Graph embedding generation
	Ktena et al. 2017 [123]	GCNN-based	Graph similarity prediction
	Parisont et al. 2017 [124]	GCNN-based	Brain disease prediction
	Parisont et al.2018 [125]	GCNN-based	Brain disease prediction
	Qui et al. 2018 [126]	DeepInf: GNN based	Social influence prediction
	Ying et al. 2018 [127]	PinSage: GCN- + RW-based	Web-based recommendation
	You et al. 2018[128]	GCPN: GCN-based	Molecular graph generation
	Cao et al. 2018 [129]	MolGAN: GAN-based	Molecular graph generation
	Zitnic et al. 2018 [130]	Dacagon:GCNN-based	drug–drug side effect prediction
	Xie et al. 2018 [131]	CGCNN: GCNN-based	Material property prediction
Intelligent transportation system	Moreira et al. 2013 [132]	Time varying Poisson and ARIMA	Destination prediction
	De et al. 2015 [133]	Bi-directional CNN+NN	Destination prediction
	Vinyals et al. 2015 [134]	RNN-based	Demand serving
	Li et al. 2015 [135]	Graph-based DL	Routing problem
	Bello et al. 2016 [136]	RNN-based	Demand serving
	Zhang et al. 2016 [137]	CNN-based	Traffic flow prediction
	Chen et al. 2016 [138]	stacked Autoencoder-based	Predicting traffic accident severity
	Endo et al. 2017 [139]	RNN-based	Destination prediction
	Ke et al. 2017 [140]	LSTM+CNN-based	Demand Prediction
	Yao et al. 2017 [141]	a Deep Multi-View Spatial–Temporal Network (DMVST-Net)	Demand Prediction
	Khalil et al. 2017 [142]	Graph-based DL	Routing problem
	Ma et al. 2017 [143]	CNN-based	Traffic flow prediction
	Jiang et al. 2017 [144]	RNN-based	Transportation mode
	Yao et al. 2017[145]	Auto-Encoder-based	Trajectory clustering
	Yang et al. 2018 [146]	Network/Graph-based DL	Navigation
	Jindal et al. 2018 [147]	Spatio-Temporal Neural Network+ reinforcement learning	Travel time estimation
	Li et al. 2018 [148]	Network-based DL	Travel time estimation
	Kool et al. 2018 [149]	Attention Model	Routing problem
	Lv et al. 2018 [150]	CNN-based	Destination prediction
	Yuan et al. 2019 [151]	CNN-based	Seen recognition
	Li et al. 2019 [152]	LSTM-based	Traffic route planning

Maas et al. [59] analyzed different architectures and parameters of the DNN for training a very large speech data. They found simple architecture and simple optimization methods that gave strong performance than the other very complicated models.

7.3. Medical applications

The forecast function of deep learning and its automatic feature identification makes it popular technique in disease diagnosis also. The applications of deep learning in medical field, either in the use of frequency or in the use of species are constantly upgrading. In 2014, Li et al. [60] proposed customized CNN to classify lung image patches. This model used the dropout method and single-volume structure to avoid overfitting. In 2015, Li et al. [61] proposed a DNN-based framework to differentiate the identity stages of Alzheimer's Disease (AD) from the MRI and PET scan data. In 2016, Srinukunwattana et al. [62] proposed a spatially constrained convolutional neural network (SC-CNN) to analyze the histopathology images and identify the nucleus of the cancerous cells. Their SC-CNN method had better performance than the classical feature classification method. In 2016, Google developed a vision system for identifying early-stage ocular diseases. They worked with the Moorfields Eye Hospital, such as diabetic retinopathy and age-related macular degeneration to provide early prevention methods. A month later, Google used deep learning techniques to design a head and neck cancer radiotherapy method which had an effective control of the patient's radiotherapy time and it could minimize the radiotherapy of the patient's injury. With the continuous development of deep learning technology, the deep learning in the field of precision medical care will lead to more prominent contributions.

7.4. Computer vision

Computer vision is an essential application of artificial intelligence. It is an interdisciplinary field that deals with how computers can gain high-level understanding from digital images or videos. It can use computers and cameras to replace the human eye for the target object recognition, tracking, measurement, and for other visual problems. And then deal with the graphics so that the computer can achieve image processing capabilities even beyond the eye. In 2015, Baidu announced that it will refresh the performance for the ImageNet image classification recognition. In the test, the error rate of the image recognition was less than 5%, which was beyond the human level error for the first time in computer performance. Computer vision is a broad term which give birth to many research directions. Followings are some well known directions which comes under umbrella of computer vision.

1. **Image segmentation:** Throughout the previous thirty years, one of the most difficult issues in computer vision has been image segmentation. Image segmentation is not quite the same as image classification or item recognition in that it is not important to understand what the visual ideas or articles are. To be explicit, an object classification will just characterize objects that it has explicit labels for, for example, horse, auto, house, dog. An ideal image segmentation calculation will likewise fragment unknown objects, that is, objects which are new or unknown. There are various applications [64,93,153]where image segmentation could be utilized to improve existing algorithms from social legacy conservation to picture duplicate identification to satellite symbolism examination to on-the-fly

visual hunt and human–computer interaction. In these applications, approaching divisions would permit the issue to be drawn closer at a semantic level. For instance, in content-based image recovery, each picture could be portioned as it is added to the information base. At the point when a question is prepared, it very well may be fragmented and permit the client to inquiry for comparative portions in the information base, e.g., discover the entirety of the cruisers in the data set. In human–computer interaction, all aspects of every video casing would be segmented so the client could connect at a better level with different people and items in the climate. With regards to an air terminal, for instance, the security group is ordinarily keen on any unattended things, some of which could hold risky materials. It is useful to make inquiries for all articles which were given up by a human. The image segmentation problem can be stated as, given an image the algorithm should identify which two pixels are closely related see work by Pavlidis et al. from 1979 [154]. An ideal algorithm should cluster pixels [155] together according two object i.e. If there are two cars in an image then algorithm should separate pixels of cars from non-car pixels. There are many work has been done specially by utilizing Deep learning tools. Conventional algorithms segment image based on clustering and more information from contours and edges; Markov process is proposed by Geman et al. [156] in 1984; and so on. With the advancement of algorithms such as deep learning techniques image segmentation of digital media is becoming more popular and challenging. A survey of image segmentation on the basis of their strengths, weaknesses and major challenges by using deep learning techniques can be found in various application areas a reviews are presented in 1996 by Zhang et al. [157], in 2013 by Narkhedo [158], in 2014 by Kaur et al. [159], and in 2016 by Kuruvilla et al. [160].

