





REVIEW

Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey

Mehdi Gheisari^{1,2,3}  | Fereshteh Ebrahimzadeh⁴ | Mohamadtaghi Rahimi⁵ |
 Mahdieh Moazzamigodarzi⁶ | Yang Liu^{1,7}  | Pijush Kanti Dutta Pramanik⁸ |
 Mohammad Ali Heravi⁹ | Abolfazl Mehbodniya¹⁰  | Mustafa Ghaderzadeh¹¹  |
 Mohammad Reza Feylizadeh¹² | Saeed Kosari¹³

¹School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China

²Department of Cognitive Computing, Institute of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, India

³Department of Computer Science, Islamic Azad University, Tehran, Iran

⁴Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran

⁵Department of Mathematics and Statistics, Iran University of Science and Technology, Tehran, Iran

⁶Department of Mathematics and Statistics, University of Northern British Columbia, BC, Canada

⁷Peng Cheng Laboratory, Shenzhen, China

⁸Department of Technology, Jodhpur Institute of Engineering & Technology, Jodhpur, India

⁹Department of Computer Engineering, Sadjad University of Technology, Mashhad, Iran

¹⁰Department of Electronics and Communications Engineering, Kuwait College of Science and Technology, Doha District, Kuwait

¹¹Department of Artificial Intelligence, Smart University of Medical Sciences, Tehran, Iran

¹²Department of Industrial Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran

¹³Institute of Computing Science and Technology, Guangzhou University, Guangzhou, China

Correspondence

Yang Liu, Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China.
 Email: liu.yang@hit.edu.cn

Mohammad Reza Feylizadeh, Department of Industrial Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran.
 Email: Mo.Feylizadeh@iau.ac.ir

[Correction added on 30 June 2023, after first online publication. The below correction needs to be noted. The affiliation of Mohammad Ali Heravi is as follows: Department of Computer Engineering, Sadjad University of Technology, Mashhad, Iran]

Abstract

Deep Learning (DL) is a subfield of machine learning that significantly impacts extracting new knowledge. By using DL, the extraction of advanced data representations and knowledge can be made possible. Highly effective DL techniques help to find more hidden knowledge. Deep learning has a promising future due to its great performance and accuracy. We need to understand the fundamentals and the state-of-the-art of DL to leverage it effectively. A survey on DL ways, advantages, drawbacks, architectures, and methods to have a straightforward and clear understanding of it from different views is explained in the paper. Moreover, the existing related methods are compared with each other, and the application of DL is described in some applications, such as medical image analysis, handwriting recognition, and so on.

KEYWORDS

data mining, data privacy, deep learning

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2023 The Authors. *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

1 | INTRODUCTION

Everything around us can generate Big data. Some of these big data are so complicated that traditional methods or databases cannot process them. Figure 1 shows the definition of Big Data in the schematic.

As Figure 1 shows, large dataset acts like a technological, cultural, and scholarly event that is based on the interaction of [1]:

- **Technology:** maximisation of computing power and the accuracy of algorithms to accumulate, analyse, link, and compare large data sets [2].
- **Analyses:** identification of patterns by drawing on large datasets to make social, economic, technological, and legal assertions [3].
- **Mythology:** the extensive belief that big data sets suggest a higher form of knowledge and intelligence that can make previously impossible insights with the aura of objectivity, accuracy, and truth [1, 4].

Extraction of advanced representation (feature) was not possible by conventional Machine learning (ML)-shallow learning or statistical algorithms because of the characteristics of generated data in this system called V's characteristics. Three significant characteristics, all of which have brought new challenges, are:

- I. Volume: The amount of generated data is high.
- II. Velocity: The speed of generating data is high, and we cannot control the system easy to work well.
- III. Variety: Data come from heterogeneous sources, such as social media, videos, images etc.
- IV. The produced data from a variety of providers can be combined.

Traditional ML algorithms are unsuitable for dealing with the characteristics mentioned above. Some of the shallow learning drawbacks are:

- I. We cannot achieve high-quality equipments with superficial learning.
- II. We cannot know online communities and political movements with shallow learning.

- III. Shallow learning methods depend on features that are often tricky to understand.
- IV. We cannot quickly solve simple algorithms like XOR with Neural Networks (NN) and one neuron.
- V. Good input features are essential for successful traditional ML (Shallow learning) [2].
- VI. Neural Network faces more than three layers in general.

In general, large dataset considers important questions about the constitution of knowledge, how we should absorb information, research processes, nature, and arrangement of reality. Large Data suggests the humanistic disciplines a new way to assert the status of quantitative science and objective method. It causes many more social spaces quantifiable. But it has some significant failures, such as:

- I. Large data is not always a sign of good data.
- II. Taken 9out of content, large Data loses its meaning divides.
- III. Since it is accessible does not make it ethical to use. (Privacy is a critical issue.)
- IV. Restricted access to Big Data makes a new problem because most scholarly researchers do not access big data.

Before 2006, we could not train deep architectures to find complex patterns. Previously, if we faced with tens of different inputs, the traditional neural network had the best performance; we had to grow the number of nodes in layers exponentially, which caused two significant problems:

- **Vanishing Gradient (Underfitting):** This means that we can balance the first layers of the network more rapidly than later layers, and optimisation is more complicated during passing so that the results will be worse than usual with a high number of neuron layers. In brief, we cannot improve the weights of each edge quickly.
- **Overfitting (High Variance):** This means that the network fits more than the requirement so that the network ends up with the wrong result for new data. From another point of view, the network fits more than enough that only results well for trained data, so it lasts bad results for test data: the more neurons, the more probability of overfitting [3].

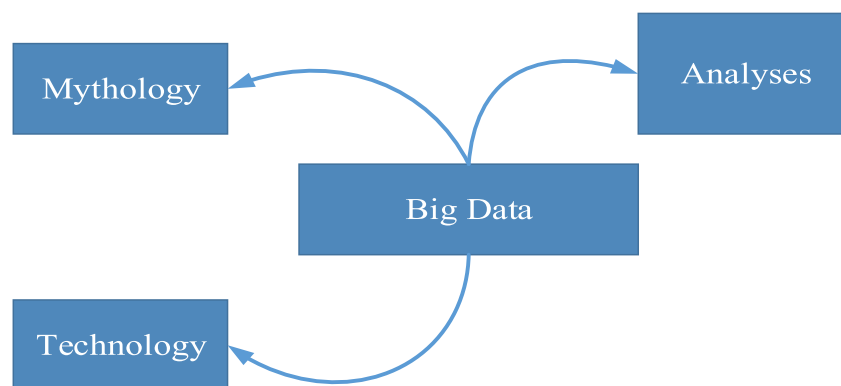


FIGURE 1 Big data definition.

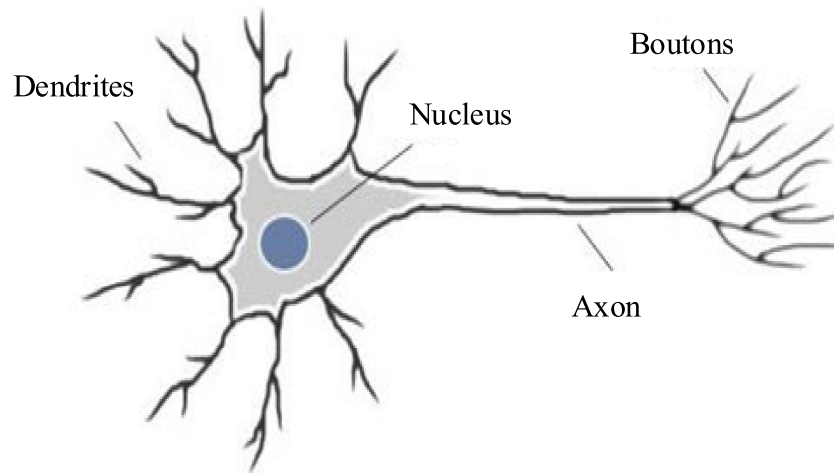


FIGURE 2 Real neural network structure in human beings.

In simple, we cannot easily generalise the network [5], based on papers by Geoffrey Hinton [6] and Bengio [7]. We have faced another way to extract advanced representations from data in an era where almost all of the things that produce data are named Greedy layer-wise unsupervised pre-training. They tried to train the network layer by layer, not considering it a black box.

Deep Learning (DL) is a powerful way of analysing big data that allows computers to learn without being taught. Everything like health care transportation, manufacturing and more are affected by DL. Companies use DL to deal with hard problems, which we can categorise them into some classes such as:

1. **Computer vision:** one of the most important applications of DL is computer vision nowadays. Image classification can be used in the image search system, categorising images automatically to be searchable from text queries or automatic tagging. Another example of the deep net is that suppose we want to predict if a given image contains a human face or not; at first, the deep net uses edges to detect different parts of faces, lips, and so on. Next, it combines the results to form the whole face.
 - a. Object Recognition: recognising objects within images.
 - b. Video Parsing: that is useful for driverless cars, robots, thief detection, and movie ratings.
2. **Speech recognition**
3. **Text Processing:**
 - a) Fact Extraction: finding facts from common sentences.
 - b) Machine Translation
 - c) Sentiment Analysis: One example is Twitter sentiment analysis (you can search any query; the result is people's sentiment on Twitter. We can reach this tool from the URL¹ up to now.
4. **Medical:**
 - a) Disease Detection: finding factors that impact a special disease such as cancer.

- b) Drug Discovery: Predicting the biological activity of different drug molecules based on chemical structure.
 - c) Radiology: Discriminating malignant tumours from benign tumours.
5. **Finance:** making, buying, and selling predictions for short-term trading and long-term investment.
 6. **Digital advertising:** segmenting customers to offer personalised or real-time advertisements.
 7. **Fraud detections**
 8. **Agriculture:** finding problematic environmental conditions.

DL algorithms stem from Artificial NN architectures that are especially useful for classification goals [8]. We can see its natural structure in Figure 2.

In other words, until now, all types of DL are based on Artificial NN. One of the ways to deal with real-life issues in the world is ML, whereas DL is a special kind of ML. Machine learning typically needs a complicated education, spanning software engineering and computer science to statistical ways and linear algebra.

If we want to compare shallow learning (traditional methods) with DL approaches visually, we can mention the following example in real: anticipating Cognitive Performance of Stroke Patients From magnetic resonance imaging (MRI) Lesion pictures (shallow learning approaches).

Figure 4 shows anticipating Cognitive accomplishment of Stroke Patients From MRI Lesion pictures (DL approaches).

As shown in Figures 3 and 4, we can understand that extracting features and selecting feature are two essential differences between the shallow and DL approaches. When we use the DL method, this method does feature extraction and feature selection itself, but in Shallow learning methods, we should do them by ourselves. We can find more advanced knowledge based on ML algorithms to which we do not know the answers upfront. If we want to understand the relationship between DL, ML, and Artificial intelligence fields, the best way in visualising them as concentric circles, as Figure 5 shows this:

And to find to what extent DL can affect the industry, we can refer to Figure 6.

¹https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/.

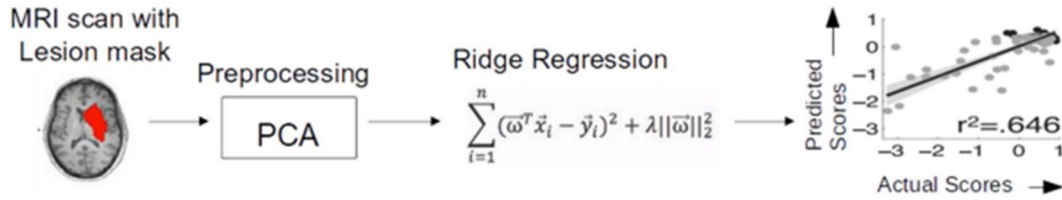


FIGURE 3 Shallow learning approach for anticipating cognitive accomplishment of stroke patients from magnetic resonance imaging lesion pictures [9] or anticipating cognitive accomplis.

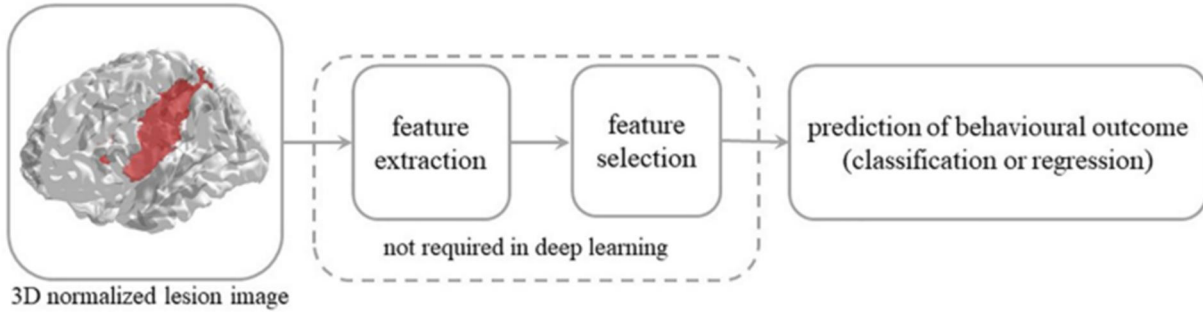


FIGURE 4 Deep learning approach for anticipating cognitive accomplishment of stroke patients from magnetic resonance imaging lesion pictures [9].

