A Survey on State-of-the-art Deep Learning Applications and Challenges

Mohd Halim Mohd Noor[a,*], Ayokunle Olalekan Ige[b]

[a]*School of Computer Sciences, Universiti Sains Malaysia, Jalan Universiti, Gelugor, 11800, Pulau Pinang, Malaysia*
[b]*Department of Computer Science, Adekunle Ajasin University, Ikare Akoko-Ipeme Rd, Akungba-Akoko, P.M.B 001, Ondo State, Nigeria*

Abstract

Deep learning, a branch of artificial intelligence, is a data-driven method that uses multiple layers of interconnected units or neurons to learn intricate patterns and representations directly from raw input data. Empowered by this learning capability, it has become a powerful tool for solving complex problems and is the core driver of many groundbreaking technologies and innovations. Building a deep learning model is challenging due to the algorithm's complexity and the dynamic nature of real-world problems. Several studies have reviewed deep learning concepts and applications. However, the studies mostly focused on the types of deep learning models and convolutional neural network architectures, offering limited coverage of the state-of-the-art deep learning models and their applications in solving complex problems across different domains. Therefore, motivated by the limitations, this study aims to comprehensively review the state-of-the-art deep learning models in computer vision, natural language processing, time series analysis and pervasive computing, and robotics. We highlight the key features of the models and their effectiveness in solving the problems within each domain. Furthermore, this study presents the fundamentals of deep learning, various deep learning model types and prominent convolutional neural network architectures. Finally, challenges and future directions in deep learning research are discussed to offer a broader perspective for future researchers.

*Corresponding author

Email addresses: `halimnoor@usm.my` (Mohd Halim Mohd Noor), `ayo.ige@aaua.edu.ng` (Ayokunle Olalekan Ige)

1. Introduction

Deep learning has revolutionized many applications across a variety of industries and research. The application of deep learning can be found in healthcare [1], smart manufacturing [2], robotics [3] and cybersecurity [4], solving challenging and complex problems such as disease diagnosis, anomaly detection, object detection and malware attack detection. Deep learning is a subset of machine learning that focuses on learning from data using artificial neural networks with many layers, known as deep neural networks. An artificial neural network is a computational model that imitates the working principles of a human brain. The computational models are composed of an input layer which receives the input data, multiple processing layers that learn the representation of data and the output layer which produces the output of the model.

Prior to the reintroduction of deep learning (DL) into the research trend, pattern recognition tasks involved a transformation of the raw input data such as pixel values of an image into a feature vector that represents the internal representation of the data. The feature vector can be used by a machine learning model to detect or classify patterns in the data. This process requires feature engineering and considerable domain knowledge to design a suitable feature representation. With deep learning, this cumbersome process can be performed automatically whereby at each processing layer known as hidden layers, the internal representation of the input data is learned or extracted in a hierarchical manner. The first layer learns the presence of basic primitive features such as edges, dots, and lines. The second layer learns patterns or motifs by recognizing the combinations of the edges, dots and lines, and the subsequent layers combine the motifs to produce more sophisticated features that correspond to the input data. This feature learning process takes place in the sequence of hidden layers until the prediction is finally produced.

Deep learning has seen numerous breakthroughs in various industries, transforming how complex problems are solved and how businesses operate. For instance, protein folding is a complex process, influenced by the intricate

sequence of amino acids. Using traditional methods to determine how a protein folds would take an immense amount of computational power and time due to its complex 3D structure and vast number of possible configurations. AlphaFold [5] has made significant progress in this area, using deep learning to predict protein structures with remarkable accuracy, helping researchers to comprehend the complete structure of a key protein associated with diseases like malaria and Parkinson's disease. In natural language processing, systems like OpenAI GPT, Google Gemini and IBM Watsonx have revolutionized chatbots and virtual assistants, enabling them to understand and respond to human language with remarkable accuracy and contextual awareness. These systems have overcome the challenge of processing vast amounts of unstructured text, allowing them to engage in more natural conversations and handle a wide range of topics. Businesses can use these systems to automate customer service, provide real-time support and improve user experience.

Several studies have been conducted to discuss the concept and application of deep learning in the last few years, as listed in Table 1. The studies addressed or focused on several aspects of deep learning, such as types of deep learning models, learning approaches and strategies, convolutional neural network (CNN) architectures, deep learning applications and challenges. In [6], the authors provided fundamentals of deep learning and highlighted different types of deep learning models, such as convolutional neural networks, autoencoder and generative adversarial networks. Then, the applications of deep learning in various domains are discussed, and some challenges associated with deep learning applications are presented. Another survey [7] provided a comprehensive analysis of supervised, unsupervised and reinforcement learning approaches and compared the different learning strategies such as online, federated and transfer learning. Finally, the current challenges of deep learning and future direction are discussed.

In [8], the authors provided a comprehensive review of the popular CNN architectures used in computer vision tasks, highlighting their key features and advantages. Then, the applications of deep learning in medical imaging and the challenges are discussed. A similar survey is reported in [9], where the different supervised and unsupervised deep learning models are highlighted, and the popular CNN architectures are compared and discussed. In another survey [10], the authors focused on the applications of deep learning in computer vision, natural language processing and speech and audio processing. The different types of deep learning models are also discussed. In

[11], the authors focused on the different types of deep learning models and provided a summary of deep learning applications in various domains.

Despite the existing surveys on deep learning that offer valuable insights, the increasing amount of deep learning applications and the existing limitations in the current studies motivated us to explore this topic in depth. In general, to the best of our knowledge, no survey paper focuses on the emerging trends in state-of-the-art applications and the current challenges associated with deep learning. Furthermore, the surveys do not discuss the issues and how deep learning addresses them by highlighting the key features and components in the models. Furthermore, most surveys either ignore or provide minimal coverage of the fundamentals of deep learning, which is crucial for understanding the state-of-the-art models. The main objective of this paper is to present the most important aspects of deep learning, making it accessible to a wide audience and facilitating researchers and practitioners in advancing and leveraging its capabilities to solve complex problems across diverse domains. Specifically, we present the fundamentals of deep learning and the various types of deep learning models, including popular deep learning architectures. Then, we discuss the progress of deep learning in state-of-the-art applications, highlighting the key features of the models and their problem-solving approaches. Finally, we discuss the challenges faced by deep learning and the future research directions.

To this end, we conducted automatic search strategies using "deep learning" keyword with keywords related to deep learning applications such as "image classification", "neural machine translation", "text generation", "human activity recognition" and "robotics". Then, a manual search was carried out by scanning the references produced by the automatic search, selecting relevant works and discarding the irrelevant ones. This survey paper collected primary studies from journals, conference proceedings, and books in English only. We use scientific databases such as IEEE Explore, ScienceDirect, SpringerLink, ACM Digital Library, Scopus and ArXiv. The ArXiv repository is used because it hosts numerous manuscripts that are highly relevant to the topics discussed in this survey. These preprints often contain novel findings and methodologies that may not be available in other repositories at the time of writing. Nevertheless, we ensure that the selected preprints are of high quality, authored by researchers from reputable institutions.

The remainder of this paper is organized as follows: Section 2 describes

Table 1: Summary of related works.

Ref.	Focus	Concepts not covered
[6]	A short review of the fundamentals of DL networks and discusses different types of neural networks, DL applications and challenges.	Lack of analysis of CNN architectures and limited coverage of deep learning fundamentals.
[7]	Discusses the learning approaches (supervised, unsupervised and reinforcement learnings), learning strategies, and DL challenges	Lack of fundamentals of deep learning, CNN architectures and DL applications.
[8]	Discusses different types of DL networks, CNN fundamentals and architectures, DL challenges and medical imaging applications	Limited discussion on DL applications such as natural language processing and time series analysis.
[9]	A short review of the fundamentals of neural networks and discusses different types of DL networks, CNN architectures and applications.	Limited discussion on DL applications and no discussion of DL challenges.
[10]	Discusses different types of DL networks and DL applications and challenges	Lack of analysis of CNN architectures and limited coverage of deep learning fundamentals.
[11]	Discusses different types of DL networks and provides a summary of DL applications	Lack of fundamentals of deep learning, analysis of CNN architectures and limited discussion on DL applications.

the fundamentals of deep learning which includes layers and attention mechanisms, activation functions, model optimization and loss functions, and regularization methods. Section 3 presents the types of deep learning models, including the CNN architectures. Section 4 discusses the state-of-the-art applications of deep learning. Section 5 discusses the challenges and future directions in the field of deep learning. The conclusion is given in Section 6.

2. Fundamentals of Deep Learning

This section describes the fundamental concepts such as layer types, activation functions, training algorithms and regularization methods to provide a comprehensive understanding of the underlying principles in advancing the field of deep learning.

2.1. Layers

A deep learning model is characterized by having numerous hidden layers. The hidden layers are responsible for learning and extracting complex features from the input data. A hidden layer is composed of an arbitrary number of neurons which serves as the fundamental building block of a neural network as shown in Fig. 1. A neuron consists of an arbitrary number of inputs, each associated with a weight, which controls the flow of information into the neuron during the forward pass. The flow of information, or forward pass, involves the computation of summation of the weighted input, followed by the application of a transformation function to the weighted sum. Let $\mathbf{x} = x_1, x_2, ..., x_d, j = 0, 1, ..., d$ be the input vector with d dimensions and $\mathbf{w}_i^l$ denotes the weights that are connected to neuron i in layer l. The forward pass to neuron i in layer l is defined as

$$z_i^l = \mathbf{w}_i^l \cdot \mathbf{x} = \sum_{j=0}^{d} w_{i,j}^l \cdot x_j \tag{1}$$

$$a_i^l = g(z_i^l) \tag{2}$$

where x_j is the input vector of size d, $w_{i,j}^l$ is the weight associated with x_j, connecting the input to neuron i at layer l, and g is the transformation function also known as activation function. It is worth noting that w_0 is called bias and $x_0 = 1$. A hidden layer wherein each neuron is connected to all neurons of the previous layer is known as the fully connected layer. The

forward pass computation of a neuron a_i^l in layer l receiving a set of input from layer $l-1$ can be generalized as follows:

$$a_i^l = g(\sum_{j=0}^{d} w_{i,j}^l \cdot a_j^{l-1}) = \mathbf{w}_i^l \cdot \mathbf{a}^{l-1} \tag{3}$$

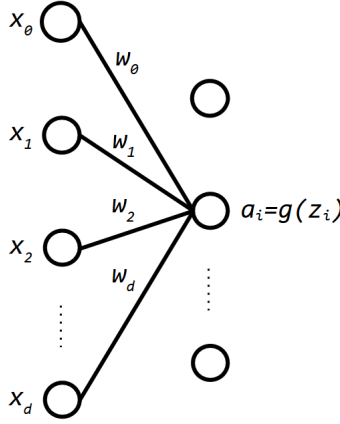where $\mathbf{a}^{l-1}$ is the input vector from layer $l-1$.

Figure 1: A graphical representation of a neuron.

Another crucial layer type is the convolutional layer, which is primarily used for processing data that has underlying structures, such as spatial patterns in image data or temporal patterns in time series data. Unlike a fully connected layer, each neuron in a convolutional layer is connected only to a subset of neurons in the previous layer, as shown in Fig. 2. As shown in the figure, each neuron in the hidden layer is connected only to a local region, a subset of nine input neurons, and the weights are shared across the input data. The weight-sharing not only significantly reduces the number of parameters of the neural network, but also allows the network to learn the same features across different spatial locations in the input [12]. Let $\mathbf{x}$ be a two-dimensional input data with $H \times W \times C$ such as a grayscale image, where H is the height, W is the width and C is the channel of the image data. The computation of a neuron in the convolutional layer l is defined as

$$z_{i,j,d}^l = \sum_{m=0}^{k_1} \sum_{n=0}^{k_2} w_{m,n,d}^l \cdot x_{i \cdot s+m, j \cdot s+n, d} \tag{4}$$

7

$$a^l_{i,j,d} = g(z^l_{i,j,d}) \tag{5}$$

where $w_{m,n,d}$ is the weight connecting the input data $x_{i,j,d}$ within the window to the neuron in layer l and s is the stride or the step size of the window as it moves across the input data. This convolution operation where the filter is represented by $w_{m,n,d}$ slides over the input data $x_{i,j,d}$, producing a set of output values called feature map. For a more hands-on understanding of this convolutional operation, readers are encouraged to explore CNN Explainer [13], an interactive tool that demonstrates how convolutional layers process input data and generate feature maps.
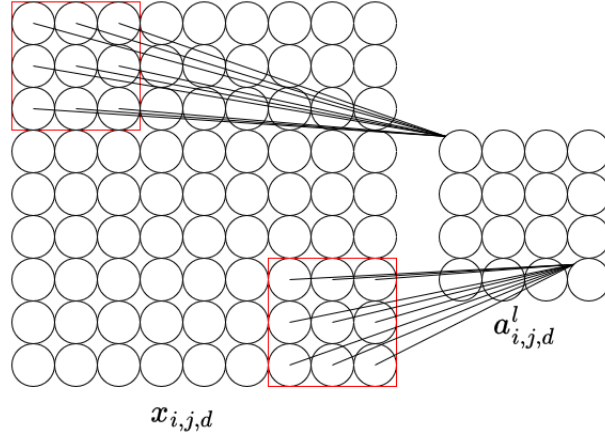

Figure 2: A neuron is connected to a local region of the input data.

Pooling layers are commonly applied after successive convolutional layers to progressively reduce the spatial dimensions of the feature maps. The spatial reduction is performed by computing the summary of a subset of the feature values in the feature maps as shown in Fig. 3. The pooling operation, as shown in the figure, can use either the maximum or average method with a pooling size of 2×2 applied across the entire feature map, thereby reducing the size of the feature map. In addition to spatial reduction, pooling layers decrease the number of parameters and provide translation-invariant features [12]. Let $\mathbf{a}^l$ denotes feature map (hidden layer) l, a pooling operation with a pooling size of $m \times n$ is defined as

$$\mathbf{a}^{l+1} = \texttt{pool}(\mathbf{a}^l_{i \cdot s+m, j \cdot s+n}) \tag{6}$$

where `pool` is either maximum or average function and s is the stride or step size of the window as it moves across the feature map.
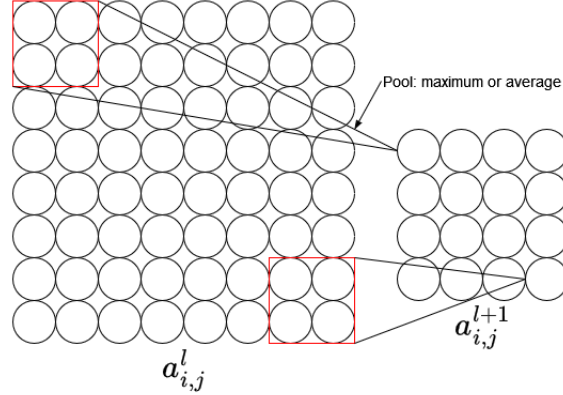

Figure 3: Summaries of the feature map using maximum or average pooling to produce a reduced feature map.

2.2. Attention Mechanisms

One of the important concepts in pattern recognition is the ability to attend to and neglect certain parts of the input data based on their relevance. This is because not all parts of the input hold equal importance for making predictions. Certain features exhibit a stronger correlation with the output while others are less relevant. To provide a simple analogy, consider a set of sensors to measure room temperature. The sensors are deployed across different corners of the room. Each sensor measures temperature in its own area, and these local measurements are used to estimate the room temperature. However, not all sensors are equally important due to various factors. For instance, a sensor near an air-conditioning or a heater might give inaccurate readings due to the external temperature fluctuation caused by these systems. A sensor near the center of the room, away from the heating or cooling sources might provide more reliable readings. Therefore, when estimating the room temperature, we give more attention to the sensors that are less affected by the localized factors. In convolutional layers, all extracted features are treated uniformly, without consideration of the varying degree of the importance of the different parts of the input data. This limitation is addressed by the introduction of attention mechanism, which can dynamically assign varying levels of significance (weights) to the different features.

This flexibility enables the deep learning models to prioritize the more relevant aspects of the input data, enhancing its ability to capture the intricate dependencies for accurate prediction. Given an input data x, the process of attending to the important components of the input is given as

$$A = f(g(x), x) \tag{7}$$

where g is a composite function that performs a sequence of operations to generate the attention or the weights and f applies the generated attention $g(x)$ on the input x.

For instance, the squeeze-and-excitation (SE) attention generates the attention through five consecutive operations [14]. First, the input is vectorized using global average pooling. Then the vector is passed to two fully connected layers, where the first one with ReLU activation and the second one with sigmoid activation. SE attention was a pioneer in channel attention. The attention module assigns varying weights to the channels of the feature maps. SE attention suffers from computational cost and the use of global average, which may cause information loss at the spatial level. Several efforts have been made to improve SE attention. Global Second-order Pooling (GSoP) attention performs 1×1 convolution on the feature maps to reduce the number of channels, and then computes the pairwise channel correlation, which is used to generate the weights [15]. Efficient Channel Attention (ECA) replaced the fully connected layers with 1D convolution to reduce the number of parameters and the computational cost [16].

Temporal attention is an attention module that focuses on specific time steps in a sequence of data such as time series and video (sequence of images). In video processing such as recognizing human actions, temporal attention is used to focus on key frames at different point in time that contain crucial information for predicting the ongoing activity. Temporal adaptive module (TAM) is a temporal attention that can focus on short-term (local) information and global context information of the data [17]. The composite function consists of a local branch for generating attention weights and a global branch for generating a channel-wise adaptive kernel. First, the input feature map is squeezed using global average pooling to reduce the computational cost. Subsequently, the local branch executes two 1D convolution operations, with the first convolution using ReLU, and sigmoid activation for the second to generate the local weights. The local weights are then multiplied with the feature map. Meanwhile, the global branch is composed of two fully connected lay-

ers, with the first layer using ReLU and the second layer employing softmax function to generate the adaptive kernel (weights). Self-attention is a form of temporal attention, initially proposed for machine translation to enable the deep learning models to attend different words in a sequence relative to other words [18]. The attention module has become a fundamental building block in various natural language processing applications. To generate the attention weights, the input (word embeddings) is transformed by linear projection to compute query, key, and value. Then, the dot product between query and key is computed, and the resultant is normalized by the square root of the size of the key. Finally, the attention weights are obtained by applying the softmax function. Self-attention is the fundamental building block of the Transformer architecture, a key deep learning model in natural language processing. For a more hand-on understanding of self-attention and Transformer, readers are encouraged to explore Transformer Explainer [19], an interactive tool that demonstrates how self-attention mechanisms work in Transformer by visualizing the attention scores and how different inputs interact with each other.