2. **Face recognition:** Face recognition is a biometric identification technology based on the features of the human faces. Firstly, the camera is used to collect the video or image data containing the face, and then the collected video or image data is used to detect the image and the face automatically. In 2012, Huang et al. [161] presented a convolutional deep belief network for image segmentation problems. In order to exploit the global structural features, this model used the local convolutional restricted machine. In 2014, Taigman et al. [68] applied 3*D* face modeling to apply piecewise affine transformation for generating lower-dimensional features, which achieved 27% error reduction with respect to the state-of-the-art models. In 2014, Sun et al. [66] proposed deep hidden identity features (DeepID) for high-level representation generation from face data, which further could be easily used with the state-of-the-art classifiers. In 2015, Schroff et al. [70] proposed FaceNet based on a convolutional network, which considered face image in Euclidian space and generated low dimensional features. The face recognition accuracy of the latest deep learning algorithm (i.e., FaceNet) was 99.63%, which is more than the eye recognition. Generally, the deep learning can obtain the essential characteristics that the manual expression does not have. Such as deep learning is moderate sparse, which has a strong selectivity for face attributes and identification. Also, it has very good robustness for the local block. The face recognition features are obtained based on deep learning and the display constraint or post-processing is added in the model. This is the main reason that deep learning is widely used in the field of face recognition. The main technology of deep learning in face recognition includes the convolution neural network technology, the robustness modeling of deep learning face pose, the deep non-linear face recognition technology, the face recognition technology in the constraints of the environment, the low-resolution face recognition technology based on deep learning etc.
3. **Object detection:** Object detection is one of the fundamental and challenging problem in computer vision. And it has been an active research area since last few decades [162–166]. The goal of object detection is to find the given object category in the image or video e.g. human face, eyes, animal etc. As the foundational task of image understanding and computer vision object detection is the basis for solving many complex problems such as scene understanding, image captioning, event detection and activity recognition etc. Object detection has plethora of applications such as consumer electronics, robot vision, security, autonomous driving, human–computer interaction, automated surveillance and so on. As the image dimensions were very high so traditional algorithms were not very effective to learn pattern. Until recently when in 2006, Hinton et al. [167] found deep neural nets as very effective in automated feature learning from high dimensional images. In fact the credit for the success of deep learning goes to computer vision community see works in 2015, Russakovsky et al. [92], in 2015, Lecun et al. [168], in 2012, Krizhevsky et al. [40]. The object detection problem can be categorize into two types. The first one aim to find particular object such as face of football player Ronaldo, Eiffel tower etc. The other task is to find generic object of some category, probably unseen objects for examples cats, dogs and cars etc. Farmer problem is a bit harder than the later. Most of the research works has been done on the later problem. There are many review articles in this field mostly specific to the problem of interest. For example pedestrian detection, see works by Dollar et al. [169] in 2012, by Enzweiler et al. [170] in 2009, by Geronimo et al. [171] in 2009; vehicle detection see works by Sun et al. [172] in 2006, by Sakhare et al. in 2020 [173], and by Yuan et al. [174] in 2020; and face detection [175–177]. In 2019, Zhao et al. [178], an object detection method by using semantic segmentation and deep learning methods is surveyed. An application of semantic segmentation of the field of maritime surveillance is presented in 2018 by Cane et al. [179]. In the year of 2014, Girshick et al. [180] has been proposed a simple and scalable object detection scheme. The experimental result gives 30 percent improves mean average precision than previous state of the art method. In 2015, Girshick et al. [181] has designed a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection, where the performance of Fast R-CNN is higher than existing CNN-based architecture.
4. **Image semantic segmentation** An image contains a large amount of information. The semantic segmentation of the image is the process of dividing the image into a specific region and extracting the relevant target. The key to image processing and image analysis lies in what is called image semantic segmentation. In image segmentation task we segregate the pixels and cluster them according to some semantic relevance but we do not classify each pixels. In semantic segmentation each pixels are classified. In 2014, Liu et al. [182] reviewed some works on Probabilistic Graphical Model (PGM) for image segmentation and analysis. Further, in 2014, Hoft et al. [65] solved the image segmentation problem for getting depth information, which had greatly improved the image semantic segmentation. In

2014, Pinheiro and Collobert [67] proposed an end to end framework based on recurrent convolutional neural networks for scene labeling. In 2015 Long et al. [69] proposed an end to end Fully Convolution Network (FCN) semantic segmentation. However, the FCN method did not adequately consider the relationship between different pixels, which resulted in insufficient segmentation. In 2015, Beyon et al. [76] proposed Long Short Term Memory (LSTM) recurrent neural network based on end to end framework for pixel-level segmentation and classification. Later, in 2015, Zheng et al. [72] combined a convolutional neural network with Conditional Random Field-based graphical models (CRF-RNN) for image segment analysis in order to reach the pixel-level segmentation and classification task. In 2015, Yu and Koltun [88] proposed dilated CNN with CRF label modeling for image segmentation. In 2015, Ronneberger et al. [73] proposed U-Net for biomedical image segmentation, which relied on data augmentation. In 2015, Badrinarayana et al. [74] proposed Segnet using a convolutional encoder and decoder-based framework for image pixel-level analysis. In 2015, Liu et al. [75] proposed Deep Parsing Network (DPN), which solved the image segmentation problem by incorporating Markov Random Field (MRF). In 2016, Lin et al. [77] proposed a fully connected CRF with linear binary cliques, which helped in identifying similar image segments. In 2016, Shen and Zeng [78] proposed a framework that considered higher-order object-level features along with the discriminative features. In 2016, Chandra and Kokkinos [79] solved the image segmentation problem using Gaussian Conditional Random Fields (G-CRF) and Deep Neural Nets. In 2016, Shuai et al. [82] proposed Directed Acyclic Graph Recurrent Neural Network (DAG-RNN), which was able to model long-range semantic dependencies among image units. The DAG-RNN was also able to learn rare classes. In 2016, Luc et al. [80] proposed generative adversarial networks for image semantic segmentation, which could work on different kinds of images once trained. In 2016, Hoffman et al. [81] proposed unsupervised adversarial generative model, which returned this model to a domain adaptive model. In 2017, Chen et al. [84] combined the atrous separable convolutional pyramid pooling for semantic image segmentation problems. In 2017, Chen et al. [86] proposed DeepLab for image segmentation problem, which considered multiscale features via parallel filters with different dilation factors. In 2017, Kozinski et al. [85] proposed a generative adversarial network-based framework for image segmentation. In 2017, Souly et al. [87] further gave GAN (Generative Adversarial Network) based generative model for pixel-level annotation. The motivation behind this model was in adding a large amount of fake visual data which forced the real samples to be closed in feature space. In consequence, the bottom-up clustering approach helped in the multiclass pixel-level classification task. Up to some extent, it reduced the problems brought by the CNN, the FCN, and the other networks in the process of image semantic segmentation. With the continuous development of deep learning, the image semantic segmentation continues to develop in a more precise and faster direction. In 2018, Wang et al. [71] solved this problem by dense up the sampling of the convolutional framework. In 2018, Teichmann [89] proposed convolutional CRFs based framework by considering conditional independence, which helped in implementing CRFs on GPUs.

5. **Video object segmentation :** Due to rapid development of online social media video data is overwhelmed on Internet. In this environment finding object of interest inside video is really beneficial and demanding task. In object segmentation the pixels are divided into two subsets of the foreground target and the background region, and generates the object segmentation mask, which is the core problem of behavior recognition and video retrieval. Further object tracking are used to locate the location of object insider a video which is very helpful in intelligent surveillance. Object tracking and object segmentation are complementary to each other. As accurate object segmentation will help in object tracking and vice versa. Instance level object segmentation is popular in video processing where object identification, video editing, and video compression can be achieved. It is an interesting research direction recently [96,97,100,101,103]. A user annotations-based target object at the first frame by using semi-supervised online video object segmentation algorithm is presented in 2017 by Jang et al. [98], it is also known as CTN. The MaskRNN in 2017 by Hu et al. [99] is a novel recurrent neural net method for instance level segmentation in video. In this method by using recurrent neural net and the combination of segmentation and localization nets, the idea takes benefit of the long-term temporal information and the location prior to improve the results than some of the states of the art methods. In 2018, Xiao et al. [102] presented a novel trainable network MoNet introduces two motion exploitation components which are feature alignment and a distance transform layer to refine segmentation results.
6. **Background/foreground separation:** It is segmentation task in which algorithm split between background and foreground area of the image. This is currently hot topic as it has wide applications such as intelligent surveillance in public space, traffic monitoring, industrial machine vision and so on [183–187]. Recently neural network based models are also being applied for Background separation tasks see works in 1996, by Schofield et al. [104], in 2013 by Ramirez et el. [108], and in 2015, by Ramirez et al. [112]. In 1996, Schofield et al. [104] were the first to use neural networks to background and foreground separation tasks. They proposed Random Access Neural Networks (RAM-NN) which needs background information correctly represented. Further Tavakkoli in 2005, [105] proposed NN approach approaching it as novelty detection. The background is divided into blocks and each block is associated with Radial Basis Function Neural Network(RBF-NN). In 2006, Culibrk et al. [106] proposed feed forward neural network for background separation task which is based on Bayesian model. Although their work was supervised but it can be work unsupervised also. In 2007, Maddalena and Petrosino [107] came up Self Organizing Background Subtraction (SOBS) model which is based on a $2D$ self-organizing neural network architecture preserving pixel spatial relations. The weight vector of neural network is same as number of pixels i.e. $n \times n$. In this setting the background is modeled using neurons weight of the network. These models used simple neural network. Further deep neural network along with Convolutional neural network used and showed promising accuracy see works: by Guo et al. [109] in 2013, by Xu et al. [110] in 2014, by Xu et al. [111] in 2019, by Xu et al. [188]in 2015 and by Qu et al. [113] in 2016. For further studies in this area we will suggest to read these papers by Ammar et al. [115] in 2019, by Bouwmans et al. [189] in 2019, by Minematsu et al. [114] in 2018, and by Sultana et al. [116] in 2020.