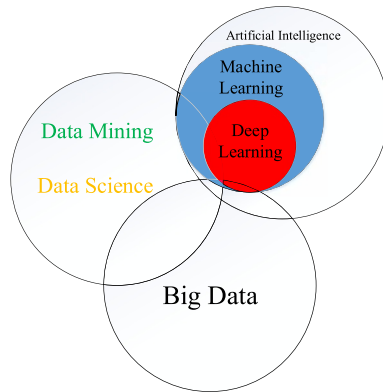


FIGURE 5 Relation of DL, ML, BD, DS, and AI.

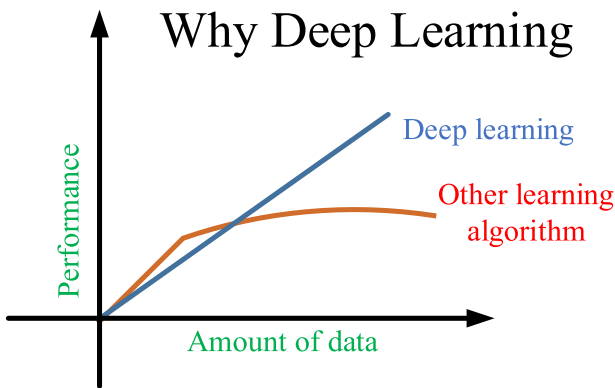


FIGURE 6 Effect of deep learning [10].

As we can see, after using DL, we achieved a remarkable breakthrough in accuracy and better knowledge [11]. DL emerges as a central tool for solving perception problems [12].

In other words, DL is vital for better knowledge extraction. So, we must have a true image of this new knowledge extractor if we want to research to find a better representation. The aim of DL is to extract features without much more hard work, especially when we encounter tens of different inputs or more. Deep nets also have strong power in data integration.

2 | DEEP LEARNING ARCHITECTURE AND APPROACHES

In general, DL ways are in three main classes:

- Generative (unsupervised) deep architectures: It aims to determine high-level correlation properties for pattern analysis or synthesis tasks.
- Discriminative (supervised) deep architectures are intended to enable pattern classification, most of the time by providing distributions of classes, as we introduce Convolutional Neural Network (CNN) in the next section.
- Hybrid deep architectures want to provide better discrimination with the help of Generative deep architectures and enhanced optimisation techniques, like Deep Neural networks (DNN).

We divide this paper into seven parts. In the first part of Section 2, we present DL architecture, applications, and approaches. In part 3, we explain six common types of DL methods. Section 4 presents some examples of the Internet of Things (IoT) field. And Section 5 compares some common DL states of the art. Then in Section 6, some applicants for DL are represented. Section 7 presents the conclusion and future works.

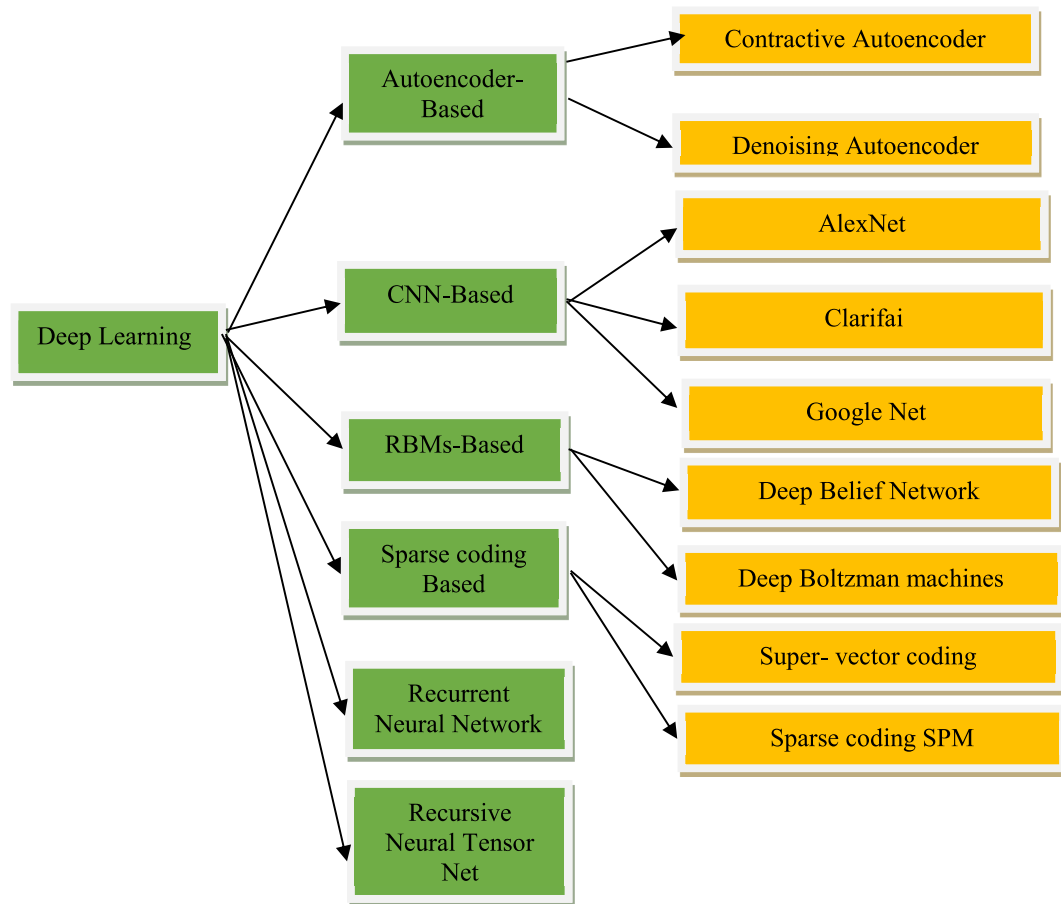


FIGURE 7 Common deep learning models. CNN, convolutional neural network; RBM, restricted Boltzmann machine

3 | COMMON TYPES OF DL MODELS

If we want to divide DL methods, it can be six major models: Recurrent Neural Networks (RNN), CNN, and Deep Belief Networks (DBN) algorithms that consist of Restricted Boltzmann Machines (RBMs: One of the most popular building blocks for deep architectures), Autoencoders, Sparse Coding and Recursive Neural Tensor Network (RNTN). These methods are an advanced version of traditional NN with enhanced training. In brief, the categorisation is not limited to this, but we show the most common ones in Figure 7:

3.1 | Convolutional neural networks (CNN)

One of computer vision's most important DL methods is CNN. It is good for applications whose input data has structure [13–15]. It is a family of techniques that adapt to all data and problems. It consists of four layers (1) convolution, (2) ReLU, (3) pooling (4) Fully connected (FC) layers. In this network, all neurons do not relate to all next-layer neurons. Each neuron in the later layers accepts only some connections from the earlier layers. These types of networks have two training steps; one is fed forward, and the other is Backpropagation. At first, data will be posted to the deep net

in a feedforward manner. Next, the result will be comparable with the correct response, called the loss function. Then, the start of the backpropagation steps depends on the amount of loss function. The computing of the gradient of each weight relies on the chain rule to minimise the error. Finally, the parameters are changed based on the calculated gradient. These steps are repeatable until reaching the wanted accuracy. In detail, it acts in the hierarchical model, we can see in the figure.

CNN includes four main layers which are that the first three layers can be repeated:

1. **Convolutional layer:** Act as a filter layer, extracting features from the input data. It has two main advantages [16]:
 - I. It reduces the impressive number of features.
 - II. It causes more stability and invariance of the network.
 - III. It discovers simple patterns that will be built up by later layers if we use more than one convolution sublayer.
2. **ReLU layer:** It uses a general function approximator that can estimate any function over input values and acts as a non-linear transformer. Without a non-linear activation function after each layer, the whole network act as a simple linear transformer, which does not have so much power for

complicated problems such as digit recognition and image-net classification [17]. By using CNN and ImageNet classification task, ImageNet is an image dataset organised in accordance with the WordNet hierarchy; if we have no activation, it results in 8% poorer performance than ReLU. If we want to see the function of this layer in schematic form, Figure 8 mentions this. This layer allows the net to be properly trained (Figure 9).

3. **Pooling layer:** It is used for reducing dimension of feature maps without losing information, preventing overfitting problems. It causes the net to only concentrate on the relevant pattern, not each founded pattern. This layer helps to limit a little bit of processing and memory requirements running by CNN.
4. **FC:** It equips the net to classify data samples. But it has one major drawback since it uses supervised learning; it needs a great number of labelled data.

Figure 10 describes a Comparison of CNN-based architectures.

Early layers detect simple patterns in CNN, and later layers construct the whole view. In brief, it has five steps to accomplish.

Step 1: Initialisation of all filters and parameters/weights with accidental values

Step 2: giving input into the neural network, undergoes the forward propagation step (convolution, ReLU, and pooling operations in addition to forwarding propagation in the FC layer) and discovers the output probabilities for each class.

Step 3: Calculation of the overall error in the output-layer (summation of all four classes)

Step 4: Utilise Backpropagation to compute the gradients of the error.

Step 5: Repeat steps 2–4 with all inputs in the training part of dataset.

3.2 | Restricted Boltzmann machines (RBMS)

It was suggested by Hinton et al. in 1986 [19]. They are a type of Generative deep architecture that can extract features to reconstruct inputs. Each node should be connected to the different side layers to build the bipartite graph and act as a learning module [20]. From another point of view, this method

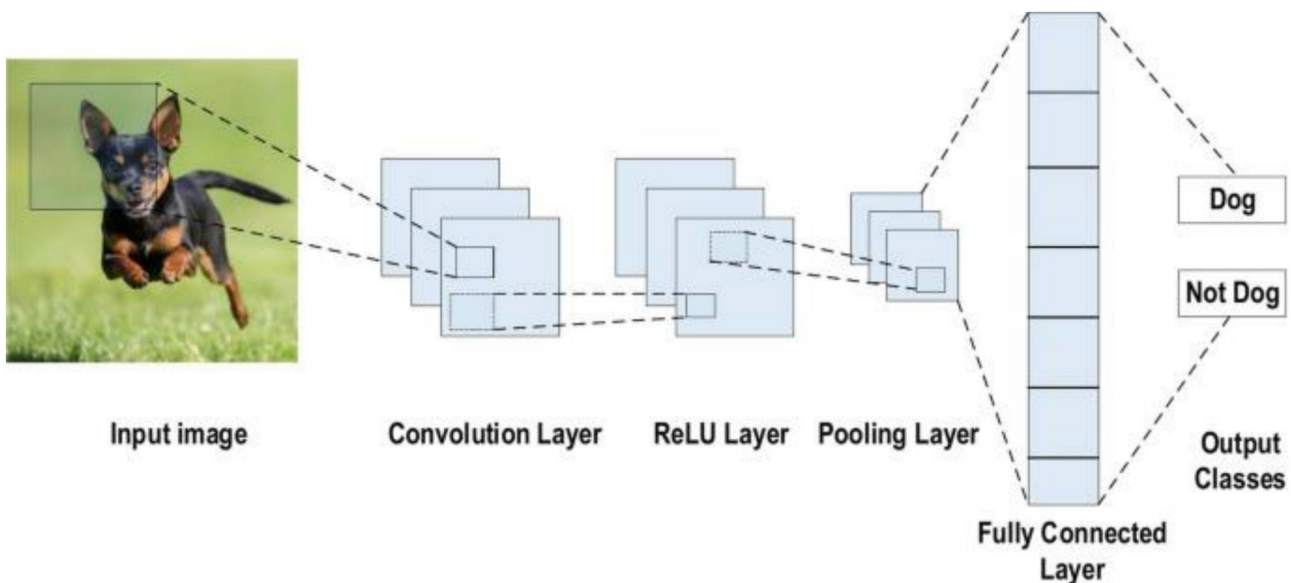


FIGURE 8 Convolutional neural network [18].

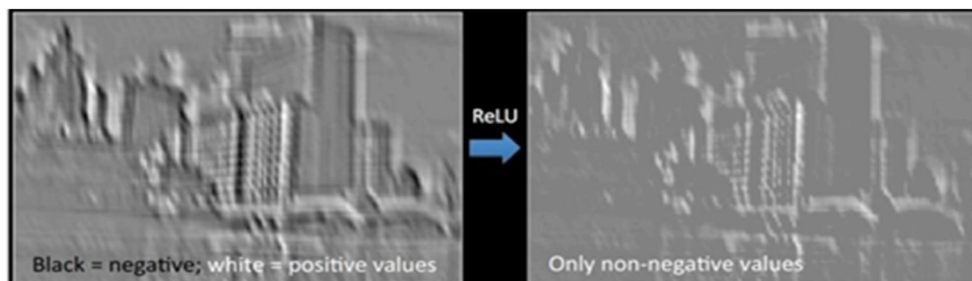


FIGURE 9 ReLU function.

Architectures based on CNN	Configuration	Advantages	Disadvantages
AlexNet	5 Convolutional-layer+ 3 fully connected layers	One of the best for computer vision understanding	<ol style="list-style-type: none"> 1. Why it works well 2. It needs fixed image sizes
Clarifai	5 Convolutional-layer+ 3 fully connected layers	<ol style="list-style-type: none"> 1. High accuracy 2. We can see the background 	It is well-high difficult to work with
GoogleNet	21 Convolutional-layer + 1 fully connected layers	Expanded the depth of the network without computational need changes	More prone to overfitting

FIGURE 10 Comparison of convolutional neural network (CNN)-based architectures.

is restricted by having only two layers (one visible and the other hidden layer). They are also limited because none of the nodes in the same layer can share a connection or have a relationship. They can find a pattern from data by reconstructing input automatically. Figure 11 shows the Restricted Boltzmann machines (RBM) structure.