Spatial attention focuses on specific regions or spatial location of the input data, enabling the deep learning models to selectively emphasize and ignore certain features. In the context of computer vision, spatial attention is crucial in capturing the spatial relationships and context within an image for accurate prediction. Attention gate is a spatial attention that can identify and focus the salient regions and suppress feature responses of the insignificant ones. The composite function consists of ReLU activation followed by 1×1 convolution to reduce channel dimension of the feature maps to a singular feature map. Finally, sigmoid is applied to the feature map to generate attention weights [20]. The self-attention in the standard Transformer is not effective in handling image data due to its inherent sequential processing nature and lacks the ability to capture spatial dependencies and local patterns. To address this limitation, the Vision Transformer (ViT) treats images as a sequence of non-overlapping patches. A similar computational pipeline is used to generate the attention weights, the sequence of patches is transformed by linear projection to compute the query, key, and value [21]. The same operations are employed to generate the attention weights. Self-attention is computationally costly due to its quadratic complexity, especially when dealing with image data. To reduce the complexity, two learnable linear layers, independent of the input data, are adopted as the key and value vectors [22].

2.3. Activation Functions

The role of the activation function is to transform the weighted sum into a more classifiable form. This is crucial to the learning behavior of the deep learning model, generating non-linear relationships between the input and the output of the model. The activation function, combined with many hidden layers, allows the neural network to approximate highly complex, non-linear functions. Many activation functions are available for use in neural networks, and some of the functions are shown in Fig. 4—Fig 6. The figures show the plots of the three popular activation functions. The sigmoid is a classic example of an activation function used in logistic regression. It maps the weighted sum to a value in the range of 0 to 1 which can be used for classification. The hyperbolic tangent (tanh) is another popular choice of bounded activation function, which produces an output between -1 and 1. Since it has a stronger gradient, neural network training often converges faster than with the sigmoid function [23]. For many years, sigmoid and hyperbolic tangent functions were the commonly used activation functions. However, both suffer from the vanishing gradient problem, which hinders the efficient training of deep neural networks with many layers [24].

It was shown that neural networks with unbounded activation functions have the universal approximation property and reduce the problem of vanishing gradients. In recent years, numerous unbounded activation functions have been proposed for neural networks, with the softplus [25] and the rectified linear unit (ReLU) [26] activation function being notable examples. These activation functions, especially ReLU has a pivotal role in improving the training and performance of deep learning models. ReLU has been a cornerstone of deep learning models due to its computational efficiency and effectiveness in addressing the vanishing gradient problem. Since then, several variants of ReLU have been proposed including Leaky ReLU [27], sigmoid linear unit [28] and exponential linear unit [29], each offering unique advantages for building deep learning-based applications.

2.4. Parameter Learning and Loss Functions

The weights (parameters) of deep learning models are often optimized using an optimization algorithm called gradient descent, though other optimization algorithms may also be used. However, it has to be noted that gradient descent is a generic algorithm which can be used to solve a wide range of optimization problems. In general, gradient descent finds the optimal weights by iteratively updating the weights such that the weights will
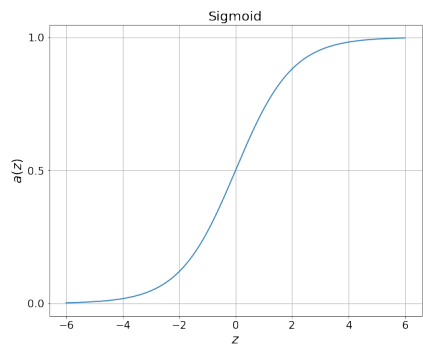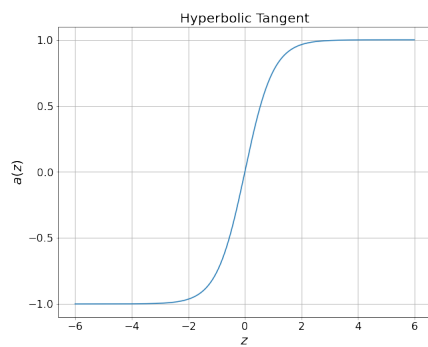
Figure 4: Sigmoid activation function.

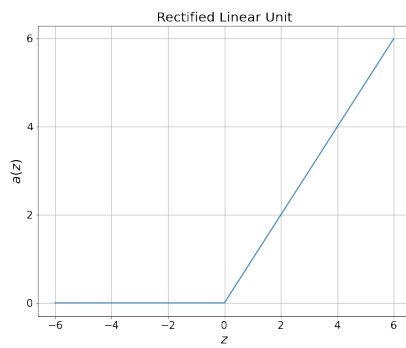Figure 5: Hyperbolic tangent activation function.

Figure 6: Rectified linear unit activation function.

result in a minimum prediction error over all instances in the training set. This process is synonymous with a person at the top of a hill who wants to climb down to the ground. Just like the person may choose a path that leads to the lowest point by taking small steps based on the steepness of the slope, gradient descent makes small adjustments to the weights, moving them in the direction that reduces the error the most. The prediction error is quantified by a loss function. For classification problems, the commonly used loss function is the negative log-likelihood loss or cross entropy loss, while the square loss and absolute loss are used for regression problems [30]. The weights of layer l are updated as

$$w_{i,j}^l = w_{i,j}^l - \alpha \nabla_{w_{i,j}} \tag{8}$$

where α is a hyperparameter called learning rate and $\nabla_{w_{i,j}}$ is the gradient or the derivative of the loss function J with respect to the weight $\frac{\partial J}{\partial w_{i,j}^l}$.

The gradient can be computed across all training set instances, an approach known as batch gradient descent. However, this approach does not always guarantee convergence to the optimal solution, as it may get stuck in local minima or saddle points, and the same gradient is used for every weight update. An alternative approach is to perform the weight update on the basis of a single instance, but the approach results in a noisy gradient and becomes computationally intensive due to the frequent weight update. A more commonly used approach is to perform the weight update over a set of training instances, known as mini-batch gradient descent. This approach strikes a balance, providing a less noisy gradient and a more stable training process. For a more hands-on understanding of neural network training, readers are encouraged to explore TF playground [31] and Initializing Neural Networks [32], interactive tools demonstrating how neural networks are trained and the impact of hyperparameters.

Several efforts have been made to improve the efficiency of gradient descent. One of the earlier efforts is the inclusion of past rate of change in the weight update to speed up the training of deep learning models, the algorithm is called gradient descent with momentum [33]. Another effort is to improve the training convergence by adapting the learning rate based on the occurrence of the features [34]. A more recent work utilizes both adaptive learning rate and momentum to improve the training efficiency and convergence of deep learning models [35].

2.5. Regularization Methods

Regularization methods are employed to prevent overfitting in deep learning models and improve their generalization performance. Early stopping is a method that can detect the onset of overfitting during training by continuously monitoring the validation error. The model is considered overfitting if the validation error starts to increase at some point of the training while the training error is decreasing. However, detecting the onset of overfitting during the training of deep learning models is challenging due to the inherent stochasticity and the presence of noisy data. Several stopping criteria can be considered, such as using a threshold to check if the decrease of (average) validation error is significant and counting the number of successive increases of validation error [36].

Dropout is a regularization method that randomly switches off some neurons in the hidden layers during training with a predefined drop probability (dropout rate) [37]. Dropout has the effect of training and evaluating a large number of different subnetworks within the models. The dropout rate is a hyperparameter that needs to be carefully tuned to balance regularization and model capacity. Different ranges of dropout rate have been suggested. The original author suggested a dropout rate between 0.5 and 0.8 [37] while others recommended a lower dropout rate between 0.1 and 0.3 [38]. Furthermore, it has been suggested a low dropout rate due to the exponential increase in the volume of training data [39].

Parameter norm penalty is a regularization method that adds a penalty term consisting of the network's weights to the loss function. During the training, the penalty term discourages large weight values and hence, constraining the model's capacity and reducing the chance of overfitting. The common penalty terms are L^1 norm penalty [40], L^2 norm penalty, also known as weight decay and a combination of L^1 and L^2 [41]. An adaptive weight decay is proposed, allowing the regularization strength for each weight to be dynamically adjusted [42].

Despite the advantages of the mini-batch gradient descent, each mini-batch may comprise data from different distributions. Furthermore, the data distribution may change after each weight update, which could slow down the training process. Batch normalization overcomes this issue by normalizing the summed input to a neuron over a mini batch of training instances [43]. An alternative method is to perform normalization across the neurons instead of the mini batch, a method known as layer normalization [44]. Layer

15

normalization is applicable in recurrent neural networks and overcomes the dependencies on the mini batch size.

3. Types of Deep Learning

Deep learning models can be categorized into deep supervised learning and deep unsupervised learning.

3.1. Deep Supervised Learning

Deep supervised models are trained with a labelled dataset. The learning process of these models involves calculating the prediction error through a loss function and utilizing the error to adjust the weights iteratively until the prediction error is minimized. Among the deep supervised models, three important models are identified, namely multilayer perceptron, convolutional neural network and recurrent neural network.

3.1.1. Multilayer Perceptron

Multilayer perceptron is a neural network model with one or more hidden fully connected layers stacked between the input and output layers as shown in Fig. 7. The width (number of neurons) of the hidden layers and the depth (number of layers) of the network influence the model's ability to learn patterns in the data. Specifically, the width affects the network's ability to capture a broader range of features, while the depth facilitates the learning of hierarchical representations. Nevertheless, studies indicated that a multilayer perceptron with a single hidden layer can approximate any continuous function [45], [46]. Multilayer perceptron is effective in various industries and applications from healthcare to finance [47]. However, a multilayer perceptron requires the input data to be structured in a one-dimensional format (e.g., tabular data), making it less suitable for unstructured data such as images, text, and speech. To leverage multilayer perceptron for unstructured data, a feature extraction or transformation into structured data is necessary.

3.1.2. Recurrent Neural Network

Recurrent neural network (RNN) is a neural network model that leverages the sequential information and memory through the use of recurrent connections, allowing it to effectively process data such as time series, text, speech and other sequential patterns. As shown in Fig. 8, a recurrent neural
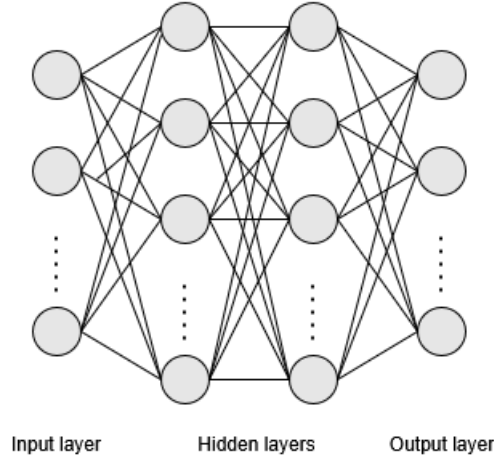
Figure 7: A fully connected neural network.

network is characterized by the recurrent connection, which enables the network to loop back and use internal state from the previous time step to the next time step. The internal state is parameterized by a set of weights shared across the sequence of data. The training of recurrent neural networks suffers from the issue of vanishing gradient due to the challenges of propagation of gradients over a long sequence of data. Variants of recurrent neural networks are introduced to overcome the problem of vanishing gradient, such as long short-term memory (LSTM) [48] and gated recurrent memory (GRU) [49]. The improved recurrent neural networks introduce memory cell and gating mechanisms to retain and discard information in every time step, allowing for more effective learning dependencies in long sequence. The network architecture can be built using fully connected and convolutional layers [50].

LSTM is arguably the most widely used variant and has been applied in various applications such as image captioning, machine translation and sentiment analysis [51]. Similar to the standard RNNs, LSTMs process information steps by steps using a chain of repeating units. Each LSTM unit consists of several gates, namely the forget gate, input gate and output gate that control the flow of information from one time step to the next time step. Figure 9 illustrates how information flows through these gates. The key component of an LSTM unit is the cell, represented by the horizontal line running through the top of the block. This cell maintains the state at each time step and is updated by the gates. Specifically, at each time step, the
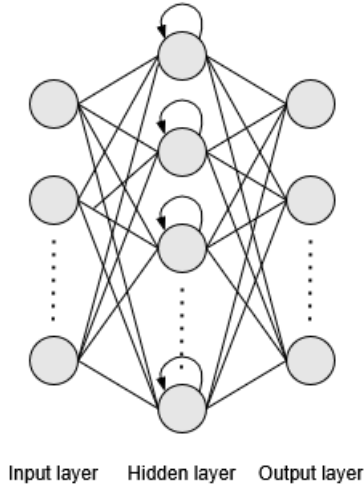
17

Input layer   Hidden layer   Output layer

Figure 8: A neural network with recurrent connection.

forget gate determines which information to be retained or discarded from the cell state based on the current input and the previous hidden state. The input gate decides what new information should be added to the cell state, while the output gate uses the updated cell state, current input and previous hidden state to produce the hidden state for the current time step.

GRU is a popular variant of RNN that is similar to LSTM but has a relatively simpler architecture. The network has two gates called reset and update to control the information flow. Unlike LSTM, GRU does not maintain a separate internal cell state, but uses the reset gate to determine which parts of the information to be retained and discarded, and the update gate to control how much of the previous hidden state should be passed to the current hidden state. Figure 10 illustrates the interconnection of the gates in a GRU unit.

3.1.3. Convolutional Neural Network

Convolutional neural network (CNN) is a neural network model that preserves and leverages the spatial local information in the data through the use of convolutional layers. Fig. 11 shows a typical architecture of a convolutional neural network which consists of convolutional, pooling and fully connected layers. The convolutional and pooling layers are stacked alternately to automatically extract salient features in a hierarchical manner.
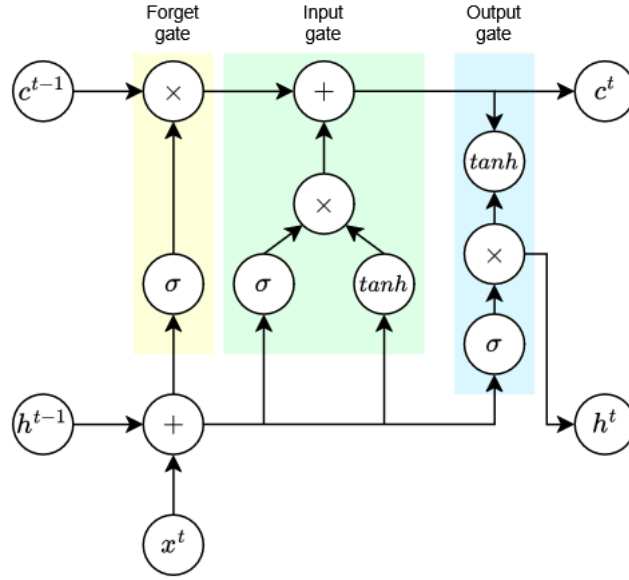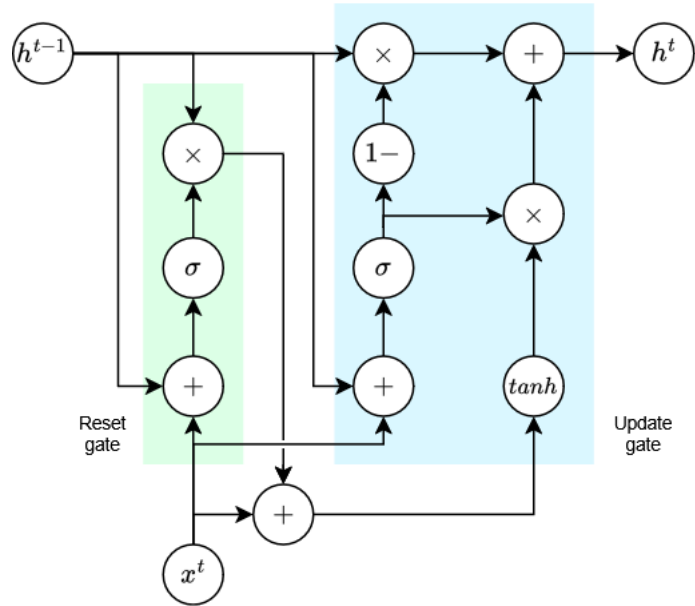
Figure 9: The architecture of an LSTM unit.

Figure 10: The architecture of an LSTM unit.

The extracted features are then fed to fully connected layers to predict the outputs. The final feature maps need to be converted to a one-dimensional vector before they are fed to the fully connected layers. The conversion can be performed by flattening the feature maps. CNN architecture is crucial in increasing the performance of the prediction, as it is designed to efficiently extract the feature representation of the input data, enabling more accurate and robust pattern recognition. Over the last decade, several CNN architectures have been proposed, whereby the focus of the improvements has been on enhancing the feature learning capabilities and addressing challenges such as vanishing gradient and diminishing feature reuse.
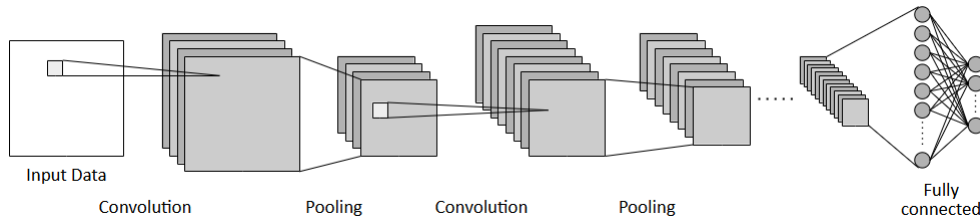

Figure 11: A neural network with convolutional and pooling layers followed by fully connected layers.

AlexNet is among the first CNN models that gained widespread recognition and success, marking a significant achievement in the field of deep learning for computer vision tasks [52]. The model consists of five convolutional layers with maximum pooling operation performed after the first and second convolutional layers, followed by three fully connected layers. The first and second convolutional layers utilize a filter size of 11×11 and 5×5 respectively, and 3×3 filter size is used for the remaining convolutional layers. ReLU activation function is used to mitigate the vanishing gradient. AlexNet is the first deep learning architecture that demonstrated CNN's capability for large-scale image recognition. While 8 layers was considered deep for its time, later deep learning architectures demonstrated that even deeper networks could achieve better performance.

ZFNet is a classic CNN model which has a similar architectural principle as AlexNet, featuring five convolutional layers with maximum pooling layers after the first and second convolution, followed by three fully connected layers [53]. The significant differences are the use of smaller filter size and stride in the convolutional layers and contrast normalization of the feature maps, which allows the model to capture better features and improve

20

the overall performance. This configuration improves feature extraction and performance compared to AlexNet. However, like AlexNet, its limited depth restricted its ability to learn more complex hierarchical features.