7.5. Deep learning on graphs

In the recent years, the researchers are trying to develop new techniques which can effectively learn patterns from graph structured data. There are wide verities of problems which have been solved using deep learning on graphs. For example in 2018, Qiu et al. [126] presented an end to end deep learning framework for influential user prediction, which took the input from the user's local graph structure. In 2018, Ying et al. [127] proposed a graph-based recommendation framework which was based on the random walk and the graph convolutional neural networks. Their framework was suitable for largescale graphs. In 2017, Berge et al. [119] proposed a graph auto-encoder framework based on differential message passing mechanism, which helped the user–item interaction and bipartite graph completion. Also, in 2017, Monti et al. [120] presented a geometric deep learning framework which was based on convolutional neural network and recurrent neural network. This model helped in matrix completion problem by predicting accurate rating in the recommendation system. Further researchers also solved the deep learning and graphs problem in chemistry such as in 2015, Duvenaud et al. [117] presented a deep learning model for generating molecular features based on convolutional neural networks. In 2017, Gilmer et al. [121] developed a deep learning framework based on message passing neural network for molecular property prediction. In 2016, Kearnes et al. [118] developed molecular graph convolutional neural network which worked on undirected molecular graph. In 2018, You et al. [128] proposed Reinforcement learning based Graph Convolutional Policy Network (GCPN) which was a goal directed graph generation model. The model was highly applied in chemistry and drug discovery, where there is a need to find new molecule within given molecular properties such as drug-likeness and synthetic accessibility. In 2018 Cao and Kipf [129] proposed Generative Adversarial Network (GAN) based on likelihood free generative model. This model was also able to generate molecule with desired molecular property. In 2017, Coley et al. [122] solved the molecular graph representation problem by applying graph convolutional network on undirected molecular graph. Along with the molecular graph structural attribute, they also considered other factors such as atom and bond attribute, atom neighbor, radii and so on. In 2018, Xie et al. [131] proposed Crystal Graph Convolutional Neural Network framework, which was able to learn material properties from the crystal atomic link structure, which could be very helpful in new material design. In 2017, Ktena et al. [123] used graph convolutional neural networks for graph similarity prediction to identity brain disorders. It was very common to treat complex disease by giving many drugs at a time that targeted complex diseased proteins. However, sometimes in the presence of another drug, the effect of changing one drug is usually not observed under clinical trains. To solve this problem, in 2018, Zitnik et al. [130] proposed Decagon, a graph convolutional network-based framework. Decagon could predict what side effects on the patient could be caused by two drugs. In 2017 and 2018 Parisot et al. [124,125] used graph convolutional network for brain disease prediction. Also, in 2018, Assouel et al. [190] proposed a conditional graph generative model.

7.6. Intelligent transportation system

Intelligent Transportation Systems (ITS) are at the heart of smart cities, which are the research focus of 21st century [151, 191]. Transportation systems are back bone of any nation throughout the ages. It is found that 40% of the world's population spent at-least 1 hour on the road everyday see paper by Zhang et al. [191] in 2011. As population of the world is growing so do the vehicles which is becoming hard to manage without the help of machines. In 2019 alone US citizens used 181,541 public transportation vehicles, took 9.9 Billion trips which resulted in 55.8+ billion *KM* traveling. It suggest smart transportation is a very demand of the all big cities around the world. The transportation data can vary from letters and digits to sound images and videos. For example an automatic passenger counter which leads to revenue generation prediction needs image recognition and video surveillance. Along with automatic passenger counter we also need to in analyze which route people took most and at what time. It needs GPS and road map information. It also sometimes require non-human generated data such as 'weather'. These heterogeneous data comes from various sensors that would be installed at various locations e.g at traffic signals, in cars etc. The main problems that ITS focuses are: destination prediction, traffic signal control, demand prediction, traffic flow prediction, transportation mode and combinatorial optimization. The deep learning has been applied in the following ITS problems see work by Veras et al. [192] in 2019.

1. **Destination prediction:**
   Destination prediction is one of the task in which we predict where the person or vehicle will end up its journey. Currently it is a hot research area. This popularity is because deep learning models improve its performance with the abundance of data and transportation systems produce Tera bytes of data every day see work by Moreira et al. [132] in 2013. There are two approaches found in the literature for destination prediction.
   - **Predicting Destination From a Trajectory Prefix:** Brébisson and J. Lv et al. [133,150] proposed deep neural network method based on basis of trajectory path information in 2015 and 2018 respectively. To achieve this they have combined the idea of fixed length trajectory, possible destination information.
   - **Predicting Destination via Next Steps:** In 2017, Endo et al. [139] solved the destination prediction by iteratively predicting as the next point in the trajectory.
2. **Demand Prediction:** As oppose to destination which aims to predict where the journey will end, in demand the aim is to predict when and where the journey will start. It is used to allocate resources for example making more available taxi at any tourist spot at closing time. In 2017, Ke et al. [140] proposed convolutional-LSTM based short term taxi demand prediction model. In 2018, Yao et al. [141] only considered local information along with spatio-temporal information for demand prediction.
3. **Traffic Flow Prediction:** Traffic flow prediction is one of the important task that predict how much time it will take to flow the traffic in advance. There are many interesting work has been done in this area using deep learning. In 2016, Zhang et al. [137] made inflow and outflow predictive model by considering city as 2D grid. In 2017, Ma et al. [143] considered the speed of vehicle sensed via GPS along highways. Ma solves this problem by considering highways into single column vector and by stacking them.
4. **Travel Time Estimation:** Travel Time Estimation(TTE) is also an interesting problem in which we predict the estimated time ahead of starting the journey. Researchers such as in 2016, Siripanpornchana et al. [193] and in 2018, Zhang et al. [194] calculate the time by already predicting the path from start to end. They achieve it by stacking the known and already estimated time of trajectory sections along the path. Some works like in 2018, by Jindal et al. [147] and in 2018, Li et al. [148]consider routes between origin and destination is unknown. They achieve it by considering every possible path between two nodes.

5. **Predicting Traffic Accident Severity:** No doubt road accidents are one of the leading cause of deaths and injuries around the world. In Canada it was found that 160,000 people got injured during road accidents in 2016. In 2016, Chen et al. [138] predicted where and how bad a traffic accident could be. To achieve this they have categorize the injuries into four classes.
6. **Predicting the Mode of Transportation:** In this task the aim is to predict how people are moving rather than where they are moving. In 2017, Jiang et al. proposed [144] TrajectoryNet to solved this prediction problem by considering the GPS information. As GPS is easily available such as smart phones. This model is based on bi-direction RNN.
7. **Trajectory Clustering:** Trajectory clustering is also a prediction task. In this task we are interested to cluster similar routes probably by minimizing the Euclidian distance in an unsupervised fashion. In 2017, Yao et al. [145] and in 2002 Longest Common Subsequence (LCSS) [195] by Vlachose et al. are some works in this direction.
8. **Navigation:** In intelligent transportation system navigation is one of the challenging task. It is affected by not only environments but also personal choices, road traffic dynamics and so on. These factors makes it hard to predict accurately. Tamar et al. In 2016, Tamar et al. [196] proposed CNN-based planning module known as value iteration network(VIN). In 2016, Yang et al. utilized "time traverse map" [146]. Following works of Yang et al. in 2019, Li et al. [152] proposed network based prediction of traffic.
9. **Demand Serving:** Demand prediction is a task when and where the passengers need vehicle to travel. While demand serving is a task how to efficiently serve those demands e.g by routing vehicles. Car pooling is one of the approach for demand serving mechanism see work by Jindal et al. [147] in 2018.
10. **Traffic Signal Control:** As demand serving comes after demand prediction. Anomalously traffic signal control comes after traffic flow prediction. Intelligent traffic signaling can have huge impact on traffic jams. Reinforcement learning is found to be very effective in this field see work by Yau et al. [197] in 2017.
11. **Combinatorial Optimization:** Combinatorial optimization problems are well researched area into (intelligent) transport systems. These problems are NP-hard and being researched since long time such as famous travel sales man problem. In some cases the environment where the vehicle works may just be somewhat referred to, for example, in the Vehicle Routing Issue with Stochastic Demands (VRPSD), or Vehicle Routing Problem with Stochastic Customers (VRPSC). Further, in a real setting, ideal courses may likewise exist as a more intricate capacity at that point essentially separation or time. These augmentations incorporate factors, for example, the kind of products, the sort of street, the sort of vehicle, or even the quantity of drivers on the street. These impact the sort and intricacy of the calculation used to play out the steering. There are many approaches followed till now. For example in 2015, Vinyals et al. [134] followed pointer network based approaches. Further in 2016, Bello et al. [136] introduced the idea of training pointer networks using Reinforcement Learning. There are likewise a few specific methodologies that influence diagram structures see for example work by Li et al. [135] in 2015 to take care of comparative issues. Ongoing work by Khalil et al. in 2017 [142], and Kool et al. [149] in 2018, they have applied these graph embedding designs to a few activities research issues, including the TSP and VRP among others.