RBM takes input and encodes them into some numbers in a forwarding pass. And in the backward pass, it reconstructs input based on those encoded inputs (numbers). They not only decipher interrelationships among input features but also decide which feature is the most important to extract them. The two most common DNN with Deep Generative methods are DBNs and Deep Boltzmann Machines (DBM) algorithms. The schematic difference between these two NN appears in Figure 12 [21].

From a structural aspect, DBN is the same as multi layer perceptron but differs in training level; we can view them as stacks of RBMs. We see a directed connection in the first two top layers of DBN that stem from an RBM. Still, in the rest of the layers, there are undirected connections in reverse DBM. Besides, all the layers relate to each other in an undirected manner [22]. Each RBM layer in DBN learns the entire input. It needs a small number of labelled data sets (for output) rather than input data. Figure 13 briefly compares DBN and DBM [6, 23].

DBN prevents local optima and needs unsupervised learning data that do not need to be labelled because it considers greedy learning for parameter initialisation of the network. They are also resistant when the amount of data is low.

3.3 | Autoencoder-based methods

In the reverse of training to predict, they trained to reconstruct its input. Autoencoder, a special neural network, relies on reconstruction and unsupervised learning. Figure 14 shows the whole procedure of these networks.

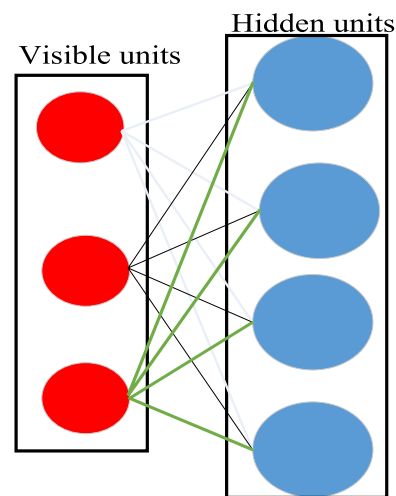


FIGURE 11 Restricted Boltzmann machines structure.

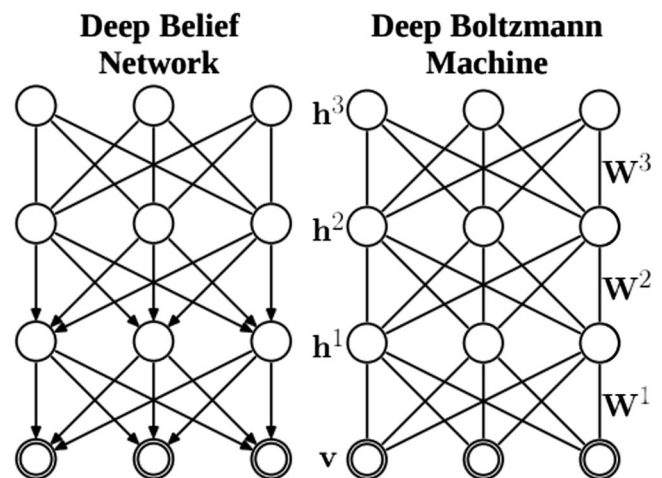


FIGURE 12 Structure difference between (a) deep Boltzmann machines and (b) deep belief network [21].

	Advantages	Disadvantages
Deep Belief Network	<ol style="list-style-type: none"> 1. Some how prevents local optima 2. Using unsupervised learning that does not need much labeled data 	<ol style="list-style-type: none"> 1. It is expensive to build DBN because of high computational need
Deep Boltzman Machine	<ol style="list-style-type: none"> 1. It faces more robustly with ambiguous data 	Optimization is time-consuming

FIGURE 13 Pros and cons of deep Boltzmann machines and deep belief network (DBN).

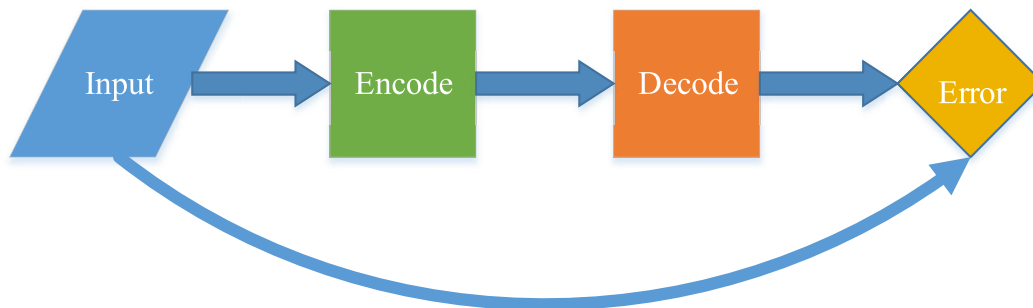


FIGURE 14 Autoencoder procedures.

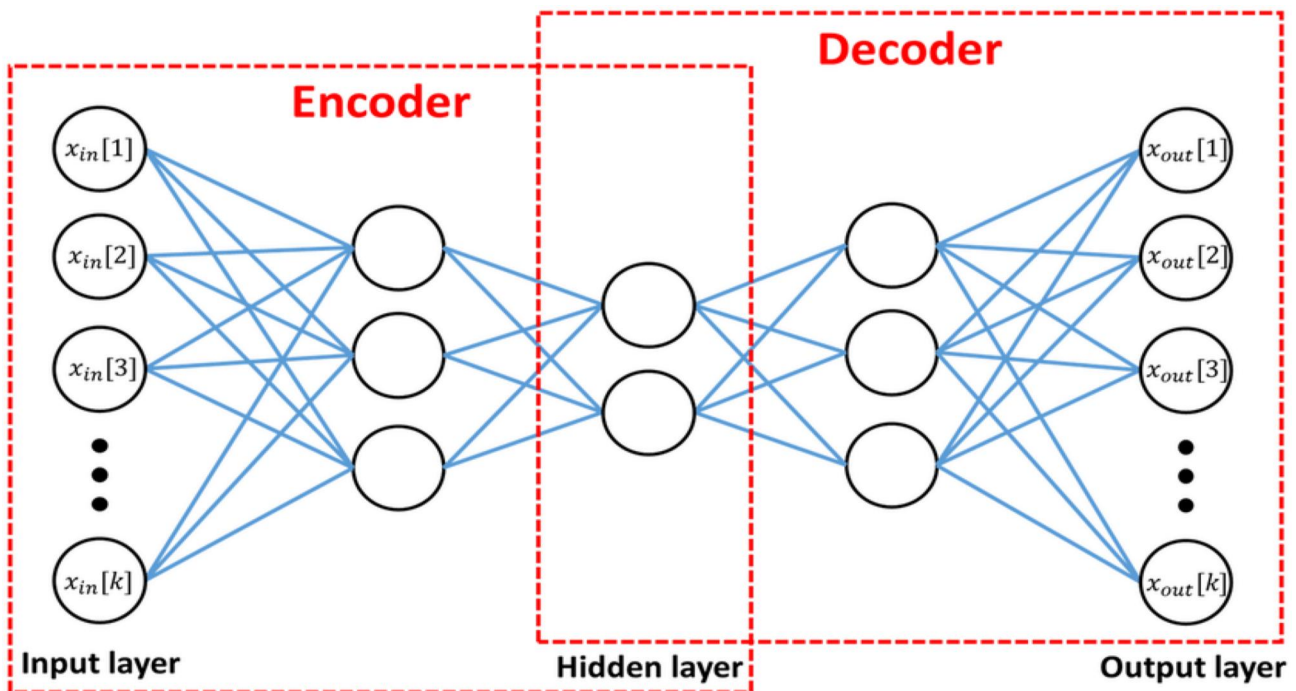


FIGURE 15 Autoencoders structure [24].

Figure 15 describes the general structure of Autoencoder methods. Most deep Autoencoders are trained using back-propagation methods such as conjugate gradient to rely more on earlier layers. There are two common types of Autoencoders: Denoising and Contractive. Denoising Autoencoder (DAE) is an enhanced autoencoder strong to the input by corrupting the

original data first and then re-creating the original input to minimise the error formula. DAE proceeds by imposing some noise into the input vector and then transferring it into the hidden layer while trying to reconstruct the real vector. Contractive auto-encoder is another enhanced auto-encoder to learn robust features by introducing the Frobenius norm of the

Jacobian matrix of the learnt feature concerning the original input [25]. They add analytic contractive error Denoising to the error reconstruction formula with the aim of better performance instead of denoising one that poses noises into inputs. Autoencoder is well-nigh more robust than other counterparts. In other words, it can reconstruct correct input from a corrupted version of input data. Autoencoders do not have the backward pass. RBMS is the same as Autoencoders in that they both are about unsupervised learning, feature extraction, feature transformer, and/or smart weight initialisation.

3.4 | Recursive neural tensor nets (RNTN)

A Recursive Neural Tensor Network (RNTN) algorithm is another kind of Recursive Neural Network (RecNN) algorithm, which is utilised to model input data with changeable length properties and tree structure dependences between input features. For computing the input presentation with RecNN, the input should be parsed into a binary tree where each leaf node represents input data. Standard RecNNs have some restrictions, where two vectors only implicitly rely on addition before using a non-linear activation function. Standard RecNNs are not able to model very long-term dependence on tree structures. Zhu et al. suggested the gating mechanism in the standard RecNN model to handle the latter problem. For the previous limitation, the RecNN accomplishment can become good by adding more interplay between the two input vectors. So, a new architecture named a Recursive Neural Network with a tensor-based feature function (RecNTN) tried to overcome the former problem by adding interplay between two vectors utilising a tensor product linked by tensor weight parameters. Each slice of the tensor weight can help catch the pattern between the left and right child vectors [26].

The goal of these types of networks is to study the hierarchical shape of dataset [27]. In other words, it discovers the hierarchy structure of a data set. So, it is suitable for sentiment analysis [28]; if we want to analyse sentiment in a group of words, we cannot explore solitary words and consider them as a group. From another point of view, we have to analyse them as a group, not only in words. We can use them for syntactic parsing and image parsing too. From the structure aspect, it seems like a tree that each component is a group of neurons.

3.5 | Deep belief networks (DBNs)

RBM are necessary to make DBNs. Initialising the weights of DBNs is described as an ill-posed problem. The weights in non-linear belief networks with more hidden layers are difficult to gain. As previously mentioned, efficient work to initial weights permits gradient descent to compute training weights efficiently. However, finding the initial weights needs an approach that simultaneously learns a layer of features. Researchers have challenges in handling this prob by training a sequence of RBMs. Therefore, a DBN trains some RBMs relying upon some hidden layers. The first RBM contains the DBNs input layer,

and the first hidden layer, the input of RBM, acts as a training set. For the second RBM, which creates the DBNs' first and second hidden layers, the information proceeds RBM and continues [29].

This type of network aims to find the dependencies between the hidden and noticeable layers that are mutually dependent [30]. In this regard, DBNs were created by accumulating a bank of RBMS conducted by Hinton in 2006. This type of network is effective when we have problems using unlabelled data [31]. In the DBN, there are two stages in the training process, including pre-training and fine-tuning stages, wherein the pre-training, an unsupervised learning method, is applied to the network in down-up directions. In contrast, a supervised learning method regulates the network parameters in the fine-tuning step. The pre-training step is essential in the network's final performance, where the network's initial weights should learn from the input data.

Different forms of DBN have been suggested. For example, Liu et al. proposed a method based on DBN to gain a good covariance kernel for the Gaussian process. They revealed the good accomplishment of the suggested approach both in regression and classification [32]. In another study, a new convolutional DBNs algorithm was proposed to enhance DBN's capability in extracting the features of high-dimensional images [33].

3.6 | Recurrent neural network

This method is used mainly when the pattern of data changes over time. A straightforward structure with a built-in feedback loop acts as a forecasting engine. It has incredibly variable applications ranging from speech recognition to driverless Automobiles. We can see the form of this NN in Figure 15. Unlike feedforward nets, RNNs can take a sequence of values as input and create a series of values as output for forecasting [34]. Still, we should use Feedforward nets if we want classification/regression. For example, when the input is singular, and the result is the sequence, the potential application is image captioning or image labelling, Sequence input, and single outcome such as document classification. One possible application is video classification frame by frame when the input is a sequence, and the output is also a sequence. If we face a delay, the net can statistically forecast the demand. RNN has one major drawback: it is a highly complex DL network to train. It uses Backpropagation and, unfortunately, vanishing gradients. The main problem of backpropagation networks shows itself worse than other networks here. It is likely to decay information during this time [35]. Some methods were proposed for addressing this issue, such as using Gating units: LSTM-GRU [36].