Network-in-network introduces two innovative concepts to enhance the performance of the model [54]. The first was introducing a block of convolutional layers consisting of $k \times k$ convolution followed by two 1×1 convolution operations. The pointwise convolutions are similar to applying a multilayer perceptron on the feature maps, allowing the model to approximate more abstract feature representations. In the preceding models, the final feature maps are vectorized by flattening operation for classification by the fully connected layers. Instead of flattening, network-in-network model calculates the spatial average of each feature map, and the resulting vector is fed to softmax function for classification. This approach is parameter-less, significantly reducing the number of parameters. Although this approach makes it more computationally efficient, its general performance does not always surpass models with deeper architecture.

VGGNet attempts to improve the CNN architecture by adding more convolutional layers, specifically up to 19 layers to capture more intricate feature representation from input data, followed by three fully connected layers [55]. ReLU activation function is used to reduce vanishing gradient. Unlike AlexNet, all convolutional layers utilize a small fix filter size of 3×3 and maximum pooling layer is added after a stack of two or three convolutional layers. This configuration allows the model to extract more discriminative features and decreases the number of parameters. The architecture provides better generalization and finer feature extraction, but it comes at the cost of a large number of parameters (138 million), making it more computationally expensive for practical applications.

GoogleNet or Inception v1 leverages the fact that visual data can be represented at different scales by incorporating a module which consists of multiple convolutional pipelines with different filter sizes [56]. The module known as inception utilizes three kernel sizes (5×5, 3×3, 1×1) to capture spatial and channel information at different scales of resolution as shown in Fig. 12. This configuration enables a more effective feature extraction at both fine-grained and coarse-grained information from input data while maintaining computational efficiency. The model architecture utilizes the inception module at the higher layers, while the traditional convolution and maximum pooling block is used to extract primitive and basic features. The inception modules are stacked upon each other, with maximum pooling oper-

ation is performed occasionally to reduce the spatial resolution of the feature maps. GoogleNet utilizes global average pooling to vectorize the final feature maps before passing it to a fully connected layer for classification. However, the complexity of the model architecture may hinder its interpretability and tuning compared to simple models. GoogleNet has been further enhanced in later Inception models by introducing batch normalization, auxiliary classifier and deeper architecture.
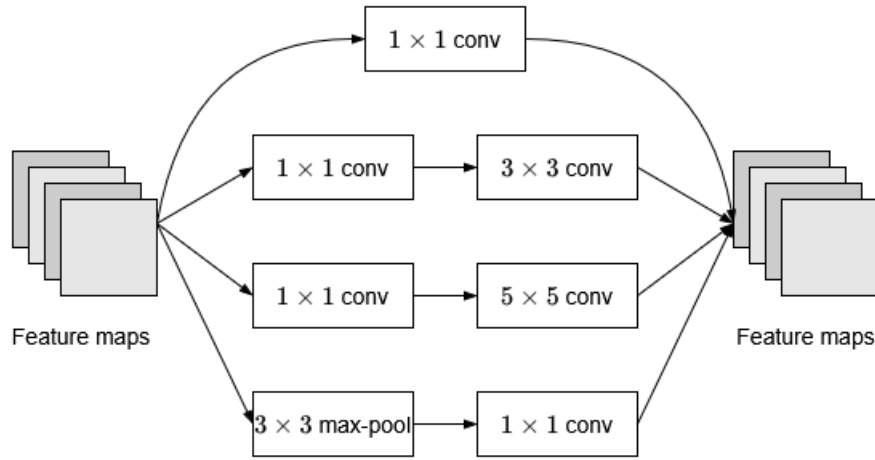

Figure 12: The inception module. Adapted from [56].

Increasing the number of layers enhances the model performance, mainly for solving complex tasks. However, training a very deep neural network is challenging due to the vanishing gradient problem, where the gradients that are used to update the network become insignificant or extremely small as they are backpropagated from the output layer to the earlier layers. A model called Highway Network overcomes this issue by introducing a gating mechanism that regulates the information flow of the layers, enabling the flow of information from the earlier layers to the later layers [57]. Consequently, this not only mitigates the vanishing gradient problem, but also renders the gradient-based training more tractable, enabling the training of very deep neural networks consisting as many as 100 layers. However, the gating mechanism increases the model complexity, making the model mode resource-intensive and less suitable for real-time applications.

The gating mechanism of Highway Network increases the number of parameters for regulating the information flow. ResNet is a CNN architecture

that incorporates residual (skip) connection that allows information to bypass certain layers, mitigating the vanishing gradient problem [58]. ResNet architecture stacks residual blocks, which consists of two or three of convolutional layers with batch normalization and ReLU, and a skip connection which adds the input to the output of the final convolutional layer as shown in Fig. 13. If the input dimension does not match with the residual output dimension, a linear projection is performed by the residual connection to match the dimensions. This concept of "feature reuse" is central to ResNet's design, as the skip connection allows the features learned in the previous layers to be directly reused in the layer layers, enhancing the model's ability to learn hierarchical features. In comparison to the gating mechanism of Highway Network, the residual connection is parameter-free, and thus does not incur additional computational costs. Furthermore, the connections are never closed whereby all information is always passed through the layers. This innovative concept enables the training of very deep neural networks boasting as many as 152 layers. However, as the model gets deeper, the effectiveness of skip connection diminishes and the performance gain is negligible. Furthermore, ResNet can be more complex to implement and fine-tune compared to simpler architecture.
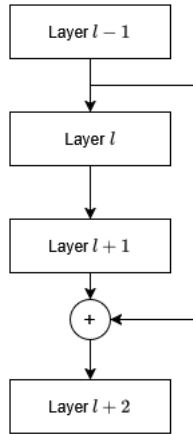

Figure 13: A residual connection which explicitly incorporates features from previous layers into the later layers. Adapted from [58].

DenseNet is another CNN architecture that overcomes the vanishing gradient problem. DenseNet follows the same approach as ResNet and Highway Network, utilizing skip connection to allow information flow from the earlier

layers to later layers. However, DenseNet takes this concept one step further, by introducing a dense block consisting of multiple convolution functions (layers) with each convolution function performs batch normalization followed by ReLU and 3×3 convolution. Each convolutional layer in the dense block receives feature maps from all its preceding layers. Hence, the connection is referred to as a dense connection [59]. This configuration as shown in Fig. 14 maximizes information flow, feature reuse and preserves the feed-forward nature of the network, improving the feature learning. To reduce the computational costs, a block of 1×1 convolutional with batch normalization and maximum pooling layers known as transition block is used to reduce the spatial dimension of the feature maps. The model architecture integrates these dense and transition blocks, stacking them alternately. The network depth can reach up to 264 layers. However, Dense blocks can become computationally expensive due to the increasing number of feature maps. Furthermore, it is prone to overfitting due to the dense connection between the layers.
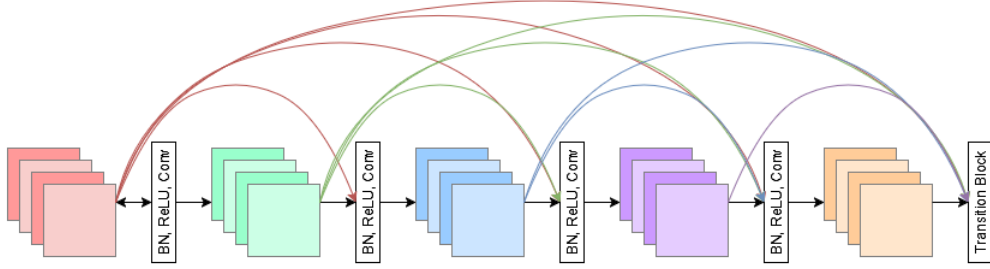
Figure 14: A 5-layer dense block. Adapted from [59]

Although skip connections in ResNet effectively mitigate the vanishing gradient problem, a new challenge arises in the form of diminishing feature reuse as the network becomes deeper. Diminishing feature reuse refers to the diminishing effectiveness of the previously learned feature maps in subsequent layers, impacting the final prediction. WideResNet is a CNN architecture that is based on ResNet with the aim to mitigate diminishing feature reuse problem. Instead of making the network deeper, WideResNet makes the network wider by increasing the number of channels by k factor [60]. The increased width allows the model to capture more diverse features, enhancing its ability to learn complex relationships in the input data. This configuration improves feature learning efficiency while mitigating depth-related issues such

as vanishing gradients and overfitting. However, wider models may increase memory usage and computational costs.

ResNeXt addresses the diminishing feature reuse by capturing more efficient and diverse features of the input data. ResNeXt introduces a concept of cardinality which is loosely based on the inception module as shown in Fig. 15. Cardinality refers to the number of independent and identical paths, where each path performs transformation of the input data, divided along the channel dimension [61]. In other words, instead of solely relying on increasing the depth of the model, ResNeXt enhances the feature learning by parallelizing the feature extraction through this cardinal path. In the proposed architecture, each path configuration is similar to the residual block of ResNet. The output from each path is then aggregated to form a comprehensive and diverse representation of the input data. The skip connection is used to mitigate the vanishing gradient problem. ResNeXt improves feature learning efficiency by balancing the depth and the width of the model, but its increased architectural complexity introduces additional parameters such as the number of cardinality, and increases memory usage and computational costs.
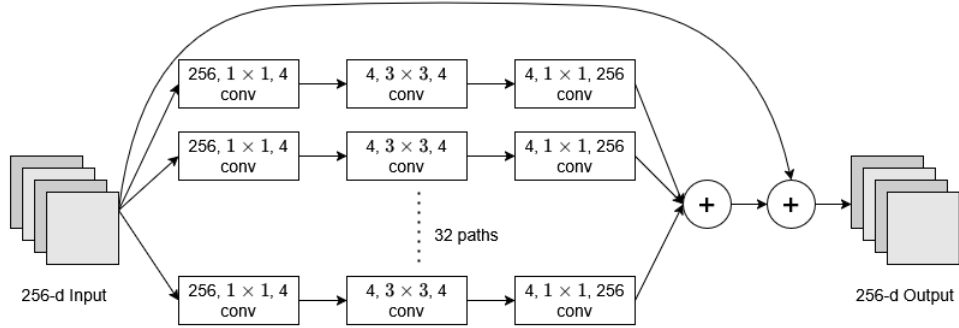
Figure 15: A cardinal block. Adapted from [61]

These CNN models were trained using the ImageNet [62] and/or CIFAR [63] datasets. ImageNet is considered as the most influential and important dataset in deep learning for computer vision research. The dataset was used to train all popular CNN models such as VGGNet, ResNet and DenseNet due to its large number of labeled images. The dataset contains 1.2 million training images, 50,000 validation images and 100,000 test images across 1000 object classes. CIFAR is a comparatively smaller dataset and includes

two variants: CIFAR-10 and CIFAR-100. Both contain 50,000 training images and 10,000 test images, with CIFAR-10 covering 10 object classes and CIFAR-100 covering 100 classes.

Figures 16 and 17 show the top five classification error rates of the CNN models on ImageNet and CIFAR-10, respectively. Error rates for CIFAR-100 are not reported, as it was used only by Network-in-network, DenseNet, Wide ResNet and ResNeXt. It is also noted that the error rates on ImageNet are sourced from the models' published manuscripts, which may differ from the values reported on the ILSVRC website. As shown in the figures, error rates are generally similar between the datasets, despite their differences in complexity. This could be attributed to ImageNet's large number of training images, allowing the models to generalize well across its 1000 classes. Furthermore, it is evident that the newer models, such as ResNeXt, DenseNet, Inception-v4 and ResNet significantly outperform older models due to their more advanced architectures. Among the evaluated models, DenseNet achieved the best performance on CIFAR-10 with a 3.46% error rate, while Inception-v4 achieved the lowest error rate on ImageNet at 3.70%.

3.1.4. Applications of Deep Supervised Learning Models

The deep supervised learning models have become integral to advancements in various applications such as medical imaging, internet of things and robotics due to their ability to extract complex hierarchical features. Typically, the studies utilize transfer learning to build the predictive models for solving the problem at hand, especially in scenarios with limited labelled data. This approach leverages pre-trained models to accelerate model development and improve accuracy. In [64], VGGNet-16, Inception-v3 and ResNet-50 are employed as pre-trained models for breast cancer screening. Specifically, the weights of the convolutional layers are frozen, and the fully connected predicting layers are replaced with new ones. The models are then fine-tuned to adapt to the specific characteristics of the breast cancer dataset, improving their performance on the target task. In medical image segmentation, ResNet [65] and DenseNet [66] have been utilized to progressively capture rich spatial features from the input images, systematically reducing their dimensions while preserving essential information. These compressed feature representations are then gradually expanded and refined through upsampling operations, followed by localizing and predicting the segmentation regions.

In other studies, the architecture of the CNN models have been enhanced
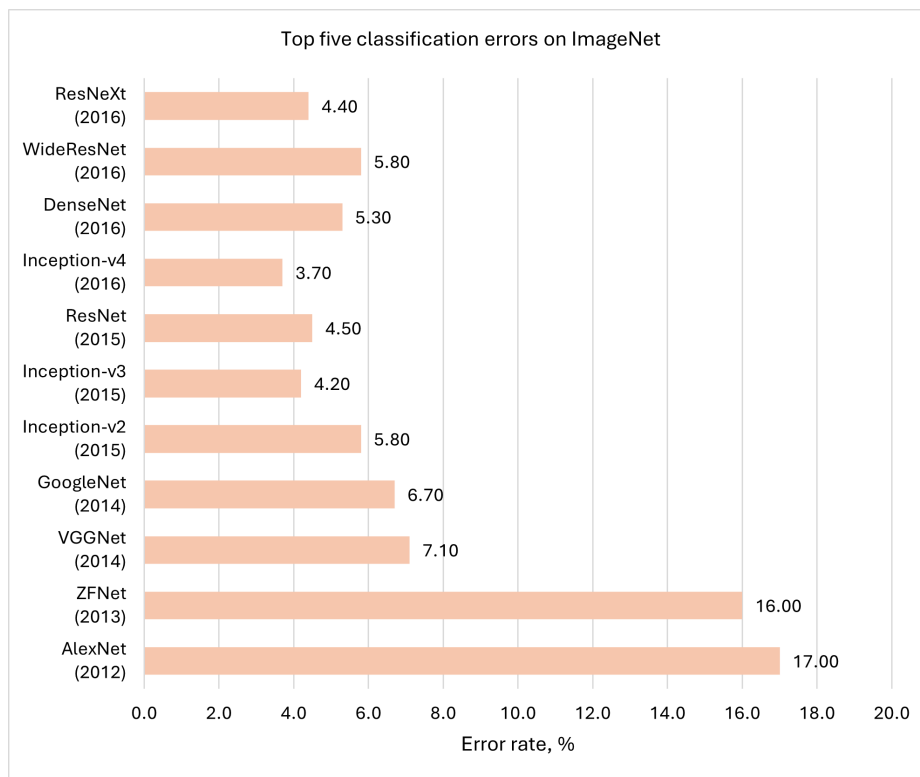
Figure 16: Top five classification errors on ImageNet.

to improve their feature extraction capability. In [67], ResNet architecture is modified by replacing ReLU with LeakyReLU and incorporating self-attention before the predicting layers to assign weights based on the feature importance. In another study, the residual block of ResNet-18 is enhanced by integrating a channel attention before the skip connection addition [68]. Additionally, a channel attention featuring two parallel processing pipelines is introduced before the predicting layers. One pipeline applies maximum pooling while the other uses average pooling, allowing the model to capture diverse features. Each pooled output is then processed by convolutional layers, and the resulting feature maps are concatenated to combine the extracted features for prediction. In [69], DenseNet is modified to reduce feature reuse by controlling the number of feature maps inputs to each layer using a parameter. This parameter controls the number of feature maps based on the distance between layers, whereby if the distance between two layers
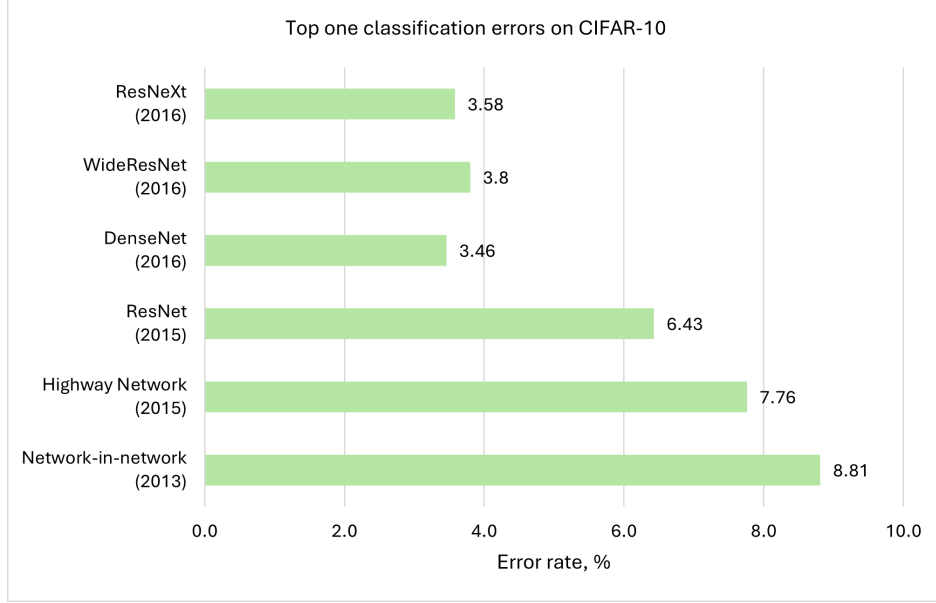
Figure 17: Top one classification errors on CIFAR-10.

is large, the input feature maps are reduced to half of the original number. Additionally, the network width is modified based on depth, leading to a gradual widening of the network as it deepens. The modification reduces the complexity of DenseNet, improving both memory usage and computational efficiency.

The LSTMs and GRUs have been exploited to capture temporal information in time series data. In [70], AlexNet, VGG16 and ResNet50 are used to extract spatial features from ECG images, while LSTM and GRU are used to extract temporal features from the ECG signals. The features are then combined and passed to a self-attention module, followed by fully connected layers for predictions. In [71], 1D-CNN is hybridized with LSTM to exploit the spatial feature extraction capabilities of convolutional layers and the temporal sequence learning capabilities of LSTM. The time series data is first processed by a series of convolutional and maximum pooling layers to extract spatial features. These features are then fed into 10 layers of LSTM capture temporal dependencies and sequential patterns within the data, improving the performance of the predictions. A hybrid deep learning model with parallel feature learning pipelines is introduced for classifying motion signals into human activities [72]. The feature learning pipelines consist of convolutional

and maximum pooling layers to extract local features from the sequence of segmented signals (windows). The local features are concatenated to form a sequence, which is then fed into an LSTM to capture temporal dependencies and patterns. A similar hybrid model is proposed, combining ResNet-18 with an LSTM for EEG signal classification. The feature maps produced by ResNet-18 are flattened and fed into the LSTM, followed by a fully connected layer for prediction.