8. How to use deep learning

There are many libraries available which are also open source. Tensorflow is one the best python-based tool available. Here we will give some guidance how to use Tensorflow. We need to follow following few steps to use Tensorflow for our problems. After installing Tensorflow in our Python environment we need to follow steps below.

- Import the Tensorflow library to use.

```
import tensorflow as tf
```

- Load and prepare the MNIST dataset. Convert the samples from integers to floating-point numbers. Tensorflow has already many datasets for learning purpose. We will use MNIST handwritten digit classification dataset. We divide the datasets into two parts: training and testing. Further we are dividing this 255 to normalize it. As pixel intensity value varies from [0,255]. By dividing it by 255 all the values will lie between [0, 1].

```
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.
load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

- Build the tf.keras.Sequential model by stacking layers. Here we are building neural network. Things to be careful while creating neural network is that we need to be choosing input and output dimensions according to our dataset and problems. Like here we are giving input shape $(28, 28)$ as our single image is of size $28 * 28$ pixel values. And dimension of final layer should be equal to number of classes(in our case it is 10). For regression problem the final layer should be 1. And for binary classification it can be either 1 or 2. Intermediate or hidden layer can be as many as we want and no restriction on number of nodes as long as it best fit our problem. We are using Relu activation function for intermediate layer. Further we want to switch off the nodes whose activation value are below 0.2 so we are using dropout with threshold value 0.2.

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10)
])
```

- For each example the model returns a vector of “logits” or “log-odds” scores, one for each class.

```
predictions = model(x_train[:1]).numpy()
```

- The tf.nn.softmax function converts these logits to “probabilities” for each class. Here we are using softmax activation for final layer activation as our problem is multi-class classification problem.

```
tf.nn.softmax(predictions).numpy()
```

- The losses.SparseCategoricalCrossentropy loss takes a vector of logits and a True index and returns a scalar loss for each example. Loss function changes according to our problem e.g. for regression it could be Route Mean Square. There are many loss functions available for one problem in Tensorflow. One should use what gives the best accuracy.

```
loss_fn = tf.keras.losses.
SparseCategoricalCrossentropy(from_logits=True)
```

- This loss is equal to the negative log probability of the true class: It is zero if the model is sure of the correct class. This untrained model gives probabilities close to random (1/10 for each class), so the initial loss should be close to -tf.log(1/10) = 2.3.

```
loss_fn(y_train[:1], predictions).numpy()
```

- Now we need to compile the model to see if it is correctly build. Here we are also mentioning about optimizer algorithm which is 'adam'. There are many optimizer available such as Gradient Descent, Momentum base gradient descent and so on.

```
model.compile(optimizer='adam',
              loss=loss_fn,
              metrics=['accuracy'])
```

- The Model.fit method adjusts the model parameters to minimize the loss:

```
model.fit(x_train, y_train, epochs=5)
```

- The Model.evaluate method checks the models performance, usually on a "Validation-set" or "Test-set".

```
model.evaluate(x_test,  y_test, verbose=2)
```

- If you want your model to return a probability, you can wrap the trained model, and attach the Softmax to it:

```
probability_model = tf.keras.Sequential([
  model,
  tf.keras.layers.Softmax()
])
```

9. The challenges of deep learning

9.1. Lack of innovation in model structure

Since the re-recognition of deep learning in 2006, the deep learning model was mainly introduced as the above several classical models. The last introductions of the deep learning models were in these traditional models based on an evolution of more than a decade. In the past, most models stacked on simple models, and due to this stacking, it is becoming more difficult to increase the efficiency of data processing. However, the depth of the advantages of learning technology is still not fully implemented, as there is a need to realize that the development of a new depth of learning model, either the current depth of the learning model or the other appropriate methods for effective integration, is the need to solve the problem.

9.2. Update training methods

The supervised and unsupervised learning are the two training methods for the current deep learning models. The use of supervised training methods, the restricted Boltzmann machine, and the automatic encoder as the core model, the main pre-training, such as the use of a large number of training methods. The way is unsupervised learning. At the same time, they are combined with supervised learning to fine-tune training to learn. There is no real sense to do complete unsupervised training. So, how to achieve complete unsupervised training is the direction of the future study of the deep learning model.

9.3. Challenges with parameter learning

There are many challenges with parameter learning in deep neural networks as listed below:

1. **Learning rate:** A small learning rate takes long time to find optimum point and it can stick in local minima. While on the other hand, large learning rate may skip the optimum points and may never converge.
2. **Local optima:** A local minimum is a major problem in many parameters learning objective. The gradient descent algorithm works on taking slop of the current point and accordingly it updates the parameters. For the ideal convex problem, there is only one minima or maxima point so the absolute minima can be found. While in the case of local minima, there are many minima and maxima points. However, when the parameter updating using gradient descent reaches to the local minimum points, its gradient value becomes zero as the slope at any local minimum point will be zero, therefore it never updates the parameter again, and this is what is called the local minima problem. In Fig. 12 presents the local and global minimum problems.
3. **Saddle points:** A saddle point is a minimax point on the graph where the derivative function is almost zero. In consequence, the gradient descent stops updating the parameters. Also, the saddle point is neither minima nor maxima. This problem is generally happened when there are many dimensions present. The Hessian matrix is used to determine saddle points. The hessian matrix is a square matrix of the second order partial derivatives. It describes the local curvature of the graph under many dimensions. On a given point, if the Hessian is indefinite then that point is a saddle point. However, due to complexity of the Hessian matrix, it is not suitable for neural networks.
4. **Vanishing and exploding gradients:** This is one of the crucial problems faced while training the large neural networks. As deep neural networks contain more than two hidden layers, so the features are propagated to the final layer by applying many affine transformations followed by activation functions. In consequence, sometimes, the value of the gradients may become very large, while some times it becomes very small. The former is known as "exploding gradients" while the latter is known as "vanishing gradients" in the literature.

9.4. Reduce training time

At present, the detection of various types of deep learning models is mostly carried out in the ideal environment. In the complex reality environment, the current technology is still unable to achieve the desired results. Also, the deep learning model is composed of either simple model or several models. As the complexity of the problem is higher, the amount of information processed is more significant, which means that there is a need for more and more training time of the deep learning model. How to change the deep learning model without any flexibility in the hardware to improve the accuracy and the speed of data processing is the future research of deep learning technology.

9.5. Online learning

Unsupervised pre-training and supervised fine-tuning are the main training methods for today's deep learning techniques. However, the online learning training requires global fine tuning, which will cause the output to be fallen into the local minimum. Therefore, the current training is not conducive to the realization of the online learning. The improvements of online learning ability based on an innovative deep learning model need to be faced.