Based on the architecture, we usually use GPU rather than CPU. The difference between GPU and CPU is high for processing. Approximately GPUs are 10 times faster, leading to 1-day training rather than 8 months (CPU). We can even train faster if we use Intel's BigDL DL framework introduced in 2017, a Spark-powered framework for distributed DL, obtainable as an open-source project [37]. In brief, CPUs are optimised for serialised tasks in reverse of GPUs that are good for parallel

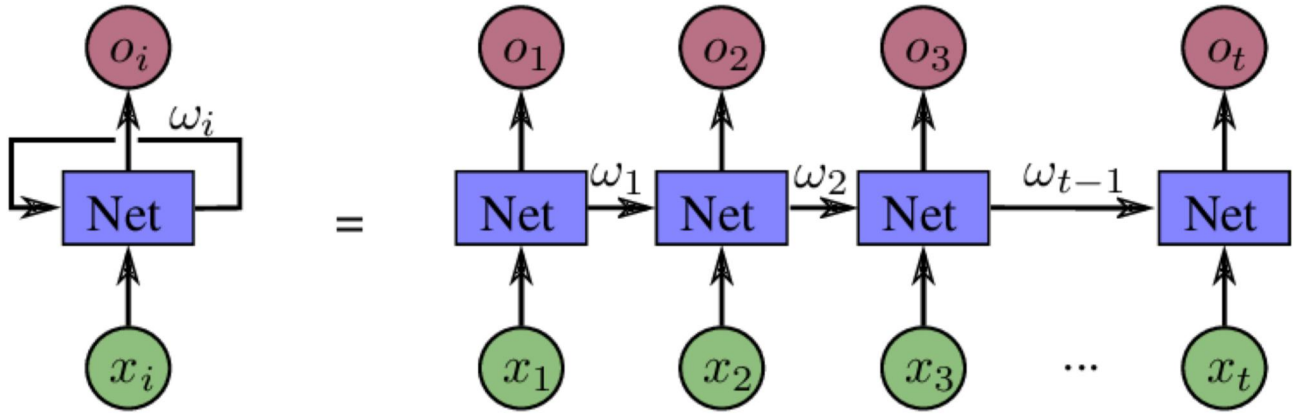


FIGURE 16 Recurrent neural network structure (O_i : result of the Network, W_i : weight of the Network, X_i : Input of the Network) [38].

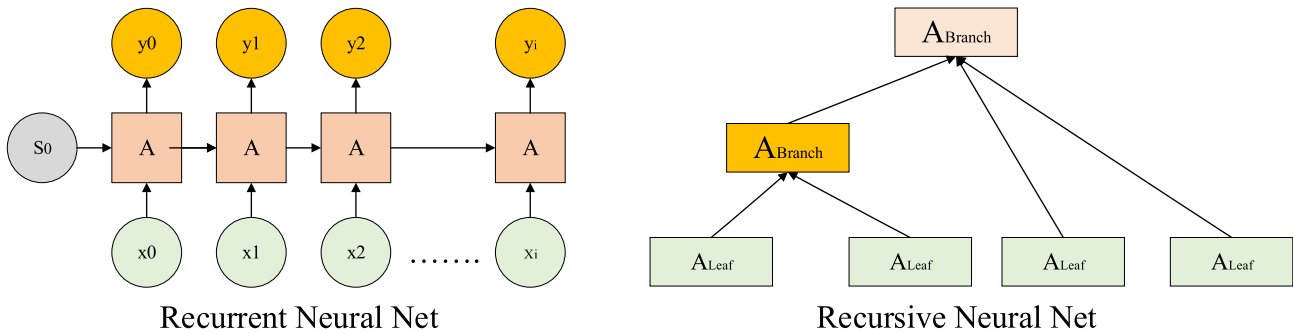


FIGURE 17 Recurrent neural network and RNTN structures.

tasks. Unlike feedforward nets, RNNs can take a sequence of values as input and create a series of values as output for forecasting. Still, we should use Feedforward nets if we want classification/regression. We can stack up a single RNN to form a complex network to achieve and find more valuable information from input data. This capability can deal with sequences and open this net to various applications. For example, when the input is singular, and the output is the sequence, the potential application is image captioning or image labelling, sequence input, and single outcome, such as document classification. One possible application is video classification frame by frame when the input is a sequence, and the output is also a sequence. If we face a time delay, the net can statistically forecast the demand; for instance, it is useable to predict the next character in the given input text. But it has, one major drawback is that RNN is a highly complex DL network to train. It uses Backpropagation and, unfortunately, vanishing gradients. The primary problem of backpropagation networks shows itself worse than other networks here, so it is likely to decay information during the time [35]. Some methods were proposed to deal with this issue, such as using Gating units: LSTMGRU24, which we later describe as DL problems (Figure 16).

Figure 17 compares RNTN and RNN from a structural aspect. As we can see, RNTN is almost parallel, but RNN is serialised, although the complexity of both stems from the way data moves throughout the network. RNTN works best when input data depends on vector representations. The existence of

cycles allows the introduction of new elements of dynamic behaviour dependent on the output results.

We can consider this parameter when choosing the best method corresponding to our application, as shown in Figure 18.

Suppose we want to compare them in more detail. In that case, we can present aspects like: the generalisation aspect, which means whether we can use them in a variety of applications or not, the unsupervised learning aspect means if they can learn a deep model via an unsupervised method or not, feature learning means if the model can find features automatically or not from the data set and invariance means if the model resistant to changes or not (Figure 19).

4 | EXAMPLES OF RECENT WORKS IN IoT

In this section, we present recent works comparing traditional and DL methods.

4.1 | The first example of CNN in IoT

In the OP performance analysis of mobile IoT connection networks is conducted by writers. They also suggest an OP intelligent forecast method corresponding to an improved

Data Sector	Use Case	Input	Transform	Neural Net
Text	Sentiment analysis	Word vector	Gaussian Rectified	RNTN or DBN (with moving window)
	Named-entity recognition	Word vector	Gaussian Rectified	RNTN or DBN (with moving window)
	Part-of-speech tagging	Word vector	Gaussian Rectified	RNTN or DBN (with moving window)
Text	Semantic-role labelling	Word vector	Gaussian Rectified	RNTN or DBN (with moving window)
Document	Topic modelling/ semantic hashing (unsupervised)	Word count probability	Can be Binary	Deep Autoencoder (wrapping a DBN or SDA)
	Document classification (supervised)	TF-IDF (or word count prob.)	Binary	Deep-belief network, Stacked Denosing Autoencoder
Image	Image recognition	Binary	Binary (visible and hidden)	Deep-belief network
		Continuous	Gaussian Rectified	Deep-belief network
	Multi-object recognition			Convolutional Net, RNTN (image vectorization forthcoming)
	Image search/ semantic hashing		Gaussian Rectified	Deep Autoencoder (wrapping a DBN)
Sound	Voice recognition		Gaussian Rectified	Recurrent Net
				Moving window for DBN or ConvNet
Time Series	Predictive analytics		Gaussian Rectified	Recurrent Net
				Moving window for DBN or ConvNet

FIGURE 18 Selecting the best methods based on usage [39]. DBN, deep belief network; SDA, Software defined architecture.

CNN. First, the mobile OP performance is studied by combining the Total Access System and decode-and-forward cooperative plans, and the exact OP expressions are derived.

An improved CNN is also created to avoid the loss of required information that contains the input layer, three convolution layers, one FC layer, and an output layer.

	RBM	CNN	AutoEncoder	Sparse Coding	RNN	RNTN
Generalization	Yes	Yes	Yes	Yes	Yes	Yes
Unsupervised learning	Yes	No	Yes	Yes	Yes	Yes
Feature learning	Weak	Yes	Weak	No	Weak	Weak
Invariance	No	Yes	No	Yes	No	No

FIGURE 19 Comparing some deep learning methods. CNN, convolutional neural network; RBM, restricted Boltzmann machine; RNN, recurrent neural network

4.1.1 | Benefits of CNN in this study

The CNN-based forecast approach is compared with the radial basis function (RBF), generalised regression (GR), Elman, and extreme learning machine (ELM) algorithms by writers. The result of comparing algorithms is the proposed CNN forecast approach can get a better forecast effect than RBF, Elman, GR, and ELM methods. The forecast accuracy of the CNN approach has increased by 44% [40]. CS can recover images from fewer quantifications than traditional signal acquisition and reconstruction ways. Recently, many DL-based CS methods have been suggested for image reconstruction and to perform better than traditional CS reconstruction methods [41].

4.2 | The second example of CNN in IoT

In ref. [41], a new image quantification and reconstruction framework to gain high-quality reconstruction is proposed. Writers use a convolutional layer instead of a random quantification matrix to get all sizes in the measurement part, giving more construction information for subsequent image reconstruction and removing the block effect. This paper first utilises a deconvolution layer in the reconstruction part to gain an initial reconstructed image with the same dimension as the input image. Then we use multiple parallel CNN to get multiple feature information. The numerous parallel CNN comprise dilated convolution kernels of the different receptive fields to increase the network's receptive field, which can obtain more image structural information for reconstruction.

4.3 | An example of restricted Boltzmann machines in IoT

The vital goal of the study in ref. [42] is to get remarkable information from the images for better clinical diagnosis. Medical image analysis is an emerging study topic in the biomedical engineering domain. In recent decades, the usage of medical images growing, which are gained from different image modalities, so it is essential for data compression for transmission, storage, and management of digital medical image datasets. The automated DL models are powerful compared to the

conventional handcrafted features. Moreover, a wireless sensor network (WSN) helps produce a primary healthcare scheme.

In contrast, the WSN scheme must comprise sensor nodes due to consuming less power and resources at a relatively low cost. So it is necessary to implement the Raspberry Pi-based WSN nodes. The sensor nodes are required for limited battery capacity and to transmit extensive medical data [42].

4.3.1 | Benefits of restricted Boltzmann machines

The DL models are utilised in the ML tools to get the feature vectors from extensive medical datasets. The automated DL models are powerful compared to the conventional handcrafted features [42].

4.4 | An example of autoencoder-based methods in IoT

This paper [43] expresses a new autoencoder method corresponding to sparse coding. The vital contribution is sparsity coding for input feature selection and a subsequent classification utilising latent information. Precisely, a typical autoencoder architecture is used for reconstructing the original input. Then the sparse coding method is applied to optimise the network structure, select optimal features and enhance the generalisation capability. In addition, the refined latent information is also considered an alternative feature for training a sparse classifier.

4.4.1 | Benefits of autoencoder-based methods

Suggested method scores the best training and generalisation outcome for forecast, compared to Random Forest, support vector machine (SVM), and Artificial Neural Network. Moreover, writers understand that the Random Forest method has a remarkable variance in the testing set, which might be created by combining individual trees to form the outcome. By contrast, the suggested way has a relatively fixed performance from the training and testing sets. Finally, it is empirically confirmed that the presented method leads to an essential improvement compared to other methods in forecast accuracy [43].

4.5 | The first example of deep belief networks in IoT

In ref. [44], the intrusion detection model for IoT is designed utilising the Taylor-Spider Monkey optimisation and progressed to train the Deep belief neural network (DBN) towards gaining an accurate intrusion detection model. The DL accuracy increases with an added number of training data samples and testing data samples. The optimisation-based algorithm for training DBN helps to reduce the Floating-Point Register and format and restore in intrusion detection. The system is implemented by utilising the NSL knowledge discovery and data mining dataset. The samples from this dataset are also used for training this model, before which feature extraction will be applied. Only a relevant set of attributes is chosen for model development.

4.5.1 | Benefits of deep belief networks

More of these system models are progressed by utilising traditional ML techniques, which have both performance limitations in terms of accuracy and timeliness [44].

4.6 | The second example of deep belief networks in IoT

In this proposed heart disease monitoring system [45], the disease level is monitored according to writers' inputs given to the IoT devices. It also classifies the patient details according to the heart disease's kind and severity of the disease. Finally, it provides an alarm/message to the patients according to the type of heart disease with the available inputs.

4.6.1 | Benefits of deep belief networks

DL algorithms perform better than ML algorithms when they control extensive data that consumes more time. DL algorithms learn the data set repeatedly instead of once in the training process. The dataset's features can be studied in depth by incorporating repeated learning [45].

4.7 | The first example of recurrent neural network in IoT

In ref. [46], environmental noise data are gained by using an IoT-based noise monitoring system. A two-layer long short-term memory (LSTM) network is implemented to predict environmental noise under the condition of large data volume. Then, the optimal hyperparameters are chosen through testing, and the raw data sets are processed.

4.7.1 | Benefits of recurrent neural network method

By comparing the Performances of this study with three classic predictive algorithms: random walk, stacked autoencoder, and SVM, writers conclude that the proposed model outperforms the other three existing classic algorithms. The data set's time interval closely connects with the performance of all methods. The results revealed that the LSTM network could reflect changes in noise levels within 1 day and has good forecast accuracy [46].

4.8 | The second example of recurrent neural network in IoT

In this paper [47], writers suggest an artificially intelligent system for reducing the congestion effects in traffic load in an Intelligent Internet of Things network based on a DL Convolutional Recurrent Neural Network with a modified Element-wise Attention Gate. The invisible layer of the modified Element-wise Attention Gate structure has self-feedback to increase its LSTM. The artificially intelligent system is created for the next step ahead of traffic estimation and clustering of the network. This study shows that researchers using DL and ML methods can effectively handle IoT problems [47].

4.9 | The first example of CNN-LSTM in IoT

In ref. [48], writers propose a prediction model for improving underground coal mines' safety and productivity using a hybrid CNN-LSTM model and IoT-enabled sensors. Spatial and temporal features from mine data and efficiently forecast different mine hazards are extracted by the hybrid CNN-LSTM model. The suggested model also improves the flexibility, scalability, and coverage area of a mine monitoring system in an underground mine's remote locations to minimise the loss of miners' lives. The suggested model efficiently predicts miners' health quality index for working faces and gases in goaf areas of mines.