3.2. Deep Unsupervised Learning

Deep unsupervised models are trained with an unlabeled dataset. The learning process of these models discovering patterns, structures, and representations within the data without relying on explicit labels or supervision. Instead, these models often learn by optimizing objective functions that capture the underlying data characteristics such as clustering, learning useful feature embeddings and reconstructing input data from compressed representations. Examples of deep unsupervised models are autoencoders, generative adversarial networks and restricted Boltzmann machines.

Restricted Boltzmann Machine is a generative neural network model that learns a probability distribution based on a set of inputs. The model consists of a visible (input) layer and a hidden layer with symmetrically weighted connections as shown in Fig. 18. The input layer represents the input data with each node corresponding to a feature or variable while the hidden layer learns the abstract representation of the input data. Restricted Boltzmann machine model is trained using contrastive divergence, an algorithm that is based on a modified form of gradient descent, utilizing a sampling-based approach to estimate the gradient [73]. It has found success in solving combinative problems such as dimensionality reduction, collaborative filtering and topic modelling.

Deep Belief Network can be viewed as a stack of restricted Boltzmann machines, comprising a visible layer and multiple hidden layers [74] as shown in Fig. 19. Deep belief network has two training phases. The initial phase is known as pretraining in which the network is trained layer by layer, with each layer serves as a pretraining layer of the subsequent layers. This sequential learning allows the hidden layers learn complex hierarchical feature representation of the data. The second phase is called fine-tuning whereby the deep belief network model can be further trained with supervision to perform tasks such as classification and regression [75].
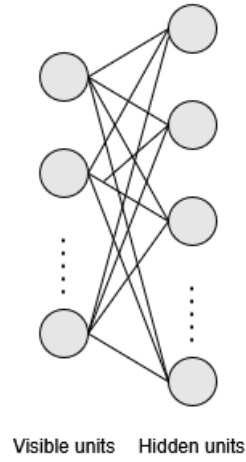
Figure 18: A restricted Boltzmann machine.
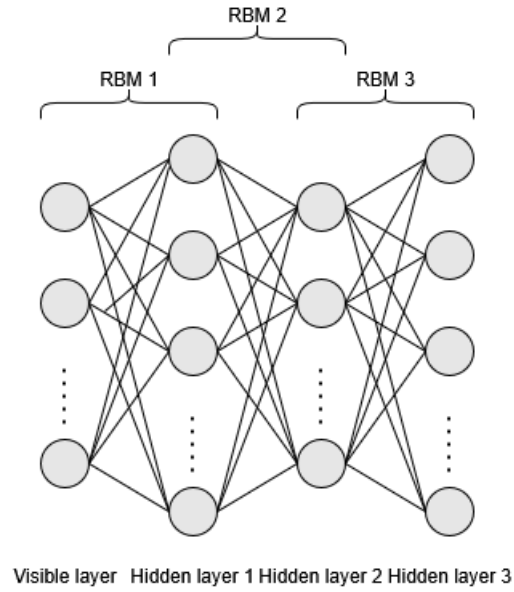
Figure 19: A deep belief network.

An autoencoder is a generative neural network model that learns to encode the input data into a compressed representation and then reconstructs the original data from this representation. The layers that encode the input data are known as encoder while the layers that responsible for the recon-

struction are referred to as the decoder as shown in Fig. 20. The encoded data (hidden layer) represents the abstract features of the input data, also known as latent space or encoding. The decoder can be removed from the autoencoder, creating a standalone model that can be used for data compression and dimensionality reduction [76], [77]. The decoder can also be replaced with predictive layers for classification task [78]. The network architecture can be built using fully connected and convolutional layers [79].
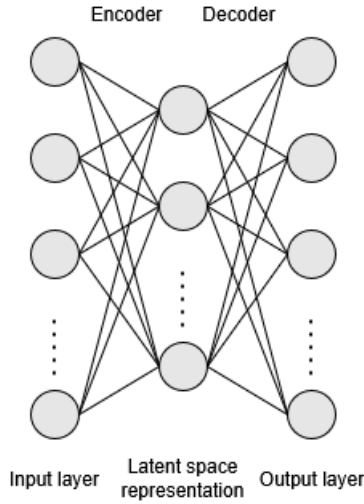
Figure 20: An autoencoder.

Several autoencoder variants have been introduced to improve the autoencoder's ability to capture better feature representation. Some introduced penalty terms to the loss function such as sparsity penalty (sparse autoencoder) [80] to encourage sparse representation and Jacobian Frobenius norm (contractive autoencoder) [81] to be less sensitive to small and insignificant variations in the input data while encoding the feature representation. Others trained the autoencoder to recover original data from corrupted data with noise [82]. An improved denoising autoencoder knowns marginalized denoising autoencoder has been proposed which marginalizes the noise by adding a term that is linked to the encoding layer [83]. Variational autoencoder is a variant of autoencoder that has similar architecture: encoder, latent space and decoder. Despite the similarity, instead of learning a fixed encoding, variational autoencoder learns the probability distribution of the input data in the latent space [84]. The model can be used to generate data by sampling

from the learned probability distribution. The network architecture can be built by stacking more than one fully connected layer and convolutional layer.

Generative Adversarial Network (GAN) is another generative neural network model that is designed for generating data that adheres closely to the distribution of the original training set. The model consists of two different neural networks namely generator and discriminator as shown in Fig. 21. The generator learns to imitate the distribution of the training set given a noise vector, effectively outsmarting the discriminator. Simultaneously, during the training, the discriminator is trained to differentiate between the real data from the training set and synthetic data generated by the generator [85]. This intricate dynamic between the networks drives an iterative learning process whereby the generator continually refines its ability to create synthetic data that closely resembles the real data, while the discriminator enhances its ability to distinguish between authentic and fake data.
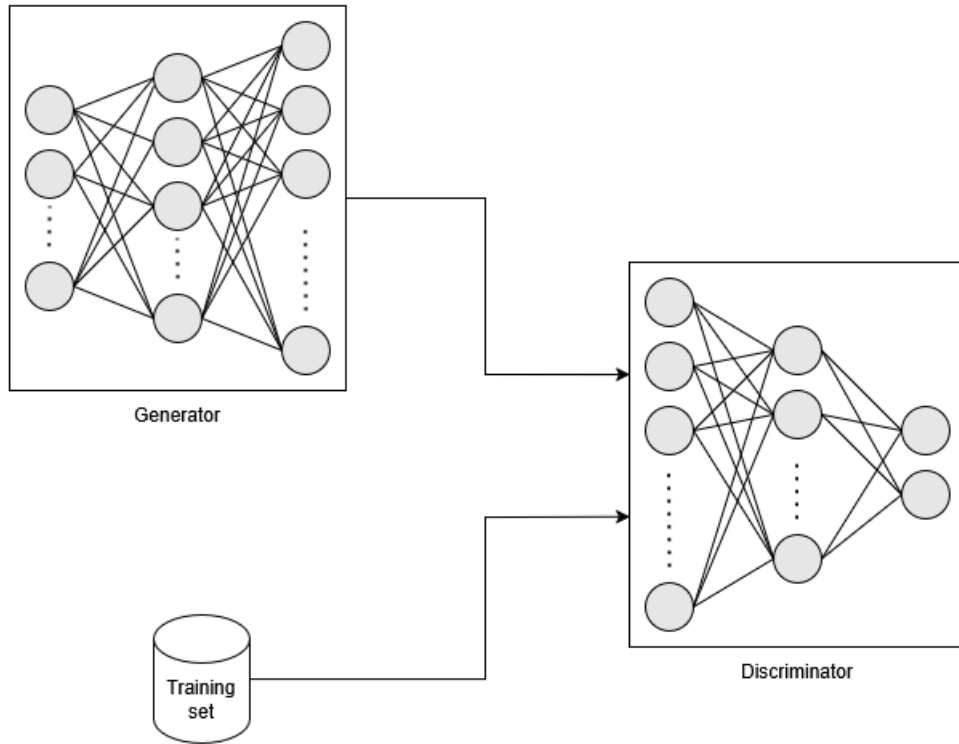
Figure 21: A generative adversarial network.

The model can be extended by providing the labels to both generator and discriminator in which the model known as conditional GAN, capable of generating 1000 image classes [86]. Conditional GANs require a labelled dataset, which might limit its application. InfoGAN is similar to conditional GAN, but the labels are substituted with latent codes, which allows the model to be trained in an unsupervised manner [87]. GANs often suffer from mode collapse, where the model can only generate a single or small set of outputs. Wasserstein GAN improves the training by utilizing Wasserstein loss function, which measures the difference between the real and synthesized data distribution [88]. ProGAN tackles the training instability of GAN by progressively growing the generator and discriminator. The idea is that the model is scaled up gradually, starting with the simplest form of the problem, and little by little the problem's complexity is increased as the training progresses [89]. StyleGAN leverages the progressive GAN's approach and neural style transfer to improve the quality of the generated data [90]. The model is characterized by the independent manipulation of both style and content, allowing it to generate diverse styles and high-quality data.

4. State-of-the-art Deep Learning Applications

As discussed in the previous section, the application of deep learning ranges from computer vision [91], natural language processing [92], healthcare [93], robotics [94], education [95], and many others. This section presents the applications of deep learning across several areas.

4.1. Computer Vision

Computer vision is an essential field in artificial intelligence (AI). It is a field of study that focuses on enabling computers to acquire, analyze and interpret visual inputs to derive meaningful information. The visual inputs can take many forms such as digital images, sequence of images or video and point cloud, and the source of these inputs can be camera, LiDAR and medical scanning machine. Deep learning, specifically CNN models have been widely used in real-world computer vision applications including image classification, object detection and image segmentation. This section discusses more details about the recent advancements in deep learning models that have been achieved over the past few years.

4.1.1. Image Classification

Image classification is a fundamental task in computer vision, which involves categorizing an image into one of predefined classes based on the visual content. The objective of image classification is to enable computers or machines to differentiate between objects within images, in a manner similar to how humans interpret visual information. Image classification is a crucial component in various applications such as robotics, manufacturing, and healthcare. LeNet-5, introduced in 1998, is one of the earliest convolutional neural networks that was successfully trained to classify handwritten digits. The model underwent a series of improvements, including the use of tanh and average pooling, which enhanced its ability to extract hierarchical features, ultimately improving overall performance. The model architecture comprises two convolutional layers, each with an average pooling layer, followed by two fully connected layers, including the output layer [96]. Since then, numerous CNN models have been proposed based on LeNet-5 for image classification [97, 98] but the most significant one is AlexNet in 2012, which saw a transformative breakthrough in deep learning. AlexNet is considered the first CNN model with a large number of parameters that significantly improved the performance of image classification on a very large dataset (ImageNet). The model won first place in ILSVRC 2012, improving the test error from the previous year by almost 10% [52]. Numerous significant CNN models have been introduced in subsequent ILSVRC competitions including ZFNet, VGG16, GoogleNet, ResNet and ResNext. In general, the research focused on increasing the number of layers, addressing the problem of vanishing gradient and diminishing of feature reuse.

Research in image classification continues to evolve with a focus on addressing key challenges to improve the classification performance. One notable trend is the formulation of the loss function to address problems such as neglecting well-classified instances and imbalance distribution of class labels. In a particular study, an additive term is introduced to the cross-entropy loss to reward the models for the correctly classified instances. This formulation encourages the models to also pay attention to well-classified instances while focusing on the bad-classified ones [99]. Another study proposes an asymmetric polynomial loss function using the Taylor series expansion. The loss function allows the training to selectively prioritize contributions of positive instances to mitigate the issue of imbalance between negative and positive classes [100]. The asymmetric polynomial loss requires a large number of pa-

rameters to be fine-tuned and may lead to overfitting. A robust asymmetric loss is formulated by introducing a multiplicative term to control the contribution of the negative gradient and making it less sensitive to parameter optimization [101]. Combining multiple deep learning models improves the overall performance by leveraging the diverse strengths of individual models. However, identifying the optimal combination is non-trivial due to the large number of hyperparameters. A straightforward method is to employ the weighted sum rule [102]. To enhance the overall performance, an algorithm, named greedy soups, adds a model based on the validation accuracy [103]. The final prediction is produced via averaging. Multi-symmetry ensembles framework improves the building of diverse deep learning models by utilizing contrastive learning [104]. Then, the diverse models are sequentially combined based on their validation accuracy.

Vision transformers (ViT) offers an alternative to convolutional neural networks that have long been the dominant architecture for image classification, by leveraging self-attention mechanisms for scalable global representation learning. Despite its effectiveness, ViT is sensitive to hyperparameter optimization and substandard performance on smaller datasets [105]. Furthermore, ViT lacks the ability to leverage local spatial features which is inherent in convolutional neural networks [106]. Therefore, several studies attempt to incorporate convolutional layers into ViT architecture to improve its performance and robustness. In particular, conformer is a network architecture with two branches: CNN branch and a transformer branch to extract local and global features respectively [107]. Both branches are connected by two "bridges" of 1×1 convolution and up or down sampling operations, allowing the branches to share their features and enhance the feature representation. Both branches output predictions which are combined to produce the final prediction.

A hybrid architecture, named MaxViT, combines convolutional networks and vision transformer to address the lack of scalability issues of self-attention mechanisms when the model is trained on large input size [108]. The improved vision transformer is composed of two modules whereby the first module attends local features in non-overlapping image patches and the global features are attended by processing a grid of sparse and uniform pixels. The transformer is stacked with a block of convolutional layers to extract local spatial features. The architecture of MaxViT is shown in Figure 22. Another study proposes a convolutional transformer network, introducing the depthwise convolutional block into the ViT [109]. This configuration allows the

model to exploit the ability of convolutional networks to extract local spatial features while the ViT attends the extracted local features to focus on relevant information, enhancing the model's ability to capture complex patterns and relationships. Specifically, instead of linear mapping, depth-wise convolutional mapping is used to generate query, key, and value matrices. Table 2 lists the summary of state-of-the-art image classification.
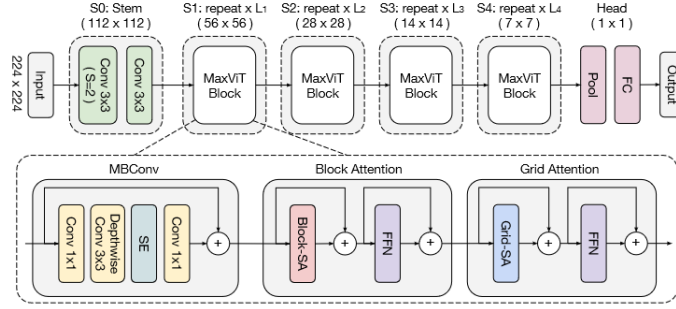
Figure 22: The architecture of MaxViT [108].

4.1.2. Object Detection

Deep learning plays a major role in significantly advancing the state-of-the-art in object detection performance. Region-based CNN (R-CNN) is the first breakthrough in object detection that combines CNN with selective region proposals [110]. The region proposals are the candidate bounding boxes serving as the potential region of interests (objects) within the input image, and the CNN are used to extract features from the region proposals and classify the regions for object detection. An improved model, named Fast R-CNN introduces two prediction branches: object classification and bounding box regression which improves the overall performance of object detection [111]. However, R-CNN and Fast R-CNN models are computationally expensive and slow, thus practically infeasible for real-time applications. Addressing this issue, Fast R-CNN is integrated with a region proposal network, referred to as Faster R-CNN [112]. The region proposal network (RPN) is used to efficiently generate region proposals for object detection. RPN takes an input image and output a set of rectangle object proposals, each with a confidence score to indicate the likelihood of an object's presence. To this end, RPN

36

Table 2: Summary of state-of-the-art image classification.

Ref.	Description	Results (Datasets and Metrics)
[99], 2022	The study introduced an encouraged loss that rewards well-classified examples with additive bonuses to enhance their contribution during training, addressing issues in representation learning, energy optimization, and makes the decision margin grows faster	CIFAR-10 Accuracy: 92.97% ImageNet Accuracy: 76.43%
[100], 2023	The study introduced asymmetric polynomial loss to address class imbalance by decoupling gradient contributions from positive and negative instances, increasing the impact of updates for minority instances and helping the model focus on the more challenging, less frequent cases	MS-COCO mAP: 90.08 NUS-WIDE mAP: 31.27
[101], 2023	The study introduced robust asymmetric loss to address class imbalance by emphasizing minority instances through an asymmetric weighting mechanism, allowing the model to focus more on the less frequent, yet critical minority instances	ISIC2018 Accuracy: 0.852 APTOS2019 Accuracy: 0.826
[107], 2023	This study introduced dual network structure to extract local and global feature representations using convolutional layers and transformer	ImageNet Accuracy: 81.3 (Conformer-Ti) Accuracy: 83.4 (Conformer-S) Accuracy: 84.1 (Conformer-B)
[108], 2022	This study introduced MaxViT, a hybrid attention model, combining convolutional layers with transformer-based architecture whereby it uses convolutional layers in the early stages to capture local patterns efficiently, and then applies a multi-axis attention mechanism to capture both local and global dependencies more effectively	ImageNet Accuracy: 85.72 (MaxViT-T) Accuracy: 86.19 (MaxViT-S) Accuracy: 86.66 (MaxViT-B) Accuracy: 86.70 (MaxViT-L)
[109], 2024	This study introduced convolutional transformer network that integrates convolutional layers and transformer encoder blocks whereby a transformer encoder block consists of multi-head self attention and feed-forward network layers with skip connections	Diving 48 Accuracy: 77.3 (CTN-a), 78.2 (CTN-b) Epic-Kitchens Accuracy: 44.1 (CTN-a), 45.9 (CTN-b)

introduces the concept of anchor boxes, whereby multiple bounding boxes of different aspect ratios are defined over the feature maps produced by the convolutional networks. These anchor boxes are then regressed over the feature maps to localize the objects, contributing to the improved speed and effectiveness of Faster R-CNN. The training of Faster R-CNN is divided into two stages. First, the RPN is pre-trained to generate the region proposals and then, the Fast R-CNN is trained using the region proposals generated by the RPN for object detection. The backbone network responsible for extracting the features for Faster R-CNN is either ZFNet or VGG16.