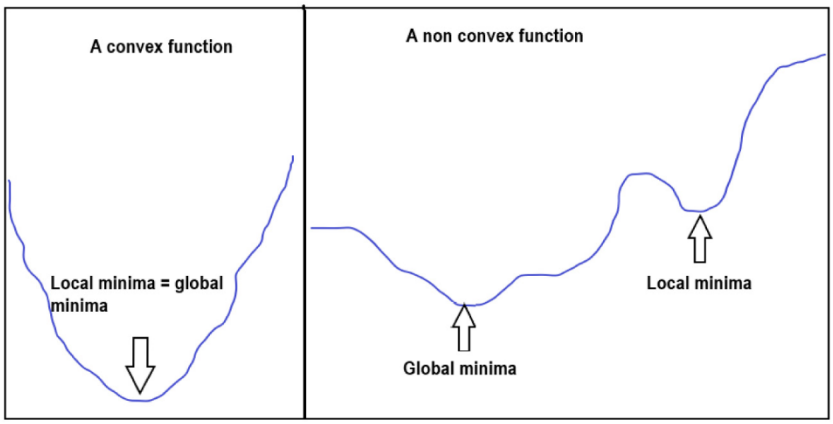

Fig. 12. Pictorial description for local and global minima.

9.6. Overcome adversarial sample

If the input sample is deliberately added, the subtle interference in the data set can cause the model to throw the wrong output with high confidence. However, the adversarial sample is a big problem in the current deep learning. This adding of the input sample cannot only effectively avoid the potential security problems when exploring how to overcome the problem of the adversarial sample, but also it can help in improving the deep learning model to solve the problem of precision. In a sense, there is a fundamental contradiction between creating a linear model for easy training and creating a non-linear model that can resist the sample. However, from the long-term development of deep learning, the creation of more powerful optimization methods and more non-linear models of training are the direction of the future in this research field.

10. Conclusion

The deep learning technology is widely applied in many fields and research areas such as speech recognition, image processing, graphs, medicine, computer vision, and so on. It is one of the fastest developing and adaptive technologies ever. The difficulties are in the presence of big and complex data on how to effectively solve the problem using deep learning. In the actual process of an application, it is more and more challenging to build an appropriate model of deep learning. Although the current deep learning is not fully matured and there are many problems which need to be solved, but the deep learning has shown strong learning ability. It is still a hot research area in the field of future artificial intelligence. This paper has discussed some classic advances of deep learning and its applications in a plethora of fields. Finally, the applications of deep learning are further presented. As there are many scientific problems which are being solved day by day, so sometimes unexpected and better performances can be achieved by deep learning in many areas such as image processing and diabetic retinopathy diagnosis, which is very difficult to diagnosed by human experts. In fact, the diabetic retinopathy diagnosis is nothing but an application of image processing. Therefore, one advance in one field might be a breakthrough solution in another field. Deep learning is getting attention very fast, every day some new application or inventions are happening. According to our limited knowledge followings are the few active research areas that will also keep getting attention in near future. (1) Generative models using deep neural networks [198] for example Generative adversarial networks, (2) Deep learning for non-Euclidean data such as Deep learning for graphs, Geometric deep learning [199], Hyperbolic neural networks [200], (3) Deep Learning for spatio-temporal data mining [201], and (4) How to improve the structures and algorithms of a deep neural network model [202] etc..

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments to improve the technical quality of the paper. This paper is supported by Science and Technology Plan Projects of Henan Province (Grant No. 192102210125, 202102210379), Zhoukou Normal University super scientific project grant *ZKNUC*2018019 and Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (Grant No. SKLNST-2020-2-01).

References

[1] D.A. Freedman, Statistical Models: Theory and Practice, Cambridge University Press, 2009.
[2] C. Mood, Logistic regression: Why we cannot do what we think we can do, and what we can do about it, Eur. Sociol. Rev. 26 (1) (2010) 67–82.
[3] D.G. Kleinbaum, M. Klein, Analysis of matched data using logistic regression, in: Logistic Regression: A Self-Learning Text, Springer, 2002, pp. 227–265.
[4] D.W. Hosmer Jr, S. Lemeshow, R.X. Sturdivant, Applied Logistic Regression, Vol. 398, John Wiley & Sons, 2013.
[5] R. Soentpiet, Advances in Kernel Methods: Support Vector Learning, MIT press, 1999.
[6] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intell. Syst. Appl. 13 (4) (1998) 18–28.
[7] I. Steinwart, A. Christmann, Support Vector Machines, Springer Science & Business Media, 2008.
[8] N.N. Schraudolph, Fast curvature matrix-vector products for second-order gradient descent, Neural Comput. 14 (7) (2002) 1723–1738.
[9] S.Z. Li, Encyclopedia of Biometrics: I-Z, Vol. 2, Springer Science & Business Media, 2009.
[10] J.J. Verbeek, N. Vlassis, B. Kröse, Efficient greedy learning of Gaussian mixture models, Neural Comput. 15 (2) (2003) 469–485.
[11] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[12] D.O. Hebb, The organization of behavior; a neuropsycholocigal theory, A Wiley Book in Clinical Psychology 62 (1949) 78.
[13] D. Crevier, AI: The Tumultuous History of the Search for Artificial Intelligence, Basic Books, Inc., 1993.
[14] J. McCarthy, M.L. Minsky, N. Rochester, C.E. Shannon, A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955, AI Mag. 27 (4) (2006) 12.