4.9.1 | Benefits of CNN-LSTM method

This paper mentions the forecast ability of CNN, LSTM, and hybrid CNN-LSTM algorithms. CNN method extracts only the time-invariant features from time interval data, whereas the LSTM network reports only long-standing dependency on time interval data. Nevertheless, the hybrid CNN-LSTM method outperformed the existing CNN and LSTM models by extracting spatial and temporal features from multi-dimensional and multiple time interval data. Thus, the hybrid

CNN-LSTM network increases the predicting accuracy and efficiency for time interval data [48].

4.10 | The second example of CNN-LSTM in IoT

In this paper [49], CNN-LSTM is proposed for a fall detection algorithm according to CNN's robust feature extraction ability and the excellent time series processing ability of LSTM. The algorithm requires only the resultant acceleration from a low-cost three-axis acceleration sensor.

4.10.1 | Benefits of CNN-LSTM method in this study

After comparing the algorithms according to SVM and CNN, writers conclude that the suggested method has higher detection accuracy with a small data volume, which is very appropriate for IoT enabled fall detection applications [49].

5 | DEEP LEARNING FRAMEWORKS AND TOOLS

In this part, we compare some frameworks and tools that are necessary for implementing DL and are famous as well that can be divided into 16 main categories:

5.1 | Theano

Frederic Bastien built it in the University of Montreal's lab, MILA group. It is probably most welcomed in academia. It is a python with GPU based library. It is written in python and optimised with C. Besides, it provides pretty low-level Application Programming Interfaces (APIs). So, to write Theano effectively, you require to become familiar with detailed methods. From another point of view, Theano requires substantial academic ML expertise, but it is more extensible and flexible (Figure 20).

5.2 | Lasagne

It provides a higher-level library based on Theano for building and training NN. Lasagne permits users to think at the Layer level [53] (Figure 21).

5.3 | Blocks

Blocks aims to add a higher-level abstraction layer for Theano to facilitate simpler definitions of DL methods rather than writing raw Theano [55]. Blocks have strong support for RNN architectures (Figure 22).

5.4 | TensorFlow

It is a new backend version of DistBelief from Google's open-source project. Recently Google released the first version of TensorFlow called TensorFlow 1.0, including faster operation and a new compiler for hardware accelerators [56]. Raw TensorFlow is a blend of lower-level libraries like Theano and higher ones, the same as Blocks. It is probably the second choice for beginners, but it supports Linux and Windows (Figure 23).

5.5 | Keras

It is a high-level library at the top of both Theano and TensorFlow for neural network manipulation. It omits the requirements for writing low-level codes, and it is user-friendly. It provides a higher level of programming requirements. When writing this essay, Keras is compatible with: Python 2.7–3.5. This makes building common deep architectures very straightforward and clear. All the components needed to create them are already provided (layers, loss function, training method, drop-outs etc.), and it is modular, so it is very intuitive (Figure 24).

5.6 | Torch

Facebook, Twitter, and Yandex rely on Torch with two types: traditional was based on LuaJIT language and python based named PyTorch, released in January 2017 by Facebook (Figure 25).

5.7 | MXNet

In November 2016, Amazon introduced a high-performance framework with GPU support called MXNet to provide efficiency and flexibility. MXNet is perhaps the most performant library of frameworks [58] (Figure 26).

A week after backing MXNet, Amazon introduced additional useable services for text-to-speech recognition and an image analysis service called Recognition [59].

5.8 | CNTK (computational network toolkit)

Microsoft released a beta of its toolkit [60]. The main purpose was speech recognition, but it changed its direction to expand its domains. CNTK offers a Python API over C++ code. It supports two interfaces, Python and C++ (Figure 27).

5.9 | MatConvNet

It is a toolbox in Matlab 2017, and was recently added to Matlab 2016. It is suitable for people who are forcefully working on Matlab language (Figure 28).

	Pros:	Cons:
	<ul style="list-style-type: none"> 1) It is well-nigh rapid. 2) It is strong in computationally intensive projects [50]. 3) Flexible. 4) Network specification library. 5) Cross-platform. 6) Fluent in sequence processing algorithms such as a recurrent neural network. 7) It is strong for matrix operations. 	<ul style="list-style-type: none"> 1) It provides a low-level API that is not easy to use [51]. 2) Currently, it does not provide a multi-node and distributed implementation. In other words, training a net in the Hadoop cluster is impossible [52]. 3) It could be hard for beginners to use since it is hard to write, and the error messages can be unhelpful.

FIGURE 20 Pros and cons of Theano.

	Pros:	Cons:
	<ul style="list-style-type: none"> 1) Higher abstraction level than Theano. 2) Network specification library. 3) It does not specify the network model according to the functional connection between symbolic variables. Still, it lets users to think at a level, providing users with building blocks such as "Conv2DLayer" and "DropoutLayer." 4) It provides a wealth of common components to help layer definition, layer initialization, model regularization, model monitoring, and model training. 5) Docs and code contain an assortment of pasta puns. 	<ul style="list-style-type: none"> 1) Smaller community.

FIGURE 21 Pros and cons of Lasagne [54].

	Pros:	Cons:
	<ul style="list-style-type: none"> 1) Easy to test. 2) It causes the deep learning model clearer, simpler, and more standardized than writing the original Theano. 3) In comparison to Lasagne, it is a bit more flexible, at the cost of a higher entry-level, and is tough to use efficiently. 4) It has good support for recurrent neural network architectures. 	<ul style="list-style-type: none"> 1) Smaller community. 2) higher learning difficulty.

FIGURE 22 Pros and cons of Blocks [54].

	Pros:	Cons:
	1)Running machine learning models across multiple GPUs, and even it supports CPUs. 2)Efficient data pipelining. 3)Has built-in modules for visualization of models. 4)Is it the most welcomed library recently? 5)It has computational graph abstraction like Theano. 6) easy install on Windows 7,10 and server 2016	1)It is well-nigh-hard to use raw TensorFlow. 2)For the RNN model, Theano is more rapid than Tensor-Flow. 3)Good support for visualization. 4)It is slower than other frameworks. 5)Do not contain many pre-trained models.

FIGURE 23 Pros and cons of TensorFlow. RNN, recurrent neural network

	Pros:	Cons:
	1)We are running machine learning models across multiple GPUs. 2)Efficient data pipelining. 3)It has built-in modules for visualization of models. 4)It supports all of the DL models.	1)Less flexible and more prescriptive than other options. 2)There are several features in Keras that have space for development. It lacks some pre-trained models to utilize. Keras does not support features of dynamic chart creation. 3)The errors given by the Keras library are not effective. There is necessary for the errors to be easily identifiable. It is not very useful and helpful to find the root cause of the error. It is hard to debug in Keras. 4)Keras gives you low-level errors many times. The reason for these low-level errors is that there are some functionality and operation that Keras is not capable of. 5)Also, Keras is not capable of low-level handling computations.

FIGURE 24 Pros and cons of Keras [57]. DL, deep learning

5.10 | Deeplearning4j

It is a java based distributed DL framework that differs from API languages, intent, and integration from others. It integrates with Hadoop and Spark [63]. It uses an arbitrary of CPUs and GPUs. An image processing task on multiple GPUs had the same performance as Caffe but was better than TensorFlow or Torch and twice that of Theano. Currently, it is useable in NASA (Figure 29).

5.11 | Caffe

The winner of ImageNet in 2014 [65] created it. It uses C++ with Cuda, so applications can easily switch between CPU and GPU. It mainly focuses on computer vision forecasting applications, which are well suited for convolutional nets. But the recent version supports speech and text, reinforcement learning, and recurrent nets for sequence processing. It is a DL framework with expression, speed, and modularity. It is almost

	Pros:	Cons:
	1) It supports lots of functions that have easy combination. 2) It has lots of pre-trained models that are usable. 3) It has comprehensive documents. 4) In the writing time, it does not support Windows.	1) Less plug-and-play, and we must write our training codes most of the time. 2) Currently, only for Linux-based platforms.

FIGURE 25 Pros and cons of Torch.

	Pros:	Cons:
	1) Portability characteristics open to various languages such as R, Go, Python, Matlab, and JavaScript. 2) Supports multiple nodes of GPUs, not just for multiple GPUs and various interfaces. 2) You can use it even on your mobile phone. 3) Fast benchmarks. 4) Flexible.	1) Steeper learning curve than Theano. 2) The smallest community

FIGURE 26 Pros and cons of MXNet.

	Pros:	Cons:
	1) It supports RNN and CNN. 2) Parallel execution is possible.	1) It does not support RBM and DBNs.

FIGURE 27 Pros and cons of CNTK. CNN, convolutional neural network; CNTK, computational network toolkit; DNN, deep neural network; RBM, restricted Boltzmann machine; RNN, recurrent neural network

	Pros:	Cons:
	1) It supports both Windows and Linux. 2) It supports GPU. 3) It has trained models the same as Caffe [61]. 4) Matlab, easy to use.	1) Complicated than DeepLearnToolbox. 2) It Supports only Convolution Network.

FIGURE 28 Pros and cons of MatConvNet [62].

suitable for beginning and starts without a sophisticated set of configurations of layers [66]. You can use premade nets to create highly complex nets. Users can put their model in the

'Model Zoo' [66]. Caffe vectorises input data through a special data representation named 'blob.' It has two branches, one Linux-based and the other Windows-based (Figure 30).

5.12 | SparkNet

UC Berkeley created this, a system for training NN based on the Spark distributed computing platform [66]. In other words, it is a framework for training deep networks in Spark [67] (Figure 31).

5.13 | Nvidia DIGITS

It is a neural network tool for data scientists that is strong for object recognition, image classification, and segmentation [69].

At first, it used Caffe as a backend, but now Torch and TensorFlow are useable (Figure 32).

5.14 | BigDL

It is a distributed DL framework for Apache Spark that was introduced in January 2017 by Intel [71]. It gives comprehensive support for DL, including numeric computing and high-level NN. It is written in Scala. More than 100 layers have been supported, including important and widely used layers such as Linear, Spatial Convolution, Recurrent etc. (Figure 33).

	Pros:	Cons:
	1) It is portable and platform-neutral. 2) It uses Computational Graph for automatic differentiation. 3) It is good for people who want to combine Big Data and Deep Learning. 4) Good for distributed systems and Java-based ones.	1) Not convenient. 2) Limited to the Java programming language. 3) Relatively less popular compared to Tensorflow and PyTorch.

FIGURE 29 Pros and cons of Deeplearning4j [64].

	Pros:	Cons:
	1) It is suitable for image processing applications and feedforward networks. 2) Useful for fine-tuning existing networks. 3) User-friendly (training models without code). 4) Python interface. 5) It is suitable for feedforward networks. 6) It has many pretrained models in its Model Zoo repository.	1) Not suitable for RNNs networks and sequence models. 2) Difficult to extend it to support big networks. 3) It could be huge for big models. 4) It does not have cohesive documents.

FIGURE 30 Pros and cons of Caffe. RNN, recurrent neural network

	Pros:	Cons:
	1) It is easy to deploy. 2) Using without no parameter tuning. 3) Compatible with Caffe Models. 4) It is compatible with Apache Spark.	1) No automatic optimization process. 2) Fewer Algorithms. 3) Small Files Issue. 4) It is not suitable for a multi-user environment.

FIGURE 31 Pros and cons of SparkNet [68].

	Pros:	Cons:
	1) They are suitable for designing, creating, and visualizing objects. 2) Using trained models such as Google Net. 3) Scaling training automatically to multiple GPUs. 4) It is compatible with Apache Spark.	1) It forces the user to pay for features (RTX, deep learning) that do not generally translate well to games. 2) It does not provide a better experience to justify the cost (individual games like Minecraft RTX are not considered).

FIGURE 32 Pros and cons of Nvidia DIGITS [70].

	Pros:	Cons:
	1) It has great support. 2) Best suitable for users to leverage existing Spark/Hadoop clusters and analyze a large amount of data on the same Big Data cluster where the data are stored. 3) We can load pre-trained Caffe and Torch models into the Spark framework and then use BigDL. 4) Extremely high performance. 5) It is compatible with Apache Spark. 6) Recently, they supported python programming with BigDL. 7) Users can conveniently utilize the Jupyter notebook to write codes and visualize their training process. 8) High single load Xeon performance	1) No automatic optimization process. 2) Fewer algorithms. 3) Small files issue. 4) It is not suitable for a multi-user environment.

FIGURE 33 Pros and cons of BigDL. DL, deep learning

5.15 | TFlearn and TFSlim

These are two high-level libraries on top of TensorFlow.

5.16 | Chainer

Another framework that is python based is Chainer, which is runnable on multiple GPUs without needing too much effort. This framework revealed a good performance for speech recognition, machine translation, and sentiment analysis [72]. This framework can support various networks, such as CNNs, RNNs etc. Some pros and cons of this framework are describable as follows (Figure 34):

6 | APPLICATIONS

There are various applications that DL supports. Some of the major applications that use the concept of DL are considered as follows [73]:

6.1 | Automation of black and white images colouring

In traditional times, the colouring of black and white images was carried out with the hand. The manual human effort was required in this process, making this a difficult task. With the arrival of DL, objects and their content within the image is similar to the human eye depicting an image for colouring it. Figure 35 represents the scenario in which the black and white images are automatically coloured.