In two-stage object detectors, the region proposals are generated first, and then used for object detection. The two-stage process is computationally intensive and infeasible for real-time object detection applications. You Only Look Once or YOLO proposes a one-stage detection by directly predicting bounding boxes and object's confidence score in a single forward pass through the neural network [113]. This single pass architecture significantly reduces the computational complexity, making YOLO suitable for real-time object detection applications. In YOLO, the input image is divided into SS grids, each grid cell is responsible for detecting the objects present in the cell. Specifically, each grid cell predicts multiple bounding boxes and associated object's confidence score, enabling simultaneous object detection across the entire image. Subsequent enhancements such as YOLOv3 [114] and YOLOv4 [115] are proposed, improving the model's capability and accuracy. Single Shot Multibox Detector (SSD) is another one-stage detector, which aims to address the issue of real-time object detection [116]. SSD also eliminates the region proposal generation and directly predicts bounding boxes and confidence scores, reducing the computational complexity. To improve the overall performance, SSD produces the predictions from different levels of feature maps, allowing detection of objects of different sizes in the input image.

A common issue in object detection problems is the extremely imbalanced ratio of foreground to background classes. Addressing this issue, RetinaNet introduces a loss function that is based on the cross entropy called focal loss. Focal loss reduces the loss contribution of easily classified objects, allowing the model training to focus on the difficult objects [117]. RetinaNet adopts the Feature Pyramid Network (FPN) [118] with ResNet as the backbone network for extracting the feature maps. FPN is a network architecture with a pyramid structure that efficiently captures multiscale feature representation, facilitating object detection across various sizes. To further improve the

overall performance, EfficientDet introduces bidirectional FPN, which incorporates multi-level feature fusion to better capture multiscale feature representation [119]. Also, the model utilizes EfficientNet [120] as the backbone network to achieve a balance between computational efficiency and accuracy.

Object detection performance often relies on a post-processing step called non-maximum suppression (NMS) to eliminate duplicate detections and select the most relevant bounding boxes. Specifically, NMS sorts all detection boxes based on their confidence scores, selects a box with the maximum score and discards the other boxes with a significant overlap with the selected box. This process is repeated on the remaining detection boxes. However, due to the inconsistency between the confidence score and the quality of object localization, NMS retains poorly localized bounding boxes with high confidence score while discarding more accurate predictions with poor confidence score. To mitigate this limitation, instead of discarding the neighboring boxes with significant overlap, soft-NMS applies Gaussian function to lower their confidence scores [121]. The idea is not to discard the neighboring bounding boxes, but gradually decline their scores based on the extent of the overlap with the selected box. This results in a smoother suppression, preserving the better-localized bounding boxes.

Adaptive NMS introduces an adaptive threshold for the suppression of bounding boxes [122]. The algorithm dynamically adjusts the threshold based on the level of overlapping of the selected box with the other bounding boxes. Similar method is reported in [123] whereby an adaptive NMS is proposed by dynamically adjusting the suppression criteria based on intersection over union (IoU) values. The IoU values are compared with a threshold and applies an additional iteration to vote for each detected proposal, ensuring better distinction between closely positioned objects. The method helps preserve multiple high-confidence bounding boxes, mitigating the issue of incorrectly merging adjacent objects into one. Redundant bounding boxes are often not filtered out due to their low IoU values with the best bounding box, leading to high false positives. In [124], a learning-based NMS is proposed to reduce false positives by integrating the NMS process into the model's learning framework. Specifically, a novel NMS-aware loss is formulated that incorporates IoU to adjust bounding box weights, boosting negative attention for low IoU bounding boxes and enhancing positive weights for high IoU bounding boxes. Furthermore, regression assisted classification branch is introduced to aid classification by leveraging regression prediction relationships between bounding boxes and their best counterparts. In [125], the authors

formulated similarity distance metric is proposed to evaluate the similarity between bounding boxes. The proposed method considers the location and shape of the boxes, and adapts dynamically to different datasets and object sizes. This method improves label assignment by learning hyperparameters automatically, eliminating the need for manual tuning and enhancing detection performance for tiny objects detection.

Detection Transformer (DETR) is an end-to-end trainable object detection model that leverages the transformer architecture to eliminate the need for handcrafted components such as anchor boxes and non-maximum suppression [126]. The self-attention mechanism of the transformer captures the global context and relationships between different parts of the image, allowing it to localize the objects and remove duplicate predictions. The model is trained with a set of loss functions that perform bipartite matching between the predicted and ground truth objects. DETR uses ResNet as backbone network. Despite the success of DETR in simplifying and improving object detection tasks, DeTR suffers from a long training time and low performance at detecting small objects due to its reliance on the self-attention mechanism of the transformer, which lacks a multiscale feature representation. To mitigate this limitation, Deformable DETR introduces a multiscale deformable attention module which can effectively capture feature representation at different scales [127]. Furthermore, the attention module leverages deformable convolution, allowing the model to adapt to spatial variation and capture more informative features in the input data.

Dynamic DETR addresses the same issues by utilizing a deformable convolution-based FPN to learn multiscale feature representation [128]. The model replaces the transformer encoder with a convolution-based encoder to attend to various spatial features and channels. Moreover, an ROI-based dynamic attention is introduced at the transformer decoder, allowing the model to focus on the region of interests. This modification allows the model to effectively detect small objects and converge faster during training. The architecture of dynamic DETR is shown in Figure 23. DETR utilizes bipartite matching between the ground truth and the predicted objects to assign each ground truth object to a unique prediction, limiting the localization supervision. In [129], a hybrid matching scheme is introduced, combining the original one-to-one matching with an auxiliary one-to-many matching during training, improving training efficiency and detection accuracy without adding inference complexity. A training scheme known as Teach-DeTR is proposed to improve the overall performance of DeTR [130]. The train-

ing scheme leverages the predicted bounding boxes by other object detection models during the training by calculating the loss of one-to-one matching between the object queries and the predicted boxes. Table 3 lists the summary of state-of-the-art object detection.
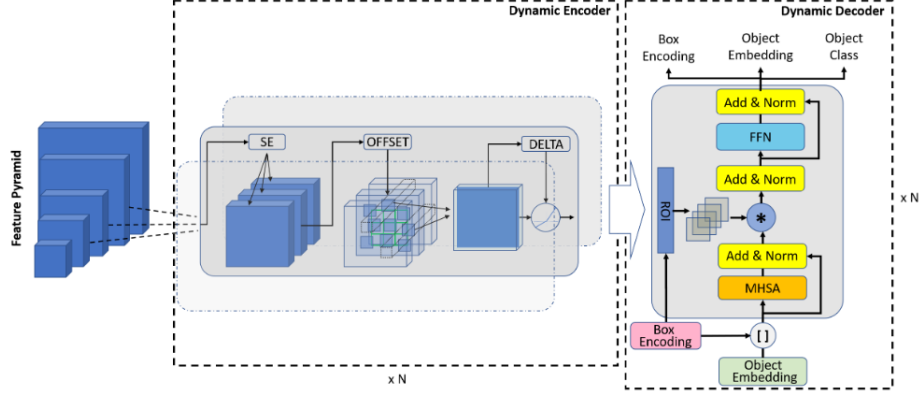
Figure 23: The architecture of dynamic DeTR [128].

4.1.3. Image Segmentation

Image segmentation is another important task in which deep learning has a significant impact. One of the earliest deep learning models for image segmentation is the fully convolutional network [131]. A fully convolutional network consists of only convolutional layers which accepts an input of an arbitrary size and produce the predicted segmentation map of the same size. The authors adopted the AlexNet, VGG16 and GoogleNet, replace their fully connected layers with convolutional layers and append a 1×1 convolutional layer, followed by bilinear up-sampling to match the size of the input. The model was considered a significant milestone in image segmentation, demonstrating the feasibility of deep learning for semantic segmentation trained in end-to-end manner. Deconvolution network is another popular deep learning model for semantic segmentation [132]. The model architecture consists of two parts: encoder and decoder. The encoder takes an input image and uses the convolutional layers to generate the feature maps. The feature maps are fed to the decoder composed of un-sampling and deconvolutional layers to predict the segmentation map. SegNet is another encoder-decoder model

41

Table 3: Summary of state-of-the-art object detection.

Ref.	Description	Results (Datasets and Metrics)
[123], 2023	This study introduced an adaptive NMS method that dynamically adjusts suppression criteria based on intersection over union (IoU) values, applying an additional iteration to improve the distinction between closely positioned objects and preserve high-confidence bounding boxes, reducing false positives	Rumex weeds IoU: 91.2, mAP: 90
[124], 2024	This study proposed a learning-based NMS that integrates the NMS process into the model's learning framework through a novel NMS-aware loss function, which adjusts bounding box weights using IoU to reduce false positives, while introducing a regression-assisted classification branch to enhance classification by leveraging bounding box prediction relationships	CrowdHuman mAP: 90.1
[125], 2024	This study proposed a similarity distance metric to evaluate bounding box similarity by considering location and shape, dynamically adapting to different datasets and object sizes, automatically learning hyperparameters to improve label assignment, and enhancing detection performance for tiny objects	AI-TOD mAP: 26.5, AP50: 57.7, AP75: 20.5 SODA-D mAP: 32.8, AP50: 59.4, AP75: 31.3 VisDrone2019 mAP: 28.7, AP50: 50.3
[128], 2021	This study introduced Dynamic DETR, which enhances the original DETR by introducing dynamic attention mechanisms in both the encoder and decoder stages. This approach addresses limitations related to small feature resolution and slow training convergence, resulting in improved performance and efficiency in object detection tasks	COCO2017 mAP: 49.3, AP50: 68.4, AP75: 53.6
[129], 2023	This study proposed a method that combines a one-to-one matching branch with an auxiliary one-to-many matching branch during training to improve the detection accuracy of DETR	COCO2017 mAP: 59.4, AP50: 77.8, AP75: 65.4
[130], 2023	This study proposed a training scheme to enhances DETR by leveraging predicted bounding boxes from other teacher models including RCNN-based and DETR-based detectors using knowledge distillation. The method does not introduce extra parameters or computational overhead during inference	COCO2017 mAP: 58.5, AP50: 77.4, AP75: 64.8

for semantic segmentation [133]. The encoder is a sequence of convolutional (with ReLU) and maximum pooling blocks which is analogous to a convolutional neural network. The decoder is composed of up-sampling layers which up-samples the inputs using the memorized pooled indices generated in the encoder phase, and convolutional layers without non-linearity. The encoder progressively reduces the resolution of the input data while extracting abstract features through a series of convolutional and pooling layers. This process causes the loss of fine-grained information, degrading the overall performance of segmentation. LinkNet mitigates this limitation by passing the feature maps at several stages generated by the encoder to the decoder, hence reducing information loss [134]. The model architecture of LinkNet is similar to SegNet, but utilizes ResNet as the encoder.

While Faster R-CNN is a significant approach in object detection task, it has been extended to perform instance segmentation task. One such extension is Mask R-CNN which is based on Faster R-CNN, introduces an additional branch for predicting the segmentation mask [135]. Similar to Faster R-CNN, Mask R-CNN utilizes the RPN to generate region proposals and then the region of interest alignment is applied to extract more accurate features from the proposed regions. Mask R-CNN does not leverage the multiscale feature representation which may degrade the overall performance of segmentation. To overcome this limitation, Path Aggregation Network (PANet) incorporates the FPN and introduces a bottom-up pathway to facilitate the propagation of the low-level information [136]. The pathway takes the feature maps of the previous stage as input and performs 3×3 convolution with stride 2 to reduce the spatial size of the feature maps. The generated feature maps are then fused with the feature maps from the FPN through the lateral connection. The model adopts the three branches as in Mask R-CNN. MaskLab is an instance segmentation model based on Faster R-CNN, consisting of object detection, segmentation, and instance (object) center direction prediction branches [137]. The direction prediction provides useful information to distinguish instances of the same semantic label, allowing the model to further refine the instance segmentation results.

Attention mechanisms have been integrated into the segmentation models to learn the weights of multiscale features at each pixel location. A multistage context refinement network introduces a context attention refinement module that is composed of two parts, context feature extraction and context feature refinement [138]. The context feature extraction captures both local and global context information, fuses both contextual information and passes

it to the context feature refinement while the context feature refinement removes redundant information and generates a refined feature representation, improving the utilization of contextual information. The context attention is added to the skip connection between the encoder and the decoder. Handcrafted features are often abandoned for automatic feature extraction using convolutional networks. However, it is argued that the interpretability and domain-specific knowledge embedded in handcrafted features can provide valuable insights. To this end, an attention module based on the covariance statistic is introduced to model the dependencies between local and global context of the input image [139]. Two types of attention are introduced: spatial covariance attention focuses on the spatial distribution and channel covariance attention attends to the important channels. Furthermore, the covariance attention does not require feature shape conversion, hence significantly reducing the space and time complexity of the model.

The convolutional layers use local receptive fields to process input data which can be effective for exploiting spatial patterns and hierarchical features but may find it difficult to capture global relationships across the entire image. The ViT has been leveraged to mitigate this issue in semantic segmentation [140]. Specifically, the input image is divided into patches and treated as input to the transformer to capture the long-range relationship between the patches, significantly improving the prediction of the segmentation map. Global context ViT aims to address the lack of ViT's ability to leverage local spatial features [141]. As shown in Figure 24, the transformer consists of local and global self-attention modules. The role of global self-attention is to capture the global contextual information from different image regions while the short-range information is captured by the local self-attention. Multi-scale feature representation is crucial for accurate semantic segmentation. However, the transformer often combines the features without considering their appropriate (optimal) scales, thus affecting the segmentation accuracy. Transformer scale gate is a module proposed to address the issue of selecting an appropriate scale based on the correlation between patch-query pairs [142]. The transformer takes attention (correlation) maps as input and calculates the weights of the multiscale features for each image patch, allowing the model to adaptively choose the optimal scale for each patch. Table 4 lists the summary of state-of-the-art image segmentation.

Table 4: Summary of state-of-the-art image segmentation.

| Ref.	Description	Results (Datasets and Metrics)
[138], 2023	The study proposed a multistage context refinement network with a context attention refinement module that improves feature representation by capturing local and global context, fusing information, and removing redundancies, with attention integrated into the encoder-decoder skip connections	PASCAL VOC2012 mIoU: 79.05 ADE20K MI mIoU: 41.25 Cityscapes mIoU: 79.42
[139], 2022	This study introduced a covariance attention to model the local and global dependency for the feature maps by formulating a covariance matrix, providing complementary information to improve segmentation performance	Cityscapes mIoU: 82.9 Pascal-Context mIoU: 51.7 ADE20K mIoU: 43.78
[140], 2021	This study introduced a transformer for semantic segmentation	Cityscapes mIoU: 81.3 Pascal-Context mIoU: 59.0 ADE20K mIoU: 53.63
[141], 2023	This study proposed an improved vision transformer architecture that employs alternating global and local self-attentions to effectively capture both local and global spatial information	ADE20K mIoU: 49.2
[142], 2023	This study introduced TSG module that uses attention maps to calculate multiscale feature weights for each image patch, enabling the model to adaptively select the optimal scale based on patch-query correlation	Cityscapes mIoU: 83.6 Pascal-Context mIoU: 64.9 ADE20K mIoU: 56.93
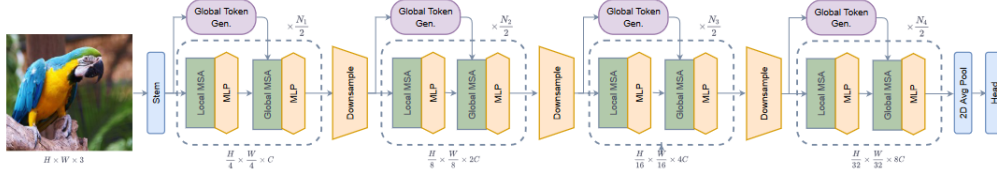
Figure 24: The architecture of global context ViT [141].

4.1.4. Image Generation

Image generation refers to the process of creating images based on input texts. Generally, the task can be divided into three stages. The first stage is extracting features from the input text, followed by generating the image and finally controlling the image generation process to ensure the output meets specific criteria and constraints. This section focuses on the progress made in the development of deep learning models of the second stage (image generation) since it directly impacts the quality of the generated images. Variational autoencoder is one of the earliest deep learning models that is capable of generating images [84]. Variational autoencoder learns to generate data by capturing the underlying (Gaussian) distribution of the training data. During the generation process, the distribution parameters are sampled and passed to the decoder to generate the output image. Although the generated images are blurry and unsatisfactory, it has shown a lot of potential in image generation tasks. The introduction of GAN significantly improved the quality of generated images. GAN consists of two connected neural networks, a generator and a discriminator that are trained simultaneously in a competitive manner [85]. The generator learns to generate realistic images to fool the discriminator, while the discriminator learns to distinguish between fake and real images. The generated images are less blurry and more realistic. Several enhanced models have been proposed to improve its usability and overall performance such as conditional GAN [86] which allows us to tell what image to be generated, and the deep convolutional GAN (DCGAN) [143] which provides a more stable structure for image generation. DCGAN is the basis of many subsequent improvements in GANs.

StackGAN divides the process of image generation into two stages [144]. Stage-I generates a low-resolution image by creating basic shapes and colors and the background layout using the random noise vector. Stage-II completes the details of the image and produces a high-resolution photo-realistic image. StackGAN++ is the enhanced model of StackGAN whereby it con-

sists of multiple generators with shared parameters to generate multiscale images [145]. The generators have a progressive goal with the intermediate generators generating images of varying sizes and the deepest generator producing the photo-realistic image. HDGAN is a generative model featuring a single-stream generator with hierarchically nested discriminators at intermediate layers [146]. These layers, each connected to a discriminator, generate multiscale images. The lower resolution outputs are used to learn semantic image structures while the higher resolution outputs are used to learn fine-grained details of the image. StackGAN heavily relies on the quality of the generated image in Stage-I. DM-GAN incorporates a memory network for image refinement to cope with badly generated images in Stage-I [147]. The memory network dynamically selects the words that are relevant to the generated image, and then refines the details to produce better photo-realistic images.

AttnGAN is the first to incorporate attention mechanisms into the multiple generators to focus on words that are relevant to the generated image [148]. To this end, in addition to encoding the whole sentence into a global sentence vector, the text encoder encodes each word into a word vector as shown in Figure 25. Then, the image vector is used to attend to the word vector using the attention modules at each stage of the multistage generators. Furthermore, AttnGAN introduces a loss function to compute the similarity between the generated image and the associated sentence, improving the performance of image generation. A similar work is reported whereby the model known as ResFPA-GAN, incorporates attention modules into the multiple generators [149]. Specifically, a feature pyramid attention module is proposed to capture high semantic information and fuse the multiscale feature, enhancing the overall performance of the model. DualAttn-GAN improves AttnGAN by incorporating visual attention modules to focus on important features along both spatial and channel dimensions [150]. This allows the model to better understand and capture both the context of the input sentence and the fine details of the image, resulting in more realistic image generation.