[15] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[16] F. Wang, H. Liu, J. Cheng, Visualizing deep neural network by alternately image blurring and deblurring, Neural Netw. 97 (2018) 162–172.
[17] M. Nouiehed, M. Razaviyayn, Learning deep models: Critical points and local openness, 2018, arXiv preprint arXiv:1803.02968.
[18] I. Diakonikolas, D.M. Kane, A. Stewart, Robust learning of fixed-structure bayesian networks, 2016, CoRR, abs/1606.07384.
[19] C. Yun, S. Sra, A. Jadbabaie, A critical view of global optimality in deep learning, 2018, arXiv preprint arXiv:1802.03487.
[20] B.D. Haeffele, R. Vidal, Global optimality in tensor factorization, deep learning, and beyond, 2015, arXiv:abs/1506.07540.
[21] B.D. Haeffele, R. Vidal, Global optimality in neural network training, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7331–7339.
[22] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[23] P. Mianjy, R. Arora, R. Vidal, On the implicit bias of dropout, in: ICML, 2018.
[24] H. Salehinejad, S. Valaee, Ising-dropout: A regularization method for training and compression of deep neural networks, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 3602–3606.
[25] B. Sengupta, K.J. Friston, How robust are deep neural networks? 2018, arXiv:abs/1804.11313.
[26] S. Zheng, Y. Song, T. Leung, I.J. Goodfellow, Improving the robustness of deep neural networks via stability training, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4480–4488.
[27] R. Giryes, G. Sapiro, A. Bronstein, On the stability of deep networks, 2015, CoRR abs/1412.5896.
[28] E. Haber, L. Ruthotto, Stable architectures for deep neural networks, Inverse Problems 34 (1) (2017) 014004.
[29] S. Malladi, I. Sharapov, FastNorm: Improving Numerical Stability of Deep Network Training with Efficient Normalization, 2018, https://openreview.net/pdf?id=BkbOsNeSM.
[30] B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert, E. Holtham, Reversible architectures for arbitrarily deep residual neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 32, (1) 2018.
[31] Y. Bengio, Learning Deep Architectures for AI, Now Publishers Inc, 2009.
[32] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, L. Bottou, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (12) (2010).
[33] U. Fiore, F. Palmieri, A. Castiglione, A. De Santis, Network anomaly detection with the restricted Boltzmann machine, Neurocomputing 122 (2013) 13–23.
[34] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, A learning algorithm for boltzmann machines, Cognitive Science 9 (1) (1985) 147–169.
[35] M. Ranzato, J. Susskind, V. Mnih, G. Hinton, On deep generative models with applications to recognition, in: CVPR 2011, IEEE, 2011, pp. 2857–2864.
[36] S. Rifai, Y. Bengio, A. Courville, P. Vincent, M. Mirza, Disentangling factors of variation for facial expression recognition, in: European Conference on Computer Vision, Springer, 2012, pp. 808–822.
[37] R. Salakhutdinov, G. Hinton, Deep boltzmann machines, in: Artificial Intelligence and Statistics, 2009, pp. 448–455.
[38] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.
[39] D.H. Hubel, T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, J. Physiol. 160 (1) (1962) 106.
[40] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: NIPS, 2012.
[41] Z. Zhang, P. Cui, W. Zhu, Deep learning on graphs: A survey, IEEE Trans. Knowl. Data Eng. (2020).
[42] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, IEEE Signal Process. Mag. 30 (3) (2013) 83–98.
[43] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, 2018, arXiv preprint arXiv:1812.08434.
[44] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
[45] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.
[46] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, 2016, pp. 3844–3852.
[47] B. Mohebali, A. Tahmassebi, A. Meyer-Baese, A.H. Gandomi, Probabilistic neural networks: a brief overview of theory, implementation, and application, in: Handbook of Probabilistic Models, Elsevier, 2020, pp. 347–367.
[48] J. Gast, S. Roth, Lightweight probabilistic deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3369–3378.
[49] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, 2016, arXiv preprint arXiv:1611.03530.
[50] L. Fan, Revisit Fuzzy Neural Network: Bridging the Gap Between Fuzzy Logic and Deep Learning, Tech. Rep., 2017.
[51] Y. Deng, Z. Ren, Y. Kong, F. Bao, Q. Dai, A hierarchical fused fuzzy deep neural network for data classification, IEEE Trans. Fuzzy Syst. 25 (4) (2016) 1006–1012.
[52] S. Zhou, Q. Chen, X. Wang, Fuzzy deep belief networks for semi-supervised sentiment classification, Neurocomputing 131 (2014) 312–322.
[53] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[54] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Advances in Neural Information Processing Systems, 2016, pp. 2234–2242.
[55] H. Schwenk, Continuous space translation models for phrase-based statistical machine translation, in: Proceedings of COLING 2012: Posters, 2012, pp. 1071–1080.
[56] L. Dong, F. Wei, M. Zhou, K. Xu, Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis, in: Proceedings of the National Conference on Artificial Intelligence, vol. 2, 2014, pp. 1537–1543.
[57] D. Tang, F. Wei, B. Qin, T. Liu, M. Zhou, Coooolll: A deep learning system for twitter sentiment classification, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 208–212.
[58] Y. You, Y. Qian, T. He, K. Yu, An investigation on DNN-derived bottleneck features for GMM-HMM based robust speech recognition, in: 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), IEEE, 2015, pp. 30–34.
[59] A.L. Maas, P. Qi, Z. Xie, A.Y. Hannun, C.T. Lengerich, D. Jurafsky, A.Y. Ng, Building DNN acoustic models for large vocabulary speech recognition, Comput. Speech Lang. 41 (2017) 195–213.
[60] Q. Li, W. Cai, X. Wang, Y. Zhou, D.D. Feng, M. Chen, Medical image classification with convolutional neural network, in: 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), IEEE, 2014, pp. 844–848.
[61] F. Li, L. Tran, K.-H. Thung, S. Ji, D. Shen, J. Li, A robust deep model for improved classification of AD/MCI patients, IEEE J. Biomed. Health Inf. 19 (5) (2015) 1610–1616.
[62] K. Sirinukunwattana, S.E.A. Raza, Y.-W. Tsang, D.R. Snead, I.A. Cree, N.M. Rajpoot, Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images, IEEE Trans. Med. Imaging 35 (5) (2016) 1196–1206.
[63] Q. Dou, H. Chen, L. Yu, L. Zhao, J. Qin, D. Wang, V.C. Mok, L. Shi, P.-A. Heng, Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks, IEEE Trans. Med. Imaging 35 (5) (2016) 1182–1195.
[64] A. Mallik, S. Chaudhury, Acquisition of multimedia ontology: an application in preservation of cultural heritage, Int. J. Multimedia Inf. Retr. 1 (4) (2012) 249–262.
[65] N. Höft, H. Schulz, S. Behnke, Fast semantic segmentation of RGB-D scenes with GPU-accelerated deep neural networks, in: Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz), Springer, 2014, pp. 80–85.
[66] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10, 000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891–1898.
[67] P. Pinheiro, R. Collobert, Recurrent convolutional neural networks for scene labeling, in: International Conference on Machine Learning, 2014, pp. 82–90.
[68] Y. Taigman, M. Yang, M. Ranzato, 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, 2014, pp. 1701–1708.
[69] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[70] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[71] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, G. Cottrell, Understanding convolution for semantic segmentation, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 1451–1460.
[72] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P.H. Torr, Conditional random fields as recurrent neural networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1529–1537.
[73] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.
[74] V. Badrinarayanan, A. Handa, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling, 2015, arXiv preprint arXiv:1505.07293.
[75] Z. Liu, X. Li, P. Luo, C.-C. Loy, X. Tang, Semantic image segmentation via deep parsing network, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1377–1385.
[76] W. Byeon, T.M. Breuel, F. Raue, M. Liwicki, Scene labeling with lstm recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3547–3555.
[77] G. Lin, C. Shen, A. Van Den Hengel, I. Reid, Efficient piecewise training of deep structured models for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3194–3203.
[78] F. Shen, G. Zeng, Fast semantic image segmentation with high order context and guided filtering, 2016, arXiv preprint arXiv:1605.04068.
[79] S. Chandra, I. Kokkinos, Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs, in: European Conference on Computer Vision, Springer, 2016, pp. 402–418.
[80] P. Luc, C. Couprie, S. Chintala, J. Verbeek, Semantic segmentation using adversarial networks, 2016, arXiv preprint arXiv:1611.08408.
[81] J. Hoffman, D. Wang, F. Yu, T. Darrell, Fcns in the wild: Pixel-level adversarial and constraint-based adaptation, 2016, arXiv preprint arXiv:1612.02649.
[82] B. Shuai, Z. Zuo, B. Wang, G. Wang, Dag-recurrent neural networks for scene labeling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3620–3629.
[83] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[84] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, 2017, arXiv preprint arXiv:1706.05587.
[85] M. Koziński, L. Simon, F. Jurie, An adversarial regularisation for semi-supervised training of structured output neural networks, 2017, arXiv preprint arXiv:1702.02382.
[86] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
[87] N. Souly, S. Concetto, S. Mubarak, Semi and weakly supervised semantic segmentation using generative adversarial network, 2017, arXiv preprint arXiv:1703.09695.
[88] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, 2015, arXiv preprint arXiv:1511.07122.
[89] M.T. Teichmann, R. Cipolla, Convolutional CRFs for semantic segmentation, 2018, arXiv preprint arXiv:1805.04777.
[90] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015, http://arxiv.org/abs/1409.1556.
[91] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, in: Y. Bengio, Y. LeCun (Eds.), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014, http://arxiv.org/abs/1312.6229.
[92] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[93] K. Chatfield, R. Arandjelović, O. Parkhi, A. Zisserman, On-the-fly learning for visual search of large-scale image and video datasets, Int. J. Multimedia Inf. Retr. 4 (2) (2015) 75–93.
[94] Y. Pi, N.D. Nath, A.H. Behzadan, Convolutional neural networks for object detection in aerial imagery for disaster response and recovery, Adv. Eng. Inform. 43 (2020) 101009.
[95] B. Gu, R. Ge, Y. Chen, L. Luo, G. Coatrieux, Automatic and robust object detection in x-ray baggage inspection using deep convolutional neural networks, IEEE Transactions on Industrial Electronics (2020) 1-1, http://dx.doi.org/10.1109/TIE.2020.3026285.
[96] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, L. Van Gool, One-shot video object segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 221–230.
[97] J. Shin Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, I. So Kweon, Pixel-level matching for video object segmentation using convolutional neural networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2167–2176.
[98] W.-D. Jang, C.-S. Kim, Online video object segmentation via convolutional trident network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5849–5858.
[99] Y.-T. Hu, J.-B. Huang, A. Schwing, Maskrnn: Instance level video object segmentation, in: Advances in Neural Information Processing Systems, 2017, pp. 325–334.
[100] J. Sasikumar, Investigating the Application of Deep Convolutional Neural Networks in Semi-supervised Video Object Segmentation, Technological University Dublin, 2018.
[101] D. Li, M. Jiang, Y. Fang, Y. Huang, C. Zhao, Deep video foreground target extraction with complex scenes, in: 2018 International Conference on Sensor Networks and Signal Processing (SNSP), IEEE, 2018, pp. 440–445.
[102] H. Xiao, J. Feng, G. Lin, Y. Liu, M. Zhang, Monet: Deep motion exploitation for video object segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1140–1148.
[103] V. Goel, J. Weng, P. Poupart, Unsupervised video object segmentation for deep reinforcement learning, in: Advances in Neural Information Processing Systems, 2018, pp. 5683–5694.
[104] A. Schofield, P. Mehta, T. Stonham, A system for counting people in video images using neural networks to identify the background scene, Pattern Recognit. 29 (8) (1996) 1421–1428.
[105] A. Tavakkoli, Foreground-background segmentation in video sequences using neural networks, in: Intelligent Systems: Neural Networks and Applications, 2005.
[106] D. Culibrk, O. Marques, D. Socek, H. Kalva, B. Furht, A neural network approach to bayesian background modeling for video object segmentation, in: VISAPP (1), 2006, pp. 474–479.
[107] L. Maddalena, A. Petrosino, A self-organizing approach to detection of moving patterns for real-time applications, in: International Symposium on Brain, Vision, and Artificial Intelligence, Springer, 2007, pp. 181–190.
[108] J.A. Ramírez-Quintana, M.I. Chacon-Murguía, Self-organizing retinotopic maps applied to background modeling for dynamic object segmentation in video sequences, in: The 2013 International Joint Conference on Neural Networks (IJCNN), IEEE, 2013, pp. 1–8.
[109] R. Guo, H. Qi, Partially-sparse restricted boltzmann machine for background modeling and subtraction, in: 2013 12th International Conference on Machine Learning and Applications, Vol. 1, IEEE, 2013, pp. 209–214.
[110] P. Xu, M. Ye, X. Li, Q. Liu, Y. Yang, J. Ding, Dynamic background learning through deep auto-encoder networks, in: Proceedings of the 22nd ACM International Conference on Multimedia, 2014, pp. 107–116.
[111] P. Xu, M. Ye, Q. Liu, X. Li, L. Pei, J. Ding, Motion detection via a couple of auto-encoder networks, in: 2014 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2014, pp. 1–6.
[112] J.A. Ramirez-Quintana, M.I. Chacon-Murguia, Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios, Pattern Recognit. 48 (4) (2015) 1137–1149.
[113] Z. Qu, S. Yu, M. Fu, Motion background modeling based on context-encoder, in: 2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR), IEEE, 2016, pp. 1–5.
[114] T. Minematsu, A. Shimada, H. Uchiyama, R.-i. Taniguchi, Analytics of deep neural network-based background subtraction, J. Imaging 4 (6) (2018) 78.
[115] S. Ammar, T. Bouwmans, N. Zaghden, M. Neji, Moving objects segmentation based on deepsphere in video surveillance, in: International Symposium on Visual Computing, Springer, 2019, pp. 307–319.
[116] M. Sultana, A. Mahmood, T. Bouwmans, S.K. Jung, Unsupervised adversarial learning for dynamic background modeling, in: International Workshop on Frontiers of Computer Vision, Springer, 2020, pp. 248–261.
[117] D.K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, R.P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in: Advances in Neural Information Processing Systems, 2015, pp. 2224–2232.
[118] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, P. Riley, Molecular graph convolutions: moving beyond fingerprints, J. Comput. Aided Mol. Des. 30 (8) (2016) 595–608.
[119] R.v.d. Berg, T.N. Kipf, M. Welling, Graph convolutional matrix completion, 2017, arXiv preprint arXiv:1706.02263.
[120] F. Monti, M. Bronstein, X. Bresson, Geometric matrix completion with recurrent multi-graph neural networks, in: Advances in Neural Information Processing Systems, 2017, pp. 3697–3707.