6.2 | Automatic sounds addition to silent movies

The training set used for preparing this system consists of 1000 examples. This system is trained using videos containing drumstick sounds striking at different surfaces. Due to this, they produce different sounds. With the emergence of DL, the DL model is associated with video frames that contain a database of recorded sounds. This pre-recorded sounds

	Pros:	Cons:
	1) It is fast concerning other leading python frameworks. 2) It is flexible and intuitive. 3) It is an open-source framework. 4) Existing networks are modifiable at runtime.	1) The debug process is not straightforward. 2) It is relatively small in terms of the community.

FIGURE 34 Pros and cons of Chainer.

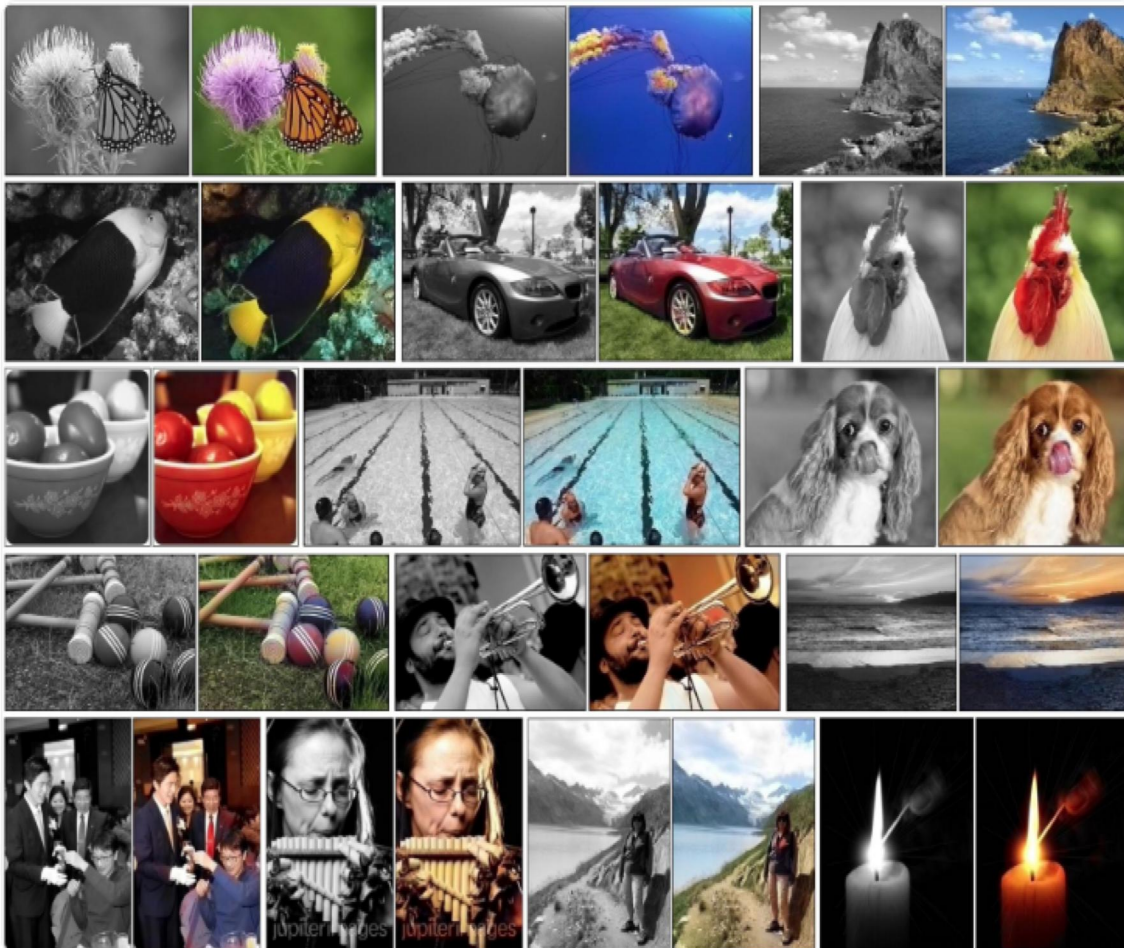


FIGURE 35 Automation black and white images colouring [30].

database selects the background for the movie's scenes. The song is chosen because it matches best with what is happening in the movie scene.

6.3 | Detection and classification of objects in photographs

DL helps to detect various objects in the existing photographs and draw a box around them. With this detection, we can easily classify the photographs' objects, which helps to understand

the images easily. The DL approach brought ease in understanding photographs using object detection, a more complex variation task. Figure 36 shows the detection and classification of various objects in photographs (Figure 37).

6.4 | Automation of machine translation

This application of DL is beneficial for translating tasks (words, sentences, or phrases) from one language to another. This process of translation is made automatic with the help of

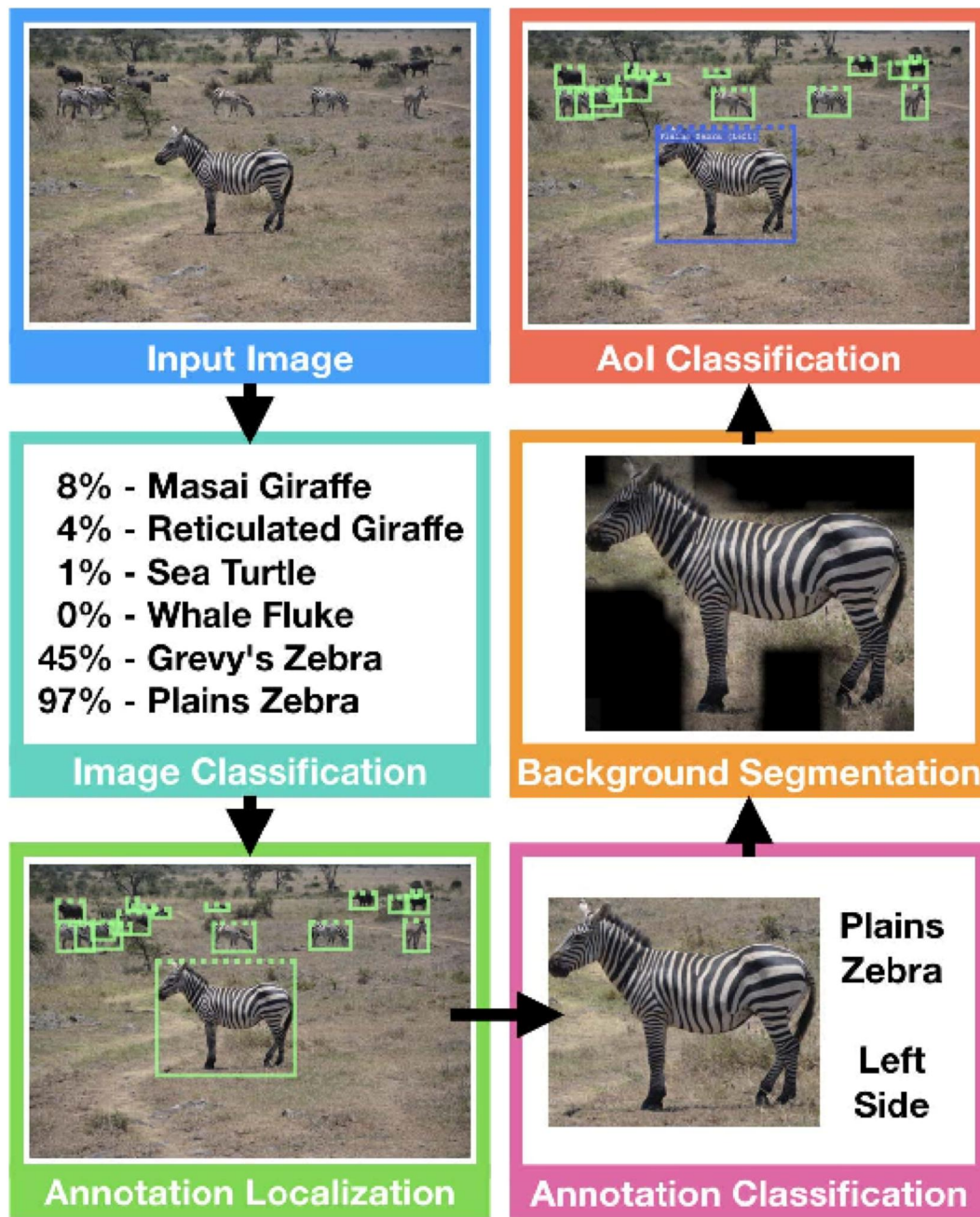


FIGURE 36 Detection and classification of objects in photographs [74].

DL. This process has main applications in the automation of image translation and the automation of text translation. Automation of text translation is performable without any sequence preprocessing directly with the DL algorithm.

6.5 | Automatic generation of handwriting

DL is essential in automatically generating new handwriting for a phrase or a word. It is accompanied by using the coordinates provided by writing the text with a human hand using the pen and collecting samples using these coordinates. These samples are then used to calculate the relationship between the

movement and the pen, which acts as learning for the DL mechanism. It is then used to generate new examples using these samples.

6.6 | Medical images analysis and deep learning

DL has been comprehensively used for medical image analysis [15]. For example, Tange et al. used imaging of phenotype and genotype to forecast the survival time of Glioblastoma Patients. They proposed a multi-task CNN to perform both the tumour genotype and overall survival time

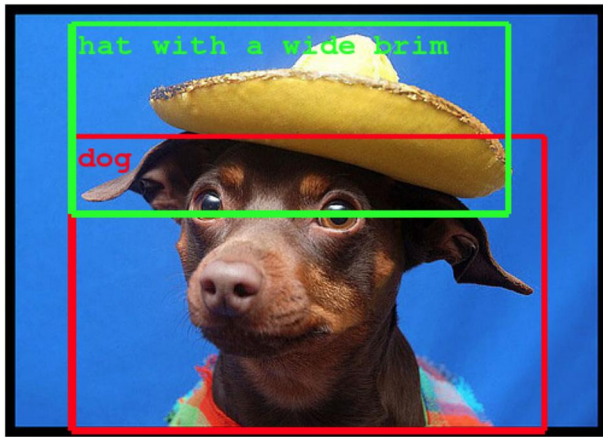


FIGURE 37 Object detected from the image [30].

together and could increase the accuracy of predicting the survival time [75]. Sun et al. utilised a fully-formed, fully convolutional network (AG-FCN) and anatomical gated U-Net (AG-UNet). They applied the proposed approach for brain images' region of interest image segmentation [76]. Many problems in image analysis and segmentation using DL are suggested in the literature. Some of them can be found in refs. [77–79].

6.7 | Intelligent transportation systems and deep learning

DL played a pivotal role in Intelligent Transportation Systems (ITS). For example, in ref. [80], a DL-based system of transportation analysis is proposed. This work utilised RBM and RNN architecture to perform the tasks related to the ITS analysis, where they could enhance the accuracy of predicting traffic congestion by 88%. Ciresan et al. [81] proposed a traffic sign recognition system in another study. They used DNNs of Convolutional and max-pooling layers and presented multi-column DNN architecture. The results of their proposed works reveal a superior accuracy of 99.46% in recognising traffic signs. DL has also been applied to self-driving cars, where utilising DNN in tasks like detecting obstacles and pedestrians is very important [82].

6.8 | 3D objects and deep learning

DL has been used for applications with 3D objects in a widespread manner. For example, Yu et al. proposed a vision-based robotic grasping system using a deep neural network for 3D object recognition. They used the Max-pooling Convolutional Neural network in the proposed system, where they fed the network with different poses of objects [83]. Socher et al. utilised CNN and RNN to classify 3D objects, where their proposed approach yields superior results [84]. In another research, Leng et al. utilised a DBN (DBN) in performing a 3D model classifier.

They first applied DBN to extracting the features of the 3-D model and then used a semi-supervised method to recognise 3D objects adopting the obtained feature from DBN [85]. Deep Boltzmann Machine is also applied to 3D model recognition in ref. [86], where it is utilised to extract features of the input data.

6.9 | Natural language processing (NLP) and deep learning

DL applied to Natural language processing (NLP), where many studies revealed different usage of DL-based systems for grammar correction, author mimicking, extraction of writer's sentiment such as their positive, negative, and neutral ideas [87]. Other applications such as spam classification and social media analysis are also different types of DL applications. Most of them utilise LSTM and CNN [88–90].

6.10 | Voice generation and deep learning

Recently DL has had great importance in voice generation and speech synthesis. DL techniques revealed great effectiveness in obtaining the parameters and statistical model effectively defining the signals. Some works have investigated how to generate waveforms out of the parameters using DL. For instance, Sánchez utilised different architectures such as LSTM and tried to create signals [91]. In another study, Arik et al. suggested DL-based text-to-speech systems to create a high-quality system entirely using DL [92]. Many works in the literature focus on DL for voice issues [93, 94].