Although multistage generators improve image generation performance by leveraging multiscale representation, the generated images may contain fuzzy shapes with coarse features. DF-GAN replaces the multistage generators with a single-stage deep generator featuring residual connections and trained with hinge loss [151]. Furthermore, DF-GAN introduces a regularization strategy on the discriminator that applies a gradient penalty on real im-

47
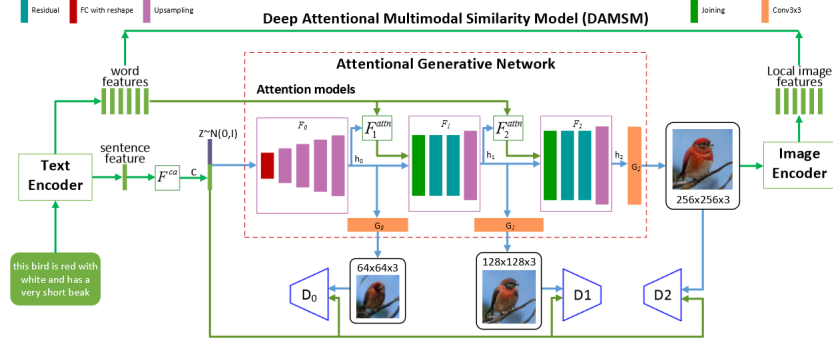
Figure 25: The architecture of AttnGAN [148].

ages with matching text, allowing the model to generate more text-matching images. DMF-GAN an improved DF-GAN, incorporates three novel components designed to leverage semantic coherence between the input text and the generated image [152]. The first component is the recurrent semantic fusion module, which models long-range dependencies between the fusion blocks. The second component is the multi-head attention module, which is placed towards the end of the generator to leverage the word features, forcing the generator to generate images conditioned on the relevant words. The last component is the word-level discriminator, which provides fine-grained feedback to the generator, facilitating the learning process and improving the overall quality of the generated images. Figure 26 shows the architecture of DMF-GAN. The process of image generation involves feeding a noise vector to the generator at the very beginning of the network. However, as the generator goes deeper, the noise effect may be diminished, affecting the diversity of the image generation results. To mitigate this issue, DE-GAN incorporates a dual injection module into the single-stage generator [153]. The dual injection module consists of two text fusion layers followed by a noise broadcast operation. The text fusion layer takes the sentence embedding and fuses it with the input feature map using the fully connected layer. Then noise is injected into the output feature map to retain the randomness in the generation process, improving diversity and generalization of the model. Table 5 lists the summary of state-of-the-art image generation.
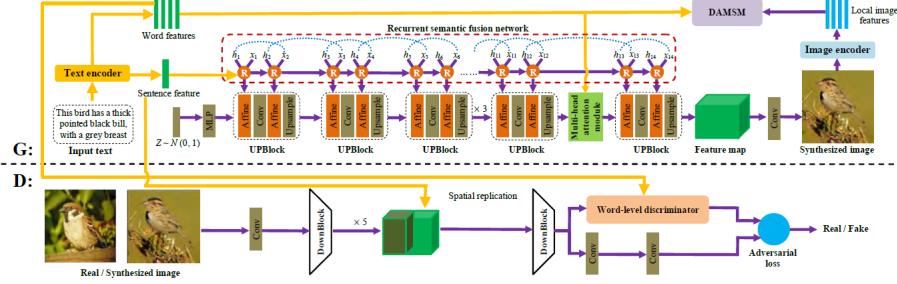
Figure 26: The architecture of DMF-GAN [152].

Table 5: Summary of state-of-the-art image generation.

Ref.	Description	Results (Datasets and Metrics)
[151], 2022	This study introduced a one-stage GAN architecture, employing a Target-Aware Discriminator to enhance text-image semantic consistency and utilizes deep text-image fusion blocks for effective feature integration, resulting in improved image authenticity and relevance	CUB IS: 5.10, FID: 14.81 COCO FID: 19.32
[152], 2024	This study proposed a deep multimodal fusion generative adversarial networks that allow effective semantic interactions for fusing text information into the image synthesis process	CUB IS: 5.42, FID: 13.21 COCO IS: 36.72, FID: 15.83
[153], 2024	This study proposed a deep learning-based method which utilizes BERT to extract sentiment information and analyze sentiment at the sentence and word level with SVM for sentiment classification	CUB IS: 4.86, FID: 18.94 COCO IS: 18.33, FID: 28.79

4.2. Natural Language Processing

Natural language processing (NLP) refers to the field of AI that concerns with enabling computers to process, analyze and interpret human languages to extract useful information. Some of the common tasks in NLP are machine translation, text classification and text generation. Deep learning has been widely applied to solve real-world NLP problems. This section presents the recent advancements in deep learning models that have been designed for NLP over the past few years.

4.2.1. Text Classification

Text classification, known as text categorization, is a task that involves assigning predefined categories or labels to a piece of text based on its content. The task is commonly used in various applications such as document classification, sentiment analysis and spam filtering. Numerous deep learning models have been proposed for text classification in the past few decades, and

multilayer perceptron is one of the earliest architectures adopted to classify documents [154, 155]. The model typically has a single hidden layer with a number of units between 15 and 150. Text data is inherently sequential, as it is composed of a series of words and symbols arranged in a specific order. This property makes RNN and its variants particularly well-suited for processing and analyzing text data. In [156], RNN with two hidden layers, each with 6 units, is used to classify news documents into eight classes. A study was conducted to investigate the variants of RNN such as LSTM and GRU for text classification [157]. The input to the model is a sequence of words of fixed length. The input sequence is also sliced into smaller subsequences of fixed length and passed to an independent model for parallelization. A convolutional layer can extract local features, allowing the model to leverage hierarchical temporal information in textual data. A hybrid model of convolutional and LSTM architecture is proposed for text classification [158]. Two parallel convolutional layers are used to extract features from word embeddings, followed by maximum pooling layers to reduce the feature dimensions. The reduced features are then concatenated and passed to LSTM for prediction.

Although CNN and RNN provide excellent results on text classification tasks, the models lack the ability to attend to specific words based on their importance and context. To address this limitation, an attention mechanism is incorporated into the model to focus on the important features, enhancing the text classification accuracy. In [159], two attention modules are introduced to capture the contextual information of the feature sequence extracted by bidirectional LSTM. The first attention module attends the sequence in forward direction, while the backward sequence is attended by the second attention module. The convolutional layers are used before the bidirectional LSTM to extract features from the word embedding. The attention modules require sequential processing using RNN-based architecture such as LSTM and GRU, which may lead to information loss and distorted representations, particularly in long sequence. Furthermore, the attention modules focus on inter-sequence relationships between the input sequence and the target, ignoring the intra-sequence relationships or the dependencies between the words. In [160], the deep learning model is integrated with self-attention to capture the intra-sequence relationships between the features in the sequence. A multilayer of bidirectional LSTMs is utilized to extract the feature sequence from the word embedding before the self-attention module attends the feature sequence to compute the attention weights. To further

improve the overall performance, a multichannel features consisting of three input pipelines is introduced [161]. Each pipeline concatenates the word vector with a feature vector derived from the input sequence such as the word position, part-of-speech and word dependency parsing. The input pipeline is connected to bidirectional LSTM, followed by a self-attention module to learn the dependencies between the features in the sequence.

The transformer is a deep learning architecture that transforms sequential data using self-attention mechanisms, allowing long-range dependencies and complex patterns to be captured. The architecture is the basis of various advanced deep learning models and the Bidirectional Encoder Representations from Transformers popularly known as BERT is one of the examples that leverage transformer for pre-training on large scale textual data [162]. BERT is a bidirectional transformer encoder that is designed for various NLP tasks, capable of capturing the contextual information from both preceding and succeeding words in the input sequence. Several improvements have been made to BERT to enhance its overall performance, such as ALBERT [163], RoBERTa [164] and DeBERTa [165]. The improvements are centered around refining the pre-training approaches such as dynamic masking of the training instances, training with a block of sentences and representing each input word using two vectors, both content and position of the word. Most of the recent works leverage BERT and its variants to capture effective feature representation of the input sequence. In [166], BERT and its variants are leveraged to capture the long-range dependencies of the input tokens. The features are then passed to a layer normalization and a linear fully connected layer with dropout for classification. Similar work is reported in [167] whereby BERT is used to extract the features and the features are then passed to a hybrid of convolutional and recurrent neural networks. The traditional machine learning algorithms have been used to classify the features extracted by BERT [168]. The study shows machine learning algorithms can effectively leverage the rich contextual features extracted by BERT for downstream classification tasks.

In text classification, the text labels can help in capturing the words relevant to the classification. The label-embedding attentive model is one of the earliest attempts to joint learn the label and word embeddings in the same latent space and measure the compatibility between labels and words using cosine similarity [169]. The joint embedding allows the model to capture more effective text representations, increasing the overall performance of the model. LANTRN is a deep learning model that leverages label em-

bedding extracted by BERT and entity information e.g. person name and organization name for text classification [170]. The entity recognition module is based on bidirectional LSTM and conditional random field layers to calculate the probability of each word in each entity label. The model introduces a label embedding bidirectional attention to learn the attention weights of token-label and sequence-label pairs. Furthermore, a transformer consisting of RNN and a multi-head self-attention mechanism is introduced to learn local short-term dependencies of multiple short text sequences and long-term dependencies of the input sequence. Aspect refers to a specific attribute of an entity within the text, and incorporating this information enhances the model's understanding of the nuances of the text. BERT-MSL is a multi-semantic deep learning model with aspect-aware enhancement and four input pipelines: left sequence, right sequence, global sequence and aspect target [171]. The aspect-aware enhancement module takes the features extracted by BERT, and performs average pooling followed by a linear transform. Then the output is concatenated with the outputs produced by the local and global semantic learning modules. The concatenated features are then jointly attended by a multi-head attention for text classification. Figure 27 shows the architecture of BERT-MSL. Table 6 lists the summary of state-of-the-art text classification.

4.2.2. Neural Machine Translation

Neural machine translation (NMT) refers to the automated process of translating text from one language to another language. Numerous deep learning models have been proposed for NMT, which can be categorized into RNN-based and CNN-based models. One of the first successful RNN-based models is the encoder-decoder [49, 172]. The model consists of two connected subnetworks (the encoder and the decoder) for modelling the translation process, as shown in Figure 28. The encoder reads the source sentence word by word and produces a fixed-length context vector (final hidden state). This process is known as source sentence encoding, as shown in the figure. Given the context vector, the decoder generates the target sentence (translation) word by word. This modelling of the translation can be seen as a mapping between the source sentence to the target sentence via the intermediate context vector in the semantic space. The context vector represents the summary of the input sequence's semantic meaning, providing a compressed

Table 6: Summary of state-of-the-art text classification.

Ref.	Description	Results (Datasets and Metrics)
[166], 2022	The study proposed an enhanced RoBERTa by incorporating layer normalization and dropout to enable smoother gradients, faster training and reduced overfitting, thus, improving model generalization	NEISS Accuracy: 0.98
[167], 2024	The study employed IndoBert to extract features from text data which were then passed to a hybrid deep learning model for sentiment analysis	Shopee Accuracy: 0.8612 (CNN-GRU) Tokopedia Accuracy: 0.8768 (LSTM-CNN) Lazada Accuracy: 0.8710 (LSTM-CNN)
[168], 2023	This study proposed a deep learning-based method which utilizes BERT to extract sentiment information and analyze sentiment at the sentence and word level with SVM for sentiment classification	NLPCC-SCDL Accuracy: 95.12%
[170], 2023	This study proposed LANRTN model for text classification, integrating an R-Transformer with label embedding, attention mechanisms, and an entity recognition model to capture both global dependencies and label-aware contextual information	Reuters Corpus Volume I Micro F1-score: 0.893, Arxiv Academic Paper Dataset Micro F1-score: 0.718
[171], 2023	The paper proposed a BERT-MSL model for aspect-based sentiment analysis, integrating multiple semantic learning modules and merging them using multi-head self-attention and linear transformation layers for enhanced classification	14Lap Accuracy: 82.24%, Macro-average F1-score: 78.98% 14Rest Accuracy: 89.11%, F1-score: 84.07% 15Rest Accuracy: 88.53%, Macro-average F1-score: 71.41% 16Rest Accuracy: 93.09%, Macro-average F1-score: 81.24% Twitter Accuracy: 74.89%, Macro-average F1-score: 73.43%
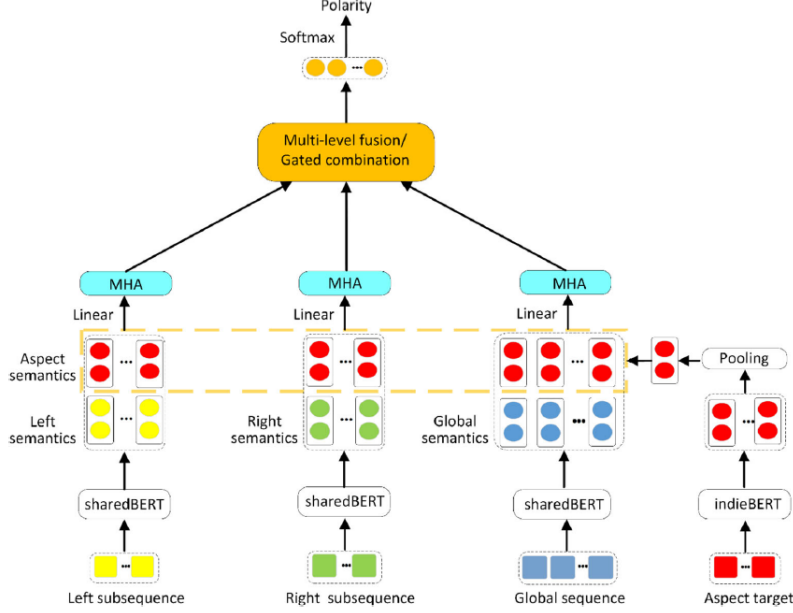
Figure 27: The architecture of BERT-MSL [171].

representation that captures the essence of the source sentence. However, the compression process can sometimes result in the loss of information, especially those early in the sequence. Bidirectional RNN may mitigate the loss of information by modelling the sequence in reverse order. However, the problem can persist, particularly in cases where the input is a long sequence.
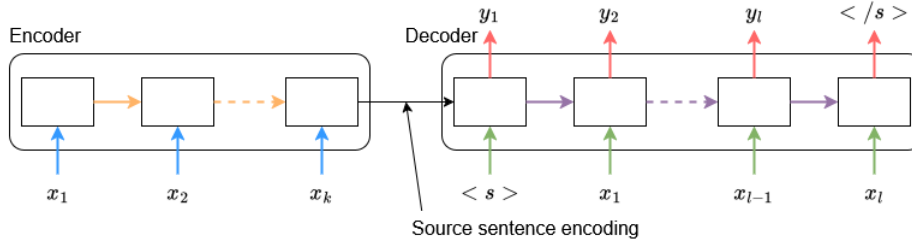
Figure 28: The architecture of an encoder-decoder. Adapted from [173].

Attention mechanism was introduced to solve the problem of learning long input sequences [174]. Attention alleviates this issue by attending on different words of the input sequences when predicting the target sequences

at each time step. Unlike the standard encoder-decoder model, attention derives the context vector from the hidden states of both the encoder and decoder, and the alignment between the source and target. This mechanism allows the model to focus on the important words, increasing the overall accuracy of the translation. Several alignment score functions have been proposed for calculating the attention weights. Some popular functions are additive [174], dot-product, location-based [175], and scaled dot-product [18]. The attention weights are calculated by attending to the entire hidden states of the encoder. This attention, also known as global attention, is computationally expensive. Instead of attending to all hidden states, local attention attends to a subset of hidden states, thus reducing the computational cost [175]. Google Neural Machine Translation is a popular encoder-decoder model with an attention mechanism that significantly improves the accuracy of machine translation [176]. As shown in Figure 29, the model consists of a multilayer of LSTMs with eight encoder and decoder layers and an attention connection between the bottom layer of the decoder to the top layer of the encoder. Furthermore, to deal with the challenging words to predict, a word is tokenized into subwords e.g. feud is broken down into "fe" and "ud", allowing the model to generalize well to new and uncommon words. A year later, the self-attention mechanism was proposed, significantly improving the overall accuracy of machine translation [18]. Self-attention, also known as intra-attention, allows the deep learning model to capture the dependencies between the input words. The self-attention mechanism is the fundamental building block of the transformer model, which has since become a cornerstone in natural language processing and other domains.

Despite the success of transformer, the model falls short in capturing nuances of human language and struggles with tasks requiring deeper understanding of context. This can be especially challenging when the tasks involve formality, colloquialism, and subtle cultural references that may not directly equivalent in the target language, resulting in inaccurate translation or losing the original meaning. One of the approaches to include context into the input sequence is concatenating the current source sentence with the previous (context) sentences and feeding the whole input to the transformer [177]. The model is trained to predict the translated sentence including the context translation. At inference time, only the translation is considered while the context translation is discarded. Furthermore, the approach encodes the sentence position and segment-shifted position to improve the distinction between current sentences and context sentences. In [178], the source sentence
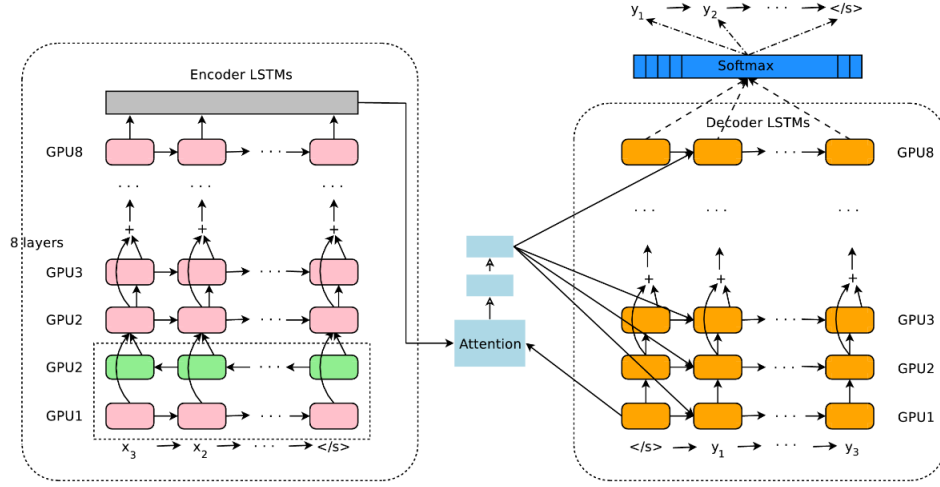
Figure 29: The architecture of Google Neural Machine Translation [176].

is prefixed with the summary of the document to contextualize the input sentence. The summary is the set of salient words that represents the essence of the document, resolving ambiguity associated with the translation. A study was conducted to determine the optimal technique of aggregating contextual features [179]. Three techniques were studied namely concatenation mode, flat mode and hierarchical mode, and the experimental results indicate that concatenation mode achieved the best results. In [180], a training method is introduced to train the deep learning machine translation model to generate translation involving honorific words. The training method indicates the honorific context in the target sentence using an honorific classifier to guide the model to attend to the related tokens. Unlike other studies where the context features are included by concatenation, the training method assigns weights to the context tokens indicated by the honorific classifier. This allows the model to generate a more accurate translation with honorifics.