[121] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, 2017, arXiv preprint arXiv:1704.01212.
[122] C.W. Coley, R. Barzilay, W.H. Green, T.S. Jaakkola, K.F. Jensen, Convolutional embedding of attributed molecular graphs for physical property prediction, J. Chem. Inf. Model. 57 (8) (2017) 1757–1772.
[123] S.I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, D. Rueckert, Distance metric learning using graph convolutional networks: Application to functional brain networks, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017, pp. 469–477.
[124] S. Parisot, S.I. Ktena, E. Ferrante, M. Lee, R.G. Moreno, B. Glocker, D. Rueckert, Spectral graph convolutions for population-based disease prediction, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017, pp. 177–185.
[125] S. Parisot, S.I. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker, D. Rueckert, Disease prediction using graph convolutional networks: Application to autism spectrum disorder and Alzheimer's disease, Med. Image Anal. 48 (2018) 117–130.
[126] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, J. Tang, Deepinf: Social influence prediction with deep learning, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2110–2119.
[127] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 974–983.
[128] J. You, B. Liu, Z. Ying, V. Pande, J. Leskovec, Graph convolutional policy network for goal-directed molecular graph generation, in: Advances in Neural Information Processing Systems, 2018, pp. 6410–6421.
[129] N. De Cao, T. Kipf, MolGAN: An implicit generative model for small molecular graphs, 2018, arXiv preprint arXiv:1805.11973.
[130] M. Zitnik, M. Agrawal, J. Leskovec, Modeling polypharmacy side effects with graph convolutional networks, Bioinformatics 34 (13) (2018) i457–i466.
[131] T. Xie, J.C. Grossman, Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties, Phys. Rev. Lett. 120 (14) (2018) 145301.
[132] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, L. Damas, Predicting taxi–passenger demand using streaming data, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1393–1402.
[133] A. De Brébisson, É. Simon, A. Auvolat, P. Vincent, Y. Bengio, Artificial neural networks applied to taxi destination prediction, 2015, arXiv preprint arXiv:1508.00021.
[134] O. Vinyals, M. Fortunato, N. Jaitly, Pointer networks, in: Advances in Neural Information Processing Systems, 2015, pp. 2692–2700.
[135] Y. Li, D. Tarlow, M. Brockschmidt, R. Zemel, Gated graph sequence neural networks, 2015, arXiv preprint arXiv:1511.05493.
[136] I. Bello, H. Pham, Q.V. Le, M. Norouzi, S. Bengio, Neural combinatorial optimization with reinforcement learning, 2016, arXiv preprint arXiv:1611.09940.
[137] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, 2016, arXiv preprint arXiv:1610.00081.
[138] Q. Chen, X. Song, H. Yamada, R. Shibasaki, Learning deep representation from big and heterogeneous data for traffic accident inference, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[139] Y. Endo, K. Nishida, H. Toda, H. Sawada, Predicting destinations from partial trajectories using recurrent neural network, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2017, pp. 160–172.
[140] J. Ke, H. Zheng, H. Yang, X.M. Chen, Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach, Transp. Res. C 85 (2017) 591–608.
[141] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, Z. Li, Deep multi-view spatial-temporal network for taxi demand prediction, 2018, arXiv preprint arXiv:1802.08714.
[142] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, L. Song, Learning combinatorial optimization algorithms over graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 6348–6358.
[143] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (4) (2017) 818.
[144] X. Jiang, E.N. de Souza, A. Pesaranghader, B. Hu, D.L. Silver, S. Matwin, Trajectorynet: An embedded gps trajectory representation for point-based classification using recurrent neural networks, 2017, arXiv preprint arXiv:1705.02636.
[145] D. Yao, C. Zhang, Z. Zhu, J. Huang, J. Bi, Trajectory clustering via deep representation learning, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 3880–3887.
[146] S. Yang, J. Li, J. Wang, Z. Liu, F. Yang, Learning urban navigation via value iteration network, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 800–805.
[147] I. Jindal, Z.T. Qin, X. Chen, M. Nokleby, J. Ye, Optimizing taxi carpool policies via reinforcement learning and spatio-temporal mining, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 1417–1426.
[148] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, Y. Liu, Multi-task representation learning for travel time estimation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1695–1704.
[149] W. Kool, H. Van Hoof, M. Welling, Attention, learn to solve routing problems!, 2018, arXiv preprint arXiv:1803.08475.
[150] J. Lv, Q. Li, Q. Sun, X. Wang, T-CONV: A convolutional neural network for multi-scale taxi trajectory prediction, in: 2018 IEEE International Conference on Big Data and Smart Computing (Bigcomp), IEEE, 2018, pp. 82–89.
[151] Y. Yuan, Z. Xiong, Q. Wang, Acm: Adaptive cross-modal graph convolutional neural networks for rgb-d scene recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 9176–9184.
[152] J. Li, D. Fu, Q. Yuan, H. Zhang, K. Chen, S. Yang, F. Yang, A traffic prediction enabled double rewarded value iteration network for route planning, IEEE Trans. Veh. Technol. 68 (5) (2019) 4170–4181.
[153] Z. Tu, X. Chen, A.L. Yuille, S.-C. Zhu, Image parsing: Unifying segmentation, detection, and recognition, Int. J. Comput. Vis. 63 (2) (2005) 113–140.
[154] T. Pavlidis, Fundamentals of picture segmentation, in: Structural Pattern Recognition, Springer, 1977, pp. 65–89.
[155] G.B. Coleman, H.C. Andrews, Image segmentation by clustering, Proc. IEEE 67 (5) (1979) 773–785.
[156] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, IEEE Trans. Pattern Anal. Mach. Intell. (6) (1984) 721–741.
[157] Y.J. Zhang, A survey on evaluation methods for image segmentation, Pattern Recogn. 29 (8) (1996) 1335–1346.
[158] H. Narkhede, Review of image segmentation techniques, Int. J. Sci. Modern Eng. 1 (8) (2013) 54–61.
[159] D. Kaur, Y. Kaur, Various image segmentation techniques: a review, Int. J. Comput. Sci. Mobile Comput. 3 (5) (2014) 809–814.
[160] J. Kuruvilla, D. Sukumaran, A. Sankar, S.P. Joy, A review on image processing and image segmentation, in: 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), 2016, pp. 198–203,
[161] G.B. Huang, H. Lee, E. Learned-Miller, Learning hierarchical representations for face verification with convolutional deep belief networks, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2518–2525.
[162] M.A. Fischler, R.A. Elschlager, The representation and matching of pictorial structures, IEEE Trans. Comput. 100 (1) (1973) 67–92.
[163] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep learning for generic object detection: A survey, Int. J. Comput. Vis. 128 (2) (2020) 261–318.
[164] F. Sultana, A. Sufian, P. Dutta, A review of object detection models based on convolutional neural network, in: Intelligent Computing: Image Processing Based Applications, Springer, 2020, pp. 1–16.
[165] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[166] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Advances in Neural Information Processing Systems, 2016, pp. 379–387.
[167] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[168] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[169] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2011) 743–761.
[170] M. Enzweiler, D.M. Gavrila, Monocular pedestrian detection: Survey and experiments, IEEE Trans. Pattern Anal. Mach. Intell. 31 (12) (2008) 2179–2195.
[171] D. Geronimo, A.M. Lopez, A.D. Sappa, T. Graf, Survey of pedestrian detection for advanced driver assistance systems, IEEE Trans. Pattern Anal. Mach. Intell. 32 (7) (2009) 1239–1258.
[172] Z. Sun, G. Bebis, R. Miller, On-road vehicle detection: A review, IEEE Trans. Pattern Anal. Mach. Intell. 28 (5) (2006) 694–711.
[173] K.V. Sakhare, T. Tewari, V. Vyas, Review of vehicle detection systems in advanced driver assistant systems, Arch. Comput. Methods Eng. 27 (2) (2020) 591–610.
[174] H. Yuan, B. Zhang, M. Liu, Vehicle detection based on area and proportion prior with faster-RCNN, in: Sensor Networks and Signal Processing, Springer, 2020, pp. 435–446.
[175] S. Zafeiriou, C. Zhang, Z. Zhang, A survey on face detection in the wild: past, present and future, Comput. Vis. Image Underst. 138 (2015) 1–24.