6.11 | Health monitoring and deep learning

DNN, one-dimensional CNN, and vision-based methods using two-dimensional CNN in the picture and video data set have been widely utilised in structural health monitoring and damage detection from vibration datasets. For example, Abdeljaber et al. [1] examine the performance of state-of-the-art 1-D CNN in vibration-based structural damage detection on a well-known IASC-ASCE phase II benchmark problem. Kiranyaz et al. [2] present a complete review of the general architecture and principles of 1D CNNs on a large mock-up stadium structure built in a laboratory. Cha et al. [3] use Faster Region-based CNN to provide quasi-real-time multiple damage detection on the five kinds of damages concrete crack, steel corrosion, bolt corrosion, and steel delamination image dataset. Liu and Zhang [5] use the CNN model to inspect steel members in steel structures rapidly. Using unmanned aerial vehicles (UAVs) for data acquisition and visual inspection of infrastructure has been recently considered. Kang and Cha [6] provide a visual inspection with collected video from and UAV for detecting concrete cracks using deep CNN damage.

6.12 | Image processing and deep learning

DL has played a significant role in image preprocessing, and many kinds of study have been conducted in this field. Below we talk about some of these. In ref. [95], Considering the significance of medical image classification and the specific challenge of small medical image datasets, this paper chose how to use CNN-based classification for small chest X-ray datasets and evaluate their performance. This paper shows that CNN-based transfer learning is the best of the three algorithms in ref. [15, 96]. The DL model solution suggests an adaptive size. First, the input rules are suggested, then the optimal size set module is gained by the input rules, and the adaptive DL model module is suggested in combination with the optimal size set. Finally, the model combines the size fine-tuning module to create an adaptive size DL model solution. The classification effect and robustness of the model are verified through the pneumonia CT image dataset. Experimental outcomes show that the suggested solution has better consistency and robustness. In ref. [97], A DL algorithm based on a partitioned CNN for an automatic cancer detection system is proposed and developed. Researchers match the performance of the designed deep CNN with other conventional classification methods such as SVM and DBN. The accuracy obtained by this regression-based partitioned algorithm is higher than different basic classifiers in terms of sensitivity, specificity, and accuracy. In ref. [98], a DL-based CNN called deep CNN architecture is implemented, a general architecture that accepts the input as medical image data and causes the disease class or type. And researchers have compared it with classic models like SVM and elm. The authors in ref. [99, 98] introduce a novel hierarchical medical image classification technique, Hybrid Microwave Integrated Circuit, which can use several deep, complex neural network ways to generate hierarchical categories. In research results, researchers use two levels of the CNN hierarchy. Experiments on medical image datasets show that this method produces robust outcomes at higher and lower levels. The accuracy is consistently higher than conventional methods using CNN, multilayer perceptron, and deep convolutional neural network (DCNN). These outcomes show that the hierarchical DL method can improve classification and includes flexibility for classifying these data in a hierarchy. In ref. [100], a framework for automatic detection of breast cancer and Gleason prostate grading is proposed, and the following steps are done:

1. The value and phase of the mixed complex coefficients have been extracted from the histological images.
2. Shearlet features are combined with image data and utilised to train CNN. This feature learning process further enhanced the features and made them more distinct.
3. The Softmax classifier was used to distinguish different microscopic images.

The outcomes show that applying the method suggested in this article cause High classification accuracy in the breast cancer dataset and Gleason grading. Also, this method has

been compared with advanced techniques that use hand-made features. It shows that in both cases, those methods have been performed better.

The authors in ref. [101] present a CNN-based method with three-dimensional filters and applies to hand and brain MRI. Two modifications to an existing CNN architecture are discussed. While most of the current literature on medical image segmentation concentrates on soft tissue and the major organs, this work is validated on data from the central nervous system as well as the bones of the hand. Two changes to the U-Net architecture of Ronneberger et al. [102] were examined. The results show that, despite the relative scarcity of labelled medical images, 3D CNN architectures can achieve good quality results. In ref. [103], a new medical image classification algorithm is presented that combines high-level feature extraction from a coding network with traditional image features, and it is named the Conditional Neural Movement Primitive technique. This study is the first time a deep model is directly incorporated by incorporating features. Traditional imaging has been utilised to classify medical images. Experimental results show that this paper method can achieve higher accuracy than significant margins of SVM, coding network, and feature combination. In ref. [104], a novel deep neural network model is proposed to identify infected malaria falciparum parasites utilising a transfer learning technique. This suggested transfer learning method can be gained by integrating the existing Visual Geometry Network (VGG) and SVM. In this paper, pre-trained VGG facilitates the role of the expert learning model and SVM as domain-specific learning models. The initial ' k ' layers of pre-trained VGG are kept, and SVM replaces the $(n - k)$ layers. In ref. [105], a new image forgery detection way is suggested based on the DL method by using a CNN to learn hierarchical representations from the input RGB colour images automatically. The suggested CNN is designed explicitly for image splicing and copy-move detection applications. Rather than a random strategy, the weights at the first layer of our network are initialised with the basic high-pass filter set utilised to calculate residual maps in the spatial rich model. The pre-trained CNN is a patch descriptor to extract dense features from the test images. A feature fusion technique is then explored to obtain the final discriminative features for SVM classification. The experimental results on several public datasets show that the proposed CNN-based model outperforms some state-of-the-art methods. As mentioned in ref. [105], a DCNN can better detect regional wall motion abnormalities (RWMAs). It is recognisable between groups of myocardial infarction regions from conventional two-dimensional echocardiographic images compared to cardiologists, sonographers, and specialists. The present outcomes support the possibility of automatically utilising DCNN to detect RWMAs in echocardiography. In ref. [106], A promising two-step method for finding cases of COVID-19 while differentiating it from bacterial pneumonia, viral pneumonia, and healthy subjects with chest X-ray images utilising residual DL networks is suggested. The first stage model have good performance in differentiating between viral pneumonia, bacterial pneumonia, and normal/healthy subjects. Researchers further studied X-ray images of viral pneumonia to detect the presence

of COVID-19. The second-stage model for identifying the fact of COVID-19 shows exceptional performance. This model is reliable, accurate, fast, and requires fewer computational requirements to distinguish the existence of pneumonia made by the COVID-19 virus from pneumonia caused by the virus so that the appropriate treatment can be administered. In the current scenario, parallel testing can be used to prevent the spread of infection to frontline workers and provide an early diagnosis to understand whether a patient has been affected by COVID-19 or not. Therefore, while influencing management, the proposed method can be used as an alternative diagnostic tool with potential candidacy in diagnosing COVID-19 cases. Finally, the present work shows that it may be possible to use DL models for detecting COVID-19, as all current researches show promising outcomes. The high accuracies gained by a number of methods show that DL models find something in images and that something makes deep networks able to recognise images correctly—research on whether the results of DL methods constitute reliable.

6.13 | Other applications

DL is used in almost every field nowadays. Many other DL applications include the automatic generation of image captions, text generation, and playing the game automatically.

7 | CONCLUSION AND FUTURE WORK

We can find new patterns and find new abstract knowledge with DL. If we want to choose DL methods, we can consider the following procedure:

If the problem is about classification, it might be better to use DBNs; if the problem is about finding a new pattern accompanied by less unlabelled data, it would be good to use pure RBM and Autoencoders. From another point of view, if we want to classify more suitable tools based on their application. In this survey, we also focussed on the application of DL in several domains such as transportation systems, medical image analysis, and so on. After this survey, we have found that for Text processing, NLP, and Sentiment analysis, Recurrent Networks or DBN have shown good performances. For medical image processing, the best ones are DBN and CNN. One of the best choices for speech recognition and time series problems (pattern will be changed over time) is Recurrent Networks (RNN). For dimensionality reduction, two of the best tools are RBMs and Auto-Encoders. A plan for future work is DL researchers can develop a new framework that covers the pros of old frameworks and ignore the cons. Besides, in future works, people can use DL algorithms to predict natural disasters to keep themselves safe.

CONFLICT OF INTEREST

The authors declared that they have no conflicts of interest to this work.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analysed in this study.

ORCID

Mehdi Gheisari  <https://orcid.org/0000-0002-5643-0021>

Yang Liu  <https://orcid.org/0000-0003-2486-5765>

Abolfazl Mehbodniya  <https://orcid.org/0000-0002-0945-512X>

Mustafa Ghaderzadeh  <https://orcid.org/0000-0003-4016-3843>

REFERENCES

- Boyd, D., Crawford, K.: Critical questions for big data: provocations for a cultural, technological, and scholarly phenomenon. *Inf. Commun. Soc.* 15(5), 662–679 (2012). <https://doi.org/10.1080/1369118x.2012.678878>
- Webb, G.I., Pazzani, M.J., Billsus, D.: Machine learning for user modeling. *User Model. User-Adapted Interact.* 11(1), 19–29 (2001). <https://doi.org/10.1023/a:1011117102175>
- Liu, Y., Starzyk, J.A., Zhu, Z.: Optimized approximation algorithm in neural networks without overfitting. *IEEE Trans. Neural Netw.* 19(6), 983–995 (2008). <https://doi.org/10.1109/tnn.2007.915114>
- Gheisari, M., Wang, G., Bhuiyan, M.Z.A.: A survey on deep learning in big data. In: *IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 173–180. IEEE (2017)
- Srivastava, N., et al.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1), 1929–1958 (2014)
- Hinton, G.E., Simon, O., Teh, Y.-W.: A fast learning algorithm for deep belief nets. *Neural Comput.* 18(7), 1527–1554 (2006). <https://doi.org/10.1162/neco.2006.18.7.1527>
- Motahari Kia, M.M., et al.: A novel method for recognition of Persian alphabet by using fuzzy neural network. *IEEE Access* 6, 77265–77271 (2018)
- Dreiseitl, S., Ohno-Machado, L.: Logistic regression and artificial neural network classification models: a methodology review. *J. Biomed. Inf.* 35(5–6), 352–359 (2002). [https://doi.org/10.1016/s1532-0464\(03\)00034-0](https://doi.org/10.1016/s1532-0464(03)00034-0)
- Chauhan, S., et al.: A comparison of shallow and deep learning methods for predicting cognitive performance of stroke patients from MRI lesion images. *Front. Neuroinf.* 13, 53 (2019). <https://doi.org/10.3389/fninf.2019.00053>
- Schmidhuber, J.: Deep learning in neural networks: an overview. *Neural Netw.* 61, 85–117 (2015). <https://doi.org/10.1016/j.neunet.2014.09.003>
- Deng, L., Dong, Y.: Deep learning: methods and applications. *Found. Trends Signal Process.* 7(3–4), 197–387 (2014). <https://doi.org/10.1561/20000000039>
- Warde-Farley, D., et al.: An empirical analysis of dropout in piecewise linear networks. *arXiv preprint arXiv:1312.6197* (2013)
- Ghaderzadeh, M., et al.: A fast and efficient CNN model for B-ALL diagnosis and its subtypes classification using peripheral blood smear images. *Int. J. Intell. Syst.* (n.d.)
- Ghaderzadeh, M., et al.: Deep convolutional neural network-based computer-aided detection system for covid-19 using multiple lung scans: design and implementation study. *J. Med. Internet Res.* 23, e27468 (2021). <https://doi.org/10.2196/27468>
- Ghaderzadeh, M., Aria, M., Asadi, F.: X-ray equipped with artificial intelligence: changing the COVID-19 diagnostic paradigm during the pandemic. *Biomed Res. Int.* 2021, 9942873 (2021). <https://doi.org/10.1155/2021/9942873>
- Mishkin, D., Sergievskiy, N., Matas, J.: Systematic evaluation of convolution neural network advances on the imageNet. *Comput. Vis.*