The performance of transformers relies on large-scale training data. However, for the vast majority of languages, only limited amounts of training data exist. To mitigate this problem, recent studies introduce shallow transformer architectures [181], explore the effect of hyperparameter finetuning [182] and leveraging visual input as contextual information for the translation task [183]. The semi-supervised neural machine translation is used for translating low resource Egyptian Arabic dialects to modern standard

56

Arabic [184]. The study utilizes three datasets: bilingual Egyptian-standard Arabic and two monolingual Egyptian Arabic and standard Arabic. First, a transformer-based model is trained in a supervised manner using the bilingual dataset. Then, the model is trained in an unsupervised manner using both monolingual datasets. The unsupervised approach employs an Encoder-Decoder model with Byte Pair Encoding for tokenization and handling unknown words. The monolingual corpora are merged and used to improve the model by iteratively generating synthetic sentence pairs between Egyptian Arabic and standard Arabic, allowing the model to learn the correspondence between them. In [185], the authors exploit monolingual corpus to enhance the bilingual dataset for model training. Furthermore, a new loss function is proposed as a replacement for traditional cross-entropy loss, allowing the model to learn with uncertainty in the presence of noise. Additionally, contrastive re-ranking is employed to refine translation results by selecting the most confident output from multiple candidates. Table 7 lists the summary of state-of-the-art neural machine translation.

4.2.3. Text Generation

Text generation refers to the process of creating texts based on a given input, whereby the input can be in the form of texts, images, graphs, tables or even tabular data. Due to the various forms of inputs, text generation has a wide range of applications, including creative writing, image captioning and music generation. This section focuses on the progress made in text-to-text generation tasks such as question answering, dialogue generation and text summarization. The recurrent neural network and its variants play an important role in text generation tasks for their strong ability to model sequential data. One of the earliest works on question answering is based on the RNN-based encoder-decoder model, whereby the encoder takes the question embedding and processes it using bidirectional LSTM, and the decoder generates the corresponding answer [186]. Additionally, to prevent semantic loss and enable the model to focus on the important words in the input sequence, a convolution operation is applied to the word embedding, and an attention mechanism is then used to attend to the output of the convolution operation. Similar work is reported in [187] in which a knowledge-based module is introduced to calculate the relevance score between the question and the relevant facts in the knowledge base. This improves the text (answer) generation by the decoder. Another work is described in [188] where

Table 7: Summary of state-of-the-art neural machine translation.

Ref.	Description	Results (Datasets and Metrics)
[177], 2023	This study incorporated context into the input sequence by concatenating the current source sentence with previous context sentences, and the model is trained to predict the translation with context, while encoding sentence positions to distinguish between current and context sentences	En→De ContraPro: 82.54 En→Ru Voita: 75.94
[178], 2023	The paper proposed a method to enhance translation accuracy of ambiguous words by prefixing source sentences with salient words extracted from related pseudo-documents, thereby providing additional context without altering standard model architectures	En→De BLEU: 22.0, COMET: 0.785
[179], 2022	The paper investigated three most common methods to aggregate the contextual features: concatenation mode, flat mode and hierarchical mode	En→De BLEU: 30.89 De→En BLEU: 36.84 En→Zh BLEU: 20.06 Zh→En BLEU: 20.50
[180], 2023	The paper introduced formality classifier to incorporate formality-related contexts into the model training	AI-HUB (En→Ko) BLEU: 27.90 OpenSubtitles201 (En→Ko) BLEU: 20.70 BSD, AMI and OpenSubttles2018 (En→Ja) BLEU: 13.67
[184], 2024	This study employed a semi-supervised neural machine translation approach to translate low-resource Egyptian Arabic dialects to Modern Standard Arabic, utilizing bilingual and monolingual datasets, with a transformer-based model trained in both supervised and unsupervised manners to generate synthetic sentence pairs and improve translation accuracy	Egyptian-standard Arabic BLEU: 24 (300 word embedding) BLEU: 29.5 (512 word embedding)
[185], 2024	The study proposed a new loss function to handle uncertainty in noisy data, and used contrastive re-ranking to select the most confident translation from multiple candidates	Zh→Ma BLEU: 28.12 Ma→Zh BLEU: 23.53 Zh→In BLEU: 28.91 In→Zh BLEU: 22.76

an encoder-decoder with attention for dialogue generation is optimized using reinforcement learning. The model is first trained in a supervised learning manner and then improved using the policy gradient method to diversify the responses. Ambiguous content in question answering sentences is a challenge in text generation and can lead to incorrect and uncertain responses. Cross-sentence context aware bidirectional model introduces a parallel attention module to compute the co-attention weights at the sentence level, accounting for the relationships and similarities in the question and the answer [189].

The transformer has been leveraged for text generation tasks. An incremental transformer-based encoder is proposed to incrementally encode the historical sequence of conversations [190]. The decoder is a two-pass decoder that is based on the deliberation network, generates the next sentence. The first pass focuses on contextual coherence of the conversations, while the second pass refines the output of the first pass. BERT and ALBERT have been used as pre-trained models for question answering task [191]. The study found that the performance of the models is sensitive to random assignment of the initial weights especially on small datasets [192]. T-BERTSum is a model based on BERT, designed to address the challenge of long text dependence and leveraging latent topic mapping in text summarization [193]. The model integrates a neural topic module to infer topics and guide summarization, uses a transformer network to capture long-range dependencies and incorporates a multilayer of LSTM for information filtering.

The generated texts often lack diversity and may exhibit repetitive patterns. To address this issue, feature-aware conditional GAN (FA-GAN) is proposed for controllable category text generation [194]. The generator consists of BERT, a category encoder, a relational memory core (RMC) decoder. BERT acts as a feature encoder, improves contextual representation and mitigates mode collapse while the category encoder embeds categorical information for text generation. The RMC decoder utilizes a self-attention mechanism to capture interactions between features, generating more expressive and diverse texts. The discriminator includes an additional classification head to ensure the generated texts match specified categories. Traditional text generation has largely focused on binary style transfers, but real-world applications require capturing diverse styles, which existing methods struggle to achieve. In [195], a multi-class conditioned text generation model is proposed using a transformer-based decoder with an adversarial module, a style attention module and a generation module. The adversarial module extracts style-excluded representation from the input text. The style attention

mechanism then introduces the desired style into the text representation through concat attention mechanism, producing a conditioned representation. Finally, the generation module utilizes the conditioned representation to generate stylistically diverse text.

Exploiting domain knowledge is essential in reducing the semantic gap between the deep learning models and the text corpus. KeBioSum is a knowledge infusion framework to inject domain knowledge into the pre-trained BERTs for text summarization [196]. In the framework, the relevant information is detected and extracted from the domain knowledge, generating label sequences of the sentences. The label data is then used to train the text summarization model using discriminative and generative training approaches, infusing the knowledge into the model. Large language models (LLMs) have been used for text generation with exceptional quality and diversity. LLMs, trained on extensive corpus data, have a deep understanding of human language, allowing them to interpret and generate texts. In [197], GPTGAN is introduced that leverages an LLM as a guiding mentor to a GAN-Autoencoder model for text generation. The approach involves the GPT model to generate a sequence of words given a subset of the input sequence, and the generated text is mapped into latent space by the transformer-based encoder. The latent representation is then used by the transformer-based decoder to generate the text. Furthermore, local discriminators are introduced to refine the text generation because the generated words by the GPT model may be inaccurate. Table 8 lists the summary of state-of-the-art text generation.

4.3. Time Series and Pervasive Computing

Pervasive computing, often referred to as ubiquitous computing, is the process of integrating computer technology into everyday objects and surroundings so that they become intelligent, networked, and able to communicate with one another to offer improved services and functionalities [198]. According to He et al. [199], the role of pervasive computing is foremost in the field where it provides the ability to distribute computational services to the surroundings where people work, leading to trust, privacy, and identity. Examples of pervasive computing applications include smart homes with connected appliances, wearable devices that monitor health and fitness, smart

Table 8: Summary of state-of-the-art text generation.

Ref.	Description	Results (Datasets and Metrics)
[193], 2021	The paper proposed a text summarization model that integrates neural topic model with BERT to guide the text summarization based on inferred topics. It leverages transformers for long-term dependencies and combines LSTM layers with a gated network to enhance extractive and abstractive summarization of social texts	CNN/Daily Mail Rouge-1: 43.58, Rouge-2: 20.45, Rouge-L: 34.60 Xsum Rouge-1: 39.90, Rouge-2: 17.48, Rouge-L: 29.85
[194], 2023	This study proposed FA-GAN featuring BERT for contextual representation, a category encoder for embedding categorical information, and an RMC decoder for generating diverse texts, while the discriminator ensures category consistency	MR-10 BLEU-2: 0.560 MR-20 BLEU-2: 0.674 AM-30 BLEU-2: 0.748
[195], 2024	This study proposed a multi-class conditioned text generation model, combining a transformer-based decoder with an adversarial module, style attention, and a generation module to produce stylistically diverse text	Amazon (neg→pos) self-BLEU: 0.257, S-ACC: 0.375 Amazon (pos→neg) self-BLEU: 0.096, S-ACC: 0.191 YELP (neg→pos) self-BLEU: 0.257, S-ACC: 0.198 YELP (pos→neg) self-BLEU: 0.081, S-ACC: 0.207 *S-ACC (style accuracy)
[196], 2022	The paper proposes a knowledge-based summarization model that integrates medical knowledge into the pretrained models using a lightweight knowledge adapter. It employs generative and discriminative training to predict and reconstruct PICO elements, enhancing domain-specific text summarization	CORD-19 Rouge-1: 32.04, Rouge-2: 12.61, Rouge-L: 29.10 PubMed-Long Rouge-1: 36.39, Rouge-2: 16.27, Rouge-L: 33.28
[197], 2025	This study proposed GPTGAN, an approach that enhances adversarial text generation using a GPT model as a mentor to the GAN model and employing a multiscale discriminator framework to balance text quality and diversity	MSCOCO BLEU: 98.7 WMTNews BLEU: 93.4 Persian COCO BLEU: 97.9

cities with sensor networks for traffic management, and industrial applications that utilize the Internet of Things (IoT) for monitoring and control. Generally, the continuous interaction of interconnected devices in pervasive computing often results in time series data, which captures the evolution of various parameters over time.

For instance, medical sensors, such as electrocardiograms (ECG) and electroencephalograms (EEG), generate time series data that contain critical diagnostic information, which deep learning can use to detect anomalies, predict diseases, and classify medical conditions with improved accuracy. Furthermore, devices such as accelerometers, magnetometers and gyroscopes, among others, can be used to capture human activity signals, which are often represented as time series of state changes [200]. In traditional machine learning, features such as mean, variance, and others are manually extracted from times series of state changes before human activity classification. However, deep learning models automatically extract features [78]. Furthermore, in other fields such as finance, which entail time series data, deep learning has been instrumental in stock price prediction [201], fraud detection [202], and algorithmic trading [203], among others. Generally, deep learning networks excel at capturing intricate temporal relationships within time-series data, enabling more precise predictions and improved decision-making. Based on this, several deep learning models have been employed for feature learning across various time series and pervasive computing domains.

4.3.1. Human Activity Recognition

Human activity recognition (HAR) finds application across various domains including intelligent video surveillance, environmental home monitoring, video storage and retrieval, intelligent human-machine interfaces, and identity recognition, among many others. It includes various research fields, including the detection of humans in video, estimating human poses, tracking humans, and analyzing and understanding time series data [204]. Despite the advancements in vision-based HAR, there exist inherent limitations. Generally, vision-based approaches heavily rely on camera systems, which may have restricted views or be affected by lighting conditions, occlusions, and complex backgrounds [200]. Additionally, vision-based HAR struggles with identifying actions that occur beyond the range of the camera or actions that are visually similar.

Wearable sensors offer a promising alternative to overcome these limitations. By directly capturing data from the individual, wearable sensors

provide more comprehensive and accurate information about human activities. The signals obtained from wearable sensors typically represent time series data reflecting state changes in activities. Deep learning models can effectively learn from these signals, allowing for robust and accurate recognition of human activities. Moreover, wearable sensors offer the advantage of mobility, enabling activity recognition in various environments and situations where vision-based systems may be impractical or ineffective [205]. Generally, the time series nature of signals from wearable sensors presents an excellent opportunity for deep learning models to excel in recognizing human activities with high accuracy and reliability.

Several researchers have proposed the use of CNN, RNN, and Hybrid models for deep learning-based feature learning in wearable sensor HAR. For instance, using two-dimensional CNN (Conv2D), several researchers, as seen in [206], [207] and [208], among others, have developed deep learning models for wearable sensor HAR, despite the time series nature of the data. This is often done by treating the time series signals from wearable sensors as 2D images by reshaping them appropriately. To achieve this, researchers often organize each time series signal into a matrix format, with time along one axis and sensor dimensions along the other, before creating a pseudo-image representation, which allows the matrix to be fed into Conv2D layers for feature extraction. Conv2D layers excel at capturing spatial patterns and relationships within images, and by treating the time series data as images, these layers can learn relevant spatial features that contribute to activity recognition. The convolution operation performed by Conv2D filters across both the time and sensor dimensions, allowing the network to identify patterns and features that may be indicative of specific activities.

Even though Conv2D can effectively capture spatial dependencies within the data, it often struggles to capture temporal dependencies inherent in time series data. Since Conv2D processes data in a grid-like fashion, it does not fully leverage the sequential nature of the time series, potentially leading to less effective feature extraction for wearable sensor HAR tasks. For this reason, recent HAR architectures have leveraged one-dimensional CNN (Conv1D) and other RNNs for automatic feature extraction. Conv1D layers are specifically designed to capture temporal dependencies within sequential data. They operate directly on the time series data without reshaping it into a 2D format, allowing them to capture temporal patterns more effectively. Conv1D layers are better suited for extracting features from time series data, making them a more natural choice for wearable sensor HAR [78].

For instance, Ragab et al. [209] proposed a random search Conv1D model, and evaluated the performance of the model on UCI-HAR dataset. The result showed that the model achieved a recognition accuracy of 95.40% when classifying the six activities in the dataset. However, the model exhibited extended training times due to the dynamic nature of some activities within the dataset. To address this, [210] proposed the use of varying kernel sizes in Conv1D layers to recognize various activities, including sitting, standing, walking, sleeping, reading, and tilting. Furthermore, a few Conv1D layers were stacked to streamline the time optimization process for training the neural network. In [211], a signal segmentation method is proposed based on similarity between two subsequent windows. The method can not only achieve more accurate segmentation, but also distinguish between transitional and non-transitional windows. Based on the distinction, two deep learning models with convolutional and fully connected layers are trained: one for transitional activities and the other for non-transitional activities. In [212], signal segmentation is treated as a binary classification task, distinguishing windows as either transitional or non-transitional. To this end, a deep learning model with parallel Conv1D pipelines is proposed to capture temporal dependencies within the window sequence. Furthermore, some researchers have proposed models that combine machine learning algorithms with Conv1D in HAR, as seen in Shuvo et al. [213]. Their work presented a two-stage learning process to improve HAR by classifying activities into static and dynamic using Random Forest, before using Support Vector Machine to identify each static activity, and Conv1D to recognize dynamic activities. The result shows that the method achieved an accuracy of 97.71% on the UCI-HAR dataset.

Following these advancements, several researchers have further explored Conv1D architectures with various modifications, to enhance feature learning in activity recognition systems. For example, Han et al. [214] developed a two-stream CNN architecture as a plug-and-play module to encode contextual information of sensor time series from different receptive field sizes. The module was integrated into existing deep models for HAR at no extra computation cost. Experiments on OPPORTUNITY, PAMAP2, UCI-HAR and USC-HAD datasets show that the module improved feature learning capabilities. A similar research is reported in [215] proposed the WSense module to address the issue of differences in the quality of features learnt, regardless of the size of the sliding window segmentation, and experimented on PAMAP2 and WISDM datasets. The results show that by plugging the

WSense module into Conv1D architectures, improved activity features can be learned from wearable sensor data for human activity recognition.

Hybrid Models: Researchers have also proposed the use of standalone RNNs in HAR, and a hybrid of Conv1D architectures with RNNs such as LSTMs [216], BiLSTMs [217, 218], GRUs [219] and BiGRUs [220] to fully harness the feature learning capabilities of both CNN and RNNs. For instance, Nafea et al. [221], leveraged Bi-LSTM and Conv1D with increasing kernel sizes to learn features at various resolutions. Human activity features were extracted using the stacked convolutional layers with a Bi-LSTM layer, before including a flattening layer and a fully connected layer for subsequent classification. However, the model had issues extracting quality features of dynamic activities compared to static activities. To address such issues, some research works have incorporated attention mechanisms in Conv1D-based architectures to improve feature learning of dynamic and complex activities from time series signals obtained from wearable sensors. For example, Khan and Ahmad [222] designed three lightweight convolutional heads, with each specialized in feature extraction from wearable sensor data. Each head comprised stacked layers of Conv1Ds, along with embedded attention mechanisms to augment feature learning as shown in Figure 30. The results demonstrated that integrating multiple 1D-CNN heads with SE attention can enhance feature learning for Human Activity Recognition.