[176] I. Masi, Y. Wu, T. Hassner, P. Natarajan, Deep face recognition: A survey, in: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), IEEE, 2018, pp. 471–478.
[177] D. Zeng, R. Veldhuis, L. Spreeuwers, A survey of face recognition techniques under occlusion, 2020, arXiv preprint arXiv:2006.11366.
[178] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: A review, IEEE Trans. Neural Netw. Learn. Syst. 30 (11) (2019) 3212–3232.
[179] T. Cane, J. Ferryman, Evaluating deep semantic segmentation networks for object detection in maritime surveillance, in: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2018, pp. 1–6.
[180] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[181] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[182] L. Jian-Wei, L. Hai-En, L. Xiong-Lin, Learning technique of probabilistic graphical models: a review, Acta Automat. Sinica 40 (6) (2014) 1025–1044.
[183] T. Bouwmans, Traditional and recent approaches in background modeling for foreground detection: An overview, Comput. Sci. Rev. 11 (2014) 31–66.
[184] T. Bouwmans, A. Sobral, S. Javed, S.K. Jung, E.-H. Zahzah, Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset, Comp. Sci. Rev. 23 (2017) 1–71.
[185] B. Garcia-Garcia, T. Bouwmans, A.J.R. Silva, Background subtraction in real applications: Challenges, current models and future directions, Comp. Sci. Rev. 35 (2020) 100204.
[186] T. Bouwmans, E.H. Zahzah, Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance, Comput. Vis. Image Underst. 122 (2014) 22–34.
[187] S. Javed, S.H. Oh, A. Sobral, T. Bouwmans, S.K. Jung, OR-PCA with MRF for robust foreground detection in highly dynamic backgrounds, in: Asian Conference on Computer Vision, Springer, 2014, pp. 284–299.
[188] L. Xu, Y. Li, Y. Wang, E. Chen, Temporally adaptive restricted Boltzmann machine for background modeling, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[189] T. Bouwmans, S. Javed, M. Sultana, S.K. Jung, Deep neural network concepts for background subtraction: A systematic review and comparative evaluation, Neural Netw. 117 (2019) 8–66.
[190] R. Assouel, M. Ahmed, M.H. Segler, A. Saffari, Y. Bengio, Defactor: Differentiable edge factorization-based probabilistic graph generation, 2018, arXiv preprint arXiv:1811.09766.
[191] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, C. Chen, Data-driven intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 12 (4) (2011) 1624–1639.
[192] M. Veres, M. Moussa, Deep learning for intelligent transportation systems: a survey of emerging trends, IEEE Transactions on Intelligent Transportation Systems 21 (8) (2019) 3152–3168.
[193] C. Siripanpornchana, S. Panichpapiboon, P. Chaovalit, Travel-time prediction with deep learning, in: 2016 IEEE Region 10 Conference (TENCON), IEEE, 2016, pp. 1859–1862.
[194] H. Zhang, H. Wu, W. Sun, B. Zheng, Deeptravel: a neural network based travel time estimation model with auxiliary supervision, 2018, arXiv preprint arXiv:1802.02147.
[195] M. Vlachos, G. Kollios, D. Gunopulos, Discovering similar multidimensional trajectories, in: Proceedings 18th International Conference on Data Engineering, IEEE, 2002, pp. 673–684.
[196] A. Tamar, Y. Wu, G. Thomas, S. Levine, P. Abbeel, Value iteration networks, in: Advances in Neural Information Processing Systems, 2016, pp. 2154–2162.
[197] K.-L.A. Yau, J. Qadir, H.L. Khoo, M.H. Ling, P. Komisarczuk, A survey on reinforcement learning models and algorithms for traffic signal control, ACM Comput. Surv. 50 (3) (2017) 1–38.
[198] R. Salakhutdinov, Learning deep generative models, Annu. Rev. Stat. Appl. 2 (2015) 361–385.
[199] J. Masci, E. Rodolà, D. Boscaini, M. Bronstein, H. Li, Geometric deep learning, in: SIGGRAPH ASIA 2016 Courses, 2016, pp. 1–50.
[200] O.-E. Ganea, G. Bécigneul, T. Hofmann, Hyperbolic neural networks, arXiv preprint arXiv:1805.09112.
[201] S. Wang, J. Cao, P. Yu, Deep learning for spatio-temporal data mining: a survey, IEEE Transactions on Knowledge and Data Engineering (2020) 1-1, http://dx.doi.org/10.1109/TKDE.2020.3025580.
[202] J. You, J. Leskovec, K. He, S. Xie, Graph structure of neural networks, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, 119, PMLR, 2020, pp. 10881–10891, http://proceedings.mlr.press/v119/you20b.html.