- Image Understand. 161, 11–19 (2017). <https://doi.org/10.1016/j.cviu.2017.05.007>
17. Hinton, G.E., Sejnowski, T.J.: Chapter learning and relearning in Boltzmann machines. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, pp. 282–317. MIT Press, Cambridge (1986)
 18. Alzubaidi, L., et al.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* 8(1), 1–74 (2021). <https://doi.org/10.1186/s40537-021-00444-8>
 19. Montavon, G., Müller, K.-R.: Deep Boltzmann machines and the centering trick. In: *Neural Networks: Tricks of the Trade*, pp. 621–637. Springer, Berlin (2012)
 20. Guo, Y., Guo, Y.: Deep learning for visual understanding. (2017)
 21. Yang, J., Zhang, X., Wang, X.: Multi-see: a shape model based on deep learning method. *Procedia Comput. Sci.* 129, 375–381 (2018). <https://doi.org/10.1016/j.procs.2018.03.093>
 22. Salakhutdinov, R., Hinton, G.E.: Deep Boltzmann machines. In: *AISTATS*, vol. 1 (2009)
 23. Adams, W.A.: *Analysis of Robustness in Lane Detection Using Machine Learning Models*. Diss. Ohio University (2015)
 24. Hung, C.-W., et al.: Design of a chamfering tool diagnosis system using autoencoder learning method. *Energies* 12(19), 3708 (2019). <https://doi.org/10.3390/en12193708>
 25. Cao, T., et al. In: *Advances in Knowledge Discovery and Data Mining: 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part I*. (2015)
 26. Tjandra, A., et al.: Gated recurrent neural tensor network. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE (2016)
 27. Pouransari, H., Ghili, S.: Deep Learning for Sentiment Analysis of Movie Reviews. *CS224N Proj*, pp. 1–8 (2014)
 28. Graves, A., Mohamed, A.-R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE (2013)
 29. Kaur, M., Singh, D.: Fusion of medical images using deep belief networks. *Cluster Comput.* 23(2), 1439–1453 (2020). <https://doi.org/10.1007/s10586-019-02999-x>
 30. Yu, D., Deng, L.: Deep learning and its applications to signal and information processing [exploratory dsp]. *IEEE Signal Process. Mag.* 28(1), 145–154 (2010). <https://doi.org/10.1109/msp.2010.939038>
 31. Liu, W., et al.: A survey of deep neural network architectures and their applications. *Neurocomputing* 234, 11–26 (2017). <https://doi.org/10.1016/j.neucom.2016.12.038>
 32. Kamalov, F., et al.: Feature selection for intrusion detection systems. In: *2020 13th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, pp. 265–269 (2020). <https://doi.org/10.1109/ISCID51228.2020.00065>
 33. Arel, I., Rose, D.C., Karnowski, T.P.: Deep machine learning—a new frontier in artificial intelligence research [research frontier]. *IEEE Comput. Intell. Mag.* 5(4), 13–18 (2010). <https://doi.org/10.1109/mci.2010.938364>
 34. <https://www.tutorialspoint.com/what-are-the-differences-between-bidg-and-caffe>
 35. Bakker, B.: Reinforcement learning by backpropagation through an LSTM model/critic. In: *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*. IEEE (2007)
 36. Tao, W.: <https://software.intel.com/en-us/articles/bidg-distributed-deep-learning-on-apache-spark.intel.2017>
 37. <https://deeplearning4j.org/neuralnetworktable.online.2017>. URL: <https://deeplearning4j.org/neuralnetworktable>
 38. Tai, L., Liu, M.: Deep-learning in mobile robotics—from perception to control systems: a survey on why and why not. *arXiv preprint arXiv:1612.07139* 1 (2016)
 39. Al-Rfou, R., et al.: Theano: a Python framework for fast computation of mathematical expressions. *arXiv e-prints* (2016), *arXiv:1605*
 40. Xu, L., et al.: Performance analysis and prediction for mobile internet-of-things (IoT) networks: a CNN approach. *IEEE Internet Things J.* 8(17), 13355–13366 (2021). <https://doi.org/10.1109/jiot.2021.3065368>
 41. Zeng, C., Wang, Z., Wang, Z.: Image reconstruction of IoT based on parallel CNN. In: *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (Green-Com) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. IEEE (2020)
 42. Deepthi, S.A., Rao, E.S., Giriprasad, M.N.: Secure MRI brain image transmission using IOT devices based on hybrid autoencoder and restricted Boltzmann approach. *J. Sens.* 2022, 1–11 (2022). <https://doi.org/10.1155/2022/5841630>
 43. Huang, G., Wang, H., Zhang, L.: Sparse-coding-based autoencoder and its application for cancer survivability prediction. *Math. Probl. Eng.* 2022, 1–12 (2022). <https://doi.org/10.1155/2022/8544122>
 44. Deshmukh, M.S., Ravikesh Bhaladhare, P.: Intrusion detection system (DBN-IDS) for IoT using optimization enabled deep belief neural network. In: *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*. IEEE (2021)
 45. Pazhoohesh, M., et al.: Dealing with missing data in the smart buildings using innovative imputation techniques. In: *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–7 (2021). <https://doi.org/10.1109/IECON48115.2021.9612650>
 46. Zhang, X., Zhao, M., Dong, R.: *Appl. Sci.* 10(3), 1144 (2020). <https://doi.org/10.3390/app10031144>
 47. Al-Jamali, N.A.S., Al-Rawashidy, H.S.: Smart IoT network based convolutional recurrent neural network with element-wise prediction system. *IEEE Access* 9, 47864–47874 (2021). (inja yadet nare favayed bodan machine learning kenar deeo). <https://doi.org/10.1109/access.2021.3068610>
 48. Dey, P., Chaulya, S.K., Kumar, S.: Hybrid CNN-LSTM and IoT-based coal mine hazards monitoring and prediction system. *Process Saf. Environ. Protect.* 152(2021), 249–263 (2021). <https://doi.org/10.1016/j.psep.2021.06.005>
 49. Xu, J., He, Z., Zhang, Y.: CNN-LSTM combined network for IoT enabled fall detection applications. *J. Phys. Conf.* 1267(1), 012044 (2019). <https://doi.org/10.1088/1742-6596/1267/1/012044>
 50. Bahrapour, S., et al.: Comparative study of caffe, neon, theano, and torch for deep learning. *arXiv preprint arXiv:1511.06435* (2015), 132
 51. Boan, J., Dominish, D., Brown, G.: Service oriented architecture and distributed computing using LASAGNE. In: *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference* (2015)
 52. Shvachko, K., et al.: The Hadoop distributed file system. In: *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSSST)*. IEEE (2010)
 53. Van Merriënboer, B., et al.: Blocks and fuel: frameworks for deep learning. *arXiv preprint arXiv:1506.00619* (2015)
 54. <https://blog.karatos.in/?ID=01500-0a0840a6-3d42-4118-8a34-39a595539770>
 55. Google: Tensorflow 1.0. <https://developers.googleblog.com/2017/02/announcingtensorflow-10.html?m=102> (2017)
 56. Chen, T., et al.: MXNet: a flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274* (2015)
 57. <https://techvidvan.com/tutorials/advantages-drawbacks-of-keras/>
 58. Amazon Team. <https://aws.amazon.com/rekognition/> (2016). November
 59. Microsoft: CNTK. <https://www.microsoft.com/en-us/research/product/cognitive-toolkit/> (2016)
 60. Seide, F., Amit, A.: CNTK: Microsoft's open-source deep-learning toolkit. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016)
 61. Jia, Y., et al.: Caffe: convolutional architecture for fast feature embedding. In: *Proceedings of the 22nd ACM International Conference on Multimedia* (2014)

62. https://piazza.com/class_profile/get_resource/i48o74a0lqu0/i5scmqklbuy326
63. Meng, X., et al.: MLlib: machine learning in Apache spark. *J. Mach. Learn. Res.* 17(1), 1235–1241 (2016)
64. <https://www.freecodecamp.org/news/deep-learning-frameworks-compared-mxnet-vs-tensorflow-vs-dl4j-vs-pytorch/>
65. Jia, Y.: Caffe model zoo (2015)
66. Moritz, P., et al.: SparkNET: training deep networks in spark. arXiv preprint arXiv:1511.06051 (2015)
67. NVIDIA DIGITS—Interactive Deep Learning GPU. Training System (2015)
68. <https://www.knowledgehut.com/blog/big-data/apache-spark-advantages-disadvantages>
69. Intel: BigDL. <https://software.intel.com/en-us/articles/bigdl-distributed-deep-learning-on-apache-spark> (2017)
70. <https://modern Catalysts.com/pros-and-cons-of-nvidia-and-amd-graphics-cards/>
71. URL 1: https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/
72. Tokui, S., et al.: Chainer: a next-generation open source framework for deep learning. In: Proceedings of Workshop on Machine Learning Systems (LearningSys) in the Twenty-Ninth Annual Conference on Neural Information Processing Systems (NIPS), vol. 5 (2015)
73. <https://machinelearningmastery.com/inspirational-applications-deep-learning/>
74. Parham, J.R., et al.: An animal detection pipeline for identification. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1075–1083 (2018)
75. Tang, Z., et al.: Deep learning of imaging phenotype and genotype for predicting overall survival time of glioblastoma patients. *IEEE Trans. Med. Imag.* 39(6), 2100–2109 (2020). <https://doi.org/10.1109/tmi.2020.2964310>
76. Sun, L., et al.: Anatomical attention guided deep networks for ROI segmentation of brain MR images. *IEEE Trans. Med. Imag.* 39(6), 2000–2012 (2019). <https://doi.org/10.1109/tmi.2019.2962792>
77. Liu, M., et al.: Landmark-based deep multi-instance learning for brain disease diagnosis. *Med. Image Anal.* 43, 157–168 (2018). <https://doi.org/10.1016/j.media.2017.10.005>
78. Chen, Y., et al.: Voxel deconvolutional networks for 3D brain image labeling. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018)
79. Lian, C., et al.: Multi-channel multi-scale fully convolutional network for 3D perivascular spaces segmentation in 7T MR images. *Med. Image Anal.* 46, 106–117 (2018). <https://doi.org/10.1016/j.media.2018.02.009>
80. Ghaderzadeh, M., Asadi, F.: Deep learning in the detection and diagnosis of COVID-19 using radiology modalities: a systematic review. *J. Healthc. Eng.* 6677314 (2021). <https://doi.org/10.1155/2021/6677314>
81. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2012)
82. Ackerman, E., Pagano, A.: Deep-learning first: drive.ai's path to autonomous driving. *IEEE Spectr.* Available online: <https://spectrum.ieee.org/video/transportation/self-driving/driveai-deep-learning-first-autonomous-driving> (2017). Accessed 26 Feb 2019
83. Yu, J., et al.: A vision-based robotic grasping system using deep learning for 3D object recognition and pose estimation. In: 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE (2013)
84. Yogesh, K., et al.: Heart failure detection using quantum-enhanced machine learning and traditional machine learning techniques for internet of artificially intelligent medical things. *Wirel. Commun. Mob. Comput.* 2021, 1616725, 1–16 (2021)
85. Leng, B., et al.: 3D object classification using deep belief networks. In: International Conference on Multimedia Modeling. Springer, Cham (2014)
86. Leng, B., et al.: A 3D model recognition mechanism based on deep Boltzmann machines. *Neurocomputing* 151, 593–602 (2015). <https://doi.org/10.1016/j.neucom.2014.06.084>
87. Otter, D.W., Medina, J.R., Kalita, J.K.: A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* 32(2), 604–624 (2020). <https://doi.org/10.1109/tnnls.2020.2979670>
88. Zheng, L., Wang, H., Gao, S.: Sentimental feature selection for sentiment analysis of Chinese online reviews. *Int. J. Mach. Learn. Cybern.* 9(1), 75–84 (2018). <https://doi.org/10.1007/s13042-015-0347-4>
89. Etter, M., et al.: Measuring organizational legitimacy in social media: assessing citizens' judgments with sentiment analysis. *Bus. Soc.* 57(1), 60–97 (2018). <https://doi.org/10.1177/0007650316683926>
90. Cliche, M.: BB_twttr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs. arXiv preprint arXiv:1704.06125 (2017)
91. Gómez Sánchez, G.: Voice Generation Using Deep Learning. BS thesis. Universitat Politècnica de Catalunya (2016)
92. Arık, S.Ö., et al.: Deep voice: real-time neural text-to-speech. In: International Conference on Machine Learning. PMLR (2017)
93. van den Oord, A., et al.: WaveNet: a generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)
94. Sotelo, J., et al.: Char2Wav: End-to-End Speech Synthesis (2017)
95. Yadav, S.S., Jadhav, S.M.: Deep convolutional neural network based medical image classification for disease diagnosis. *J. Big Data* 6(1), 1–18 (2019). <https://doi.org/10.1186/s40537-019-0276-2>
96. Liu, X., et al.: Medical image classification based on an adaptive size deep learning model. *ACM Trans. Multimed. Comput. Commun. Appl.* 17(3s), 1–18 (2021). <https://doi.org/10.1145/3465220>
97. Ghaderzadeh, M., et al.: Efficient framework for detection of COVID-19 omicron and delta variants based on two intelligent phases of CNN models. *Comput. Math. Methods Med.* 4838009 (2022). <https://doi.org/10.1155/2022/4838009>
98. Neelapu, R., Devi, G.L., Srinivasa Rao, K.: Deep learning based conventional neural network architecture for medical image classification. *Trait. du Signal.* 35(2), 169–182 (2018). <https://doi.org/10.3166/ts.35.169-182>
99. Kowsari, K., et al.: HMIC: hierarchical medical image classification, a deep learning approach. *Information* 11(6), 318 (2020). <https://doi.org/10.3390/info11060318>
100. Rezaeilouyeh, H., Ali, M., Mahoor, M.H.: Microscopic medical image classification framework via deep learning and shearlet transform. *J. Med. Imaging* 3(4), 044501 (2016). <https://doi.org/10.1117/1.jmi.3.4.044501>
101. Kayalibay, B., Jensen, G., van der Smagt, P.: CNN-based segmentation of medical imaging data. arXiv preprint arXiv:1701.03056 (2017)
102. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham (2015)
103. Lai, Z.F., Deng, H.F.: Medical image classification based on deep features extracted by deep model and statistic feature fusion with multilayer perceptron. *Comput. Intell. Neurosci.* 2018, 1–13 (2018). <https://doi.org/10.1155/2018/2061516>
104. Vijayalakshmi, A., Rajesh Kanna, B.: Deep learning approach to detect malaria from microscopic images. *Multimed. Tool. Appl.* 79(21), 15297–15317 (2020). <https://doi.org/10.1007/s11042-019-7162-y>
105. Rao, Y., Ni, J.: A deep learning approach to detection of splicing and copy-move forgeries in images. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE (2016)
106. Jain, G., et al.: A deep learning approach to detect Covid-19 coronavirus with X-ray images. *Biocybern. Biomed. Eng.* 40(4), 1391–1405 (2020). <https://doi.org/10.1016/j.bbe.2020.08.008>

How to cite this article: Gheisari, M., et al.: Deep learning: applications, architectures, models, tools, and frameworks: a comprehensive survey. *CAAI Trans. Intell. Technol.* 8(3), 581–606 (2023). <https://doi.org/10.1049/cit2.12180>