Ayo and Noor [223] designed three feature learning pipelines, each pipeline consisting of two concurrent layers of Conv1D and LSTM with maximum pooling, which are then concatenated and processed using a channel-wise attention mechanism to enhance feature learning. In [224], a sequential channel-temporal attention is proposed for multi-modal activity recognition. The channel attention is similar to SE attention, but with average pooling and maximum pooling pipelines to squeeze to temporal dimension. The pooled features are then combined through addition operation. As for the temporal attention, average pooling and maximum pooling are applied along the channel dimension, producing two pooled features which are then combined through concatenation. The attention module is incorporated into the residual blocks of ResNet-like model to improve feature extraction. Similar work is reported in [225] whereby a hybrid sequential channel-spatial attention is proposed for a lightweight activity recognition model. The lightweight model consists of four convolutional blocks integrated with the attention module and two LSTM layers. Although the proposed attentions have shown a strong potential in improving model performance, their sequential process-

ing inherently prioritize one of the feature map dimensions, impeding the model's ability to capture holistic feature representations. In [226], a triple cross-domain attention is proposed to blend three attention branches to improve feature extraction in HAR. For each attention branch, z-pooling [227] and convolution operations are applied along the corresponding dimension to generate the attention weights. Then, the feature maps are combined via average operation. The model consists of three residual blocks where each block is integrated with the attention module. These diverse modifications and adaptations showcase the versatility and potential of deep learning models in achieving state-of-the-art in HAR systems.
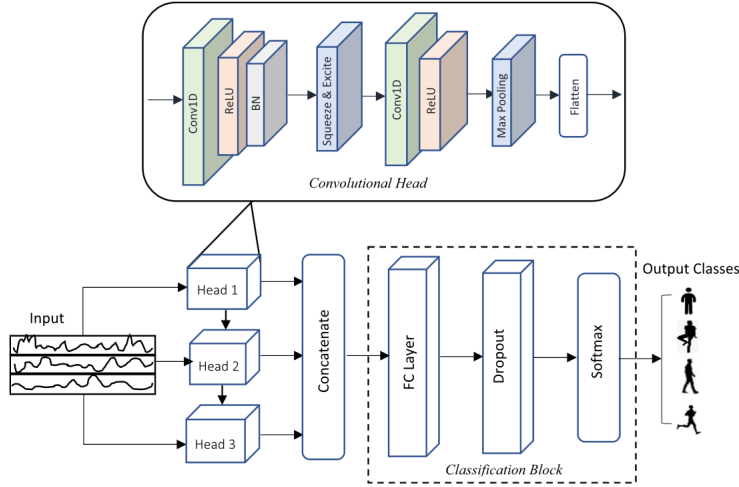
Figure 30: The architecture of multi-head CNN model [222].

Transformers: Transformers have been employed in human activity recognition to capture long-range dependencies in sensor data, enabling more accurate activity classification. In [228], a transformer-based model is enhanced by integrating a bidirectional GRU and a linear fully connected layer into each encoder block. The decoder is reduced to a fully connected layer with softmax function. This hybrid model leverages the transformer's ability to capture long-range dependencies in sensor data and the bidirectional GRU's strength in modeling temporal sequences, enhancing the model performance in classifying complex and collaborative activities. Similar work is reported in [229] whereby the encoder blocks of the transformer are integrated with 1D convolutional layers instead of fully connected layers, en-

66

abling the model to extract local features. Furthermore, self-supervised contrastive learning is employed to learn from unlabeled data before the model is fine-tuned using a smaller dataset. Although transformer-based models have shown remarkable performance, they suffer from high computational cost and memory requirements, making them less suitable for real-time or resource-constrained activity recognition. A study was conducted to determine the applicability of transformer models on a low-power ESP32 microcontroller [230]. The study concludes that transformer models are not suitable for tiny devices. Furthermore, the results show that a tiny transformer model with two encoder blocks achieved lower accuracy compared to the standard LSTM model, with a difference up to 14%. However, transformer models run 3x faster than LSTM, making them a viable option provided the architecture of transformers is optimized for low-power platforms.

Generative Models: A reliable deep learning model requires a large amount of training data to learn the underlying patterns of the data. However, in HAR, data collection is expensive, and the available datasets are often limited in the number of samples. To overcome this limitation, researchers employ GAN models to generate synthetic data, thus augmenting the training set. In [231], a conditional GAN is proposed to generate realistic sensor data of different activities for human activity classification. The generator consists of four 1D convolutional and maximum pooling blocks followed by two layers of LSTM and fully connected layers, while the discriminator is a standard CNN model. An enhanced conditional GAN is proposed to improve synthetic data generation for HAR [232]. This architecture integrates 1D convolutional layers with multiple fully connected networks at the generator's input and discriminator's output, aiming to produce higher-quality synthetic samples compared to existing CGAN models. Similar work is reported in [233], where the conditional Wasserstein GAN is employed to generate synthetic data for HAR.

In [234], sensor data is transformed into its frequency spectrum, forming an RGB-based feature space for human activity classification. Then, BiGAN is employed to generate synthetic frequency spectrum images to increase minority sample classes, thus diversifying the training set. The generator and discriminator of the proposed BiGAN are built using 2D convolutional and fully connected layers. In [235], a conditional GAN is employed to generate synthetic sensor data for therapeutic activity recognition. The generator utilizes Inception-like modules and transposed convolutional layers, while the discriminator consists of two classification pipelines: one classifies raw

Table 9: Summary of state-of-the-art human activity recognition.

Ref.	Description	Results (Datasets and Metrics)
[219], 2023	This study proposed an Inception inspired model, combining convolutional layers with GRU. The convolutional blocks use multiple-sized filters to extract multiscale feature representations	MHEALTH Accuracy: 99.25% PAMAP2 Accuracy: 97.64%
[223], 2023	This study proposed a model with three parallel feature learning pipelines, each pipeline has two sub-feature learning pipelines consisting of convolutional layers and bidirectional LSTM, and channel-wise attention before the classifier	PAMAP2 Accuracy: 98.52% WISDM Accuracy: 97.90%
[225], 2024	This study proposed A lightweight model consists four convolutional blocks integrated with a sequential channel-spatial module and two LSTM layers	PAMAP2 Accuracy: 98.52% WISDM Accuracy: 97.90%
[229], 2024	A modified transformer integrated with convolutional layer instead of fully connected layer	UCI-HAR Accuracy: 95.49% Skoda Accuracy: 87.88% Mhealth Accuracy: 98.43%
[233], 2024	This study employed a conditional Wasserstein GAN to generate synthetic human activity recognition accelerometry signals	Local Ulster University dataset Accuracy: 0.7453
[234], 2024	This study converted sensor data into its frequency spectrum to create an RGB-based feature space, employed BiGAN to generate synthetic frequency spectrum images to balance minority classes, and used 2D convolutional and fully connected layers in the generator and discriminator	Up-Fall Accuracy: 99.1% Opportunity Accuracy: 86.8% WISDM Accuracy: 99.12%
[235], 2025	This study employed a conditional GAN to generate synthetic sensor data for therapeutic activity recognition which has Inception-like generator and a discriminator with two classification pipelines, one for raw signals and the other for Fourier-transformed signals	Local Sharif University of Technology dataset Avg. F1-score: 0.897

signals and the other pipeline classifies the Fourier-transformed signal (frequency spectrum images). Both predictions are averaged to obtain the final prediction. Table 9 lists the summary of state-of-the-art human activity recognition.

4.3.2. Speech Recognition

Speech, as the primary mode of human communication, has captivated researchers for over five decades, especially since the inception of AI [236]. From the earliest endeavors to understand and replicate the complexities of human speech, to contemporary advancements leveraging cutting-edge technologies, the quest for accurate and efficient speech recognition systems has

been relentless. In recent years, the emergence of deep learning techniques has revolutionized the speech recognition field. Deep learning has demonstrated unparalleled success in processing and extracting intricate patterns from vast amount of data. When applied to the realm of speech recognition, deep learning have surpassed traditional approaches by learning intricate features directly from raw audio signals, circumventing the need for handcrafted features and complex preprocessing pipelines. This paradigm shift has significantly advanced the state-of-the-art in speech recognition, enabling systems to achieve unprecedented levels of accuracy and robustness across various languages, accents, and environmental conditions. Generally, deep learning has been extended to other essential applications of speech recognition, such as speaker identification [237, 238], emotion recognition [239], language identification [240], accent recognition [241], age recognition [242] and gender recognition [243], among many others.

Prior to the adoption of deep learning in speech recognition, the foundation of traditional speech recognition systems was the use of Gaussian Mixture Models (GMMs), which are often combined with Hidden Markov Models (HMMs) to represent speech signals [244]. This is because a speech signal can be thought of as a short-term stationary signal. The spectral representation of the sound wave is modelled by each HMM using a mixture of Gaussian. However, they are considered statistically inefficient for modelling non-linear or near non-linear functions [245, 236]. This is because HMMs rely on a set of predefined states and transition probabilities, making assumptions about the linearity and stationarity of the underlying data. While suitable for modelling certain aspects of speech, HMMs often fall short when tasked with representing the intricate nonlinearities and variability present in speech signals. Speech, by nature, exhibits nonlinear and dynamic characteristics, with features such as intonation, rhythm, and phonetic variations challenging the simplistic assumptions of traditional statistical models like HMMs. In other words, GMM-HMM approach had limitations in capturing complex acoustic patterns and long-term dependencies in speech [246].

Sequence-to-Sequence Models: In recent times, CNN and RNNs have been leveraged for automatic speech recognition in order to consider a longer or variable temporal window for context information extraction [247]. Generally, CNNs are well-suited for capturing local patterns and hierarchical features in data, making them effective for modelling acoustic features in speech. By directly learning features from raw speech signals, CNNs bypassed the need for handcrafted features used in traditional GMM-HMM systems.

Additionally, CNNs can capture long-range dependencies in the data, which is crucial for understanding the context of speech. Likewise, the RNNs are suitable choice for exploring extended temporal context information in one processing level for feature extraction and modelling.

Based on this, several researchers have proposed the use of both CNN and variants of RNNs for automatic speech recognition and for other speech related tasks. For instance, [248], used CNN to classify speech emotions and benchmarked on a dataset consisting of seven classes (anger, disgust, fear, happiness, neutral, sadness, and surprise). However, CNNs lack the ability to model temporal dependencies explicitly. In speech recognition, understanding the temporal context of speech is essential for accurate transcription. Furthermore, speech signals are inherently sequential, and information from previous time steps is crucial for understanding the current speech segment. CNNs, by design, do not inherently capture this sequential nature. For this reason, variants of RNNs have been leveraged to collect extended contexts in speeches. This is because RNNs are designed to model sequential data by maintaining hidden states that capture information from previous time steps. This allows them to capture temporal dependencies effectively, making them well-suited for ASR tasks. In [249], the authors evaluated the performance of RNN, LSTM, and GRU on a popular benchmark speech dataset (ED-LIUM). The results show that LSTM achieved the best word error rate, while the GRU optimization was faster and achieved word error rate close to that of LSTM.

However, RNN architectures process input sequences sequentially, which limits their ability to capture global context information effectively. As a result, they may struggle to understand the entire context of a spoken utterance, leading to lower transcription accuracy, particularly in tasks requiring understanding beyond local dependencies. Furthermore, most CNN and RNN automatic speech recognition systems consist of separate acoustic, pronunciation, and language modelling components that are trained independently. Usually, the acoustic model bootstraps from an existing model that is used for alignment to train it to recognize context dependent (CD) states or phonemes. The pronunciation model, curated by expert linguists, maps the sequences of phonemes produced by the acoustic model into word sequences. For this reason, Sequence-to-Sequence models are being proposed in automatic speech recognition to train the acoustic, pronunciation, and language modelling components jointly in a single system [250]. Sequence-to-Sequence models in automatic speech recognition are a class of models that aim to

directly transcribe an input sequence of acoustic features such as speech spectrograms or Mel-frequency cepstral coefficients into a sequence of characters or words representing the recognized speech. There have been various sequence-to-sequence models explored in the literature, including Recurrent Neural Network Transducer (RNN-T) [251], Listen, Attend and Spell (LAS) [252], Neural Transducer [253], Monotonic Alignments [254] and Recurrent Neural Aligner (RNA) [255].
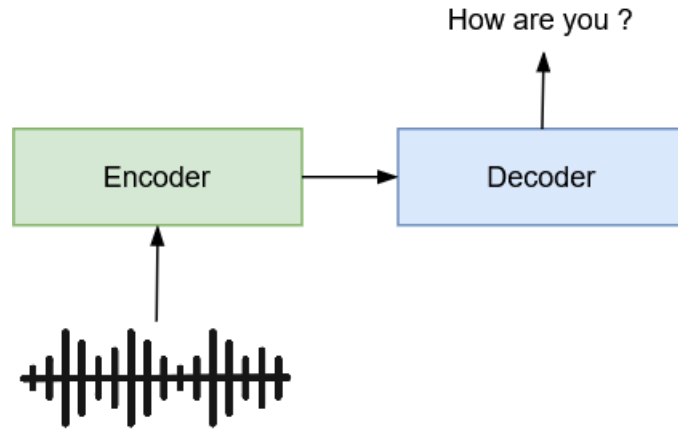
Figure 31: Sequence-to-Sequence

As shown in Figure 31, the encoder component takes the input sequence of acoustic features and processes it to create a fixed-dimensional representation, often called the context vector. This representation captures the essential information from the input sequence and serves as the basis for generating the output sequence. The decoder component takes the context vector produced by the encoder and generates the output sequence. In ASR, this output sequence consists of characters or words representing the recognized speech. The decoder is typically implemented as a recurrent neural network (RNN), such as a Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) network, or it could be a transformer-based architecture. During training, the model learns to map input sequences to their corresponding output sequences by minimizing a suitable loss function, such as cross-entropy loss. This is typically done using techniques like backpropagation through time (BPTT) or teacher forcing, where the model is trained to predict the next token in the output sequence given the previous tokens. Thereafter, the trained model is used to transcribe unseen speech input. The

encoder processes the input sequence to produce the context vector, which is then fed into the decoder to generate the output sequence. In some cases, beam search [256, 257] or other decoding strategies may be used to improve the quality of the generated output.

In [258], the authors explored various structural and optimization enhancements to their LAS Sequence to Sequence model, resulting in significant performance improvements. They introduce several structural enhancements, including the utilization of word piece models instead of graphemes and the incorporation of a multi-head attention architecture, which outperforms the commonly used single-head attention mechanism. Additionally, they investigate optimization techniques such as synchronous training, scheduled sampling, label smoothing, and minimum word error rate optimization, all of which demonstrate improvements in accuracy. The authors presented experimental results utilizing a unidirectional LSTM encoder for streaming recognition. On a 12,500-hour voice search task, they observe a decrease in Word Error Rate (WER) from 9.2% to 5.6% with the proposed changes, while the best-performing conventional system achieves a WER of 6.7%. Moreover, on a dictation task, their model achieves a WER of 4.1%, compared to 5% for the conventional system. Similarly, the work of Prabhavalkar et al. [250] investigated a number of sequence-to-sequence methods in automatic speech recognition. These included the RNN transducer (RNN-T), attention-based models, a new model that augments the RNN-T with attention, and a Connectionist Temporal Classification (CTC) trained system that directly outputs grapheme sequences. According to their research, sequence-to-sequence approaches can compete on dictation test sets against state-of-the-art when trained on a large volume of training data.

Transformers: Transformers have become the basis of state-of-the-art models in speech recognition. Wav2Vec 2.0 reads raw audio signals using multilayer convolutional encoder to generate latent speech representations [259]. The feature representation is subsequently fed to a transformer to capture the contextual information in the data. The transformer uses a convolutional layer for the positional encoding. Furthermore, the authors introduced vector quantization technique to convert the feature encoder output to discrete codes which are then used for contrastive learning during pretraining. Whisper is a robust speech recognition model proposed by OpenAI, trained on 680,000 hours of labeled audio data [260]. Of these, 117,000 hours cover 96 non-English languages and 125,000 hours consist of non-English to English translation data. The model is based on encoder-decoder transformer ar-

chitecture, where the input data is first processed using convolutional layers followed by the positional encoding. Then, the input is fed into the encoder, which consists of a series of transformer blocks, each including multi-head self-attention mechanisms. The encoder's output is then fed into each decode transformer block to generate the final output. The decoder's transformer blocks consist of multi-head self-attention and cross attention mechanisms.Conformer is a Transformer-based encoder, combining self-attention with convolutional layers to better capture both global and local features in speech signals [261]. The encoder provides a balance approach leveraging the strengths of convolution and self-attention mechanism. Conformer is often used with CTC or RNN-T, or other decoding mechanisms.

Auto-regressive decoding in transformer-based models is computationally expensive, as it requires repeated processing of the complete encoded speech, resulting in slow operation. In [262], Aligner-Encoder, a transformer-based encoder is proposed that perform internal alignment between audio and text, eliminating the need for complex decoding steps and complex dynamic programming during training. The approach simplifies the model architecture using a lightweight decoder without cross attention that processes embedding frames sequentially until an end-of-message token is generated. In [263], the Spike Window Decoding (SWD) algorithm is introduced that leverages the spike property of CTC outputs where each spike represents a strong signal indicating the presence of a specific token. By focusing on these spikes within a fixed window, SWD reduces the complexity of decoding, enabling faster and more efficient processing. Although transformer-based models are highly effective, they often capture highly entangled feature representations, leading to a lack of clear interpretability. In [264], the authors proposed the Disentangled-Transformer to enhance the interpretability of transformer-based models. The transformer disentangles the internal representations into sub-embeddings based on the various temporal characteristics of the speech signals. To this end, the loss function is improved by introducing time-invariant regularization terms for each time frame. A state-of-the-art speech recognition model called Samba-ASR is proposed utilizing Mamba architecture that is based on state-space-models [265]. This approach overcomes the limitations of transformer-based models such as inability to handle long-range dependencies and their quadratic scaling with input length, by efficiently capturing both local and global temporal dependencies through state-space-dynamics. Table 10 lists the summary of state-of-the-art speech recognition applications.

Table 10: Summary of state-of-the-art speech recognition.

Ref.	Description	Results (Datasets and Metrics)
[260], 2022	This study proposed Whisper an automatic speech recognition (ASR) system that transcribes and translates spoken language using a robust deep learning model trained on diverse multilingual and multitask datasets	Various datasets (first three datasets only) LibriSpeech Clean WER: 2.7 Artie WER: 6.2 Common Voice WER: 9.0
[262], 2025	This study introduced Aligner-Encoder, a transformer-based automatic speech recognition model that internally aligns audio-text representations before decoding, enabling a simpler and more efficient architecture trained with frame-wise cross-entropy loss	LibriSpeech-960H WER: 2.2 Voice Search WER: 3.7 YouTube videos WER: 7.3
[263], 2025	An algorithm that leverages spike property of CTC outputs to reduce the complexity of WFST decoding process	AISHELL-1 CER: 3.89 In House CER: 2.09
[264], 2024	A transformer-based model architecture designed to disentangle internal representations and enhance model explainability.	LibriSpeech-100H DER: 8.1 LibriMix 1 DER: 5.7 LibriMix 2 DER: 6.9 LibriMix 3 DER: 2.5 LibriMix 4 DER: 5.6
[265], 2025	A speech recognition model based on state-space-models	GigaSpeech WER: 9.12 LibriSpeech Clean WER: 1.17 LS Other WER: 2.48 SPGISpeech WER: 1.84

4.3.3. Finance

Over the past few decades, computational intelligence in finance has been a hot issue in both academia and the financial sector [266]. Deep learning, especially RNN models have gained significant traction in the field of finance due to its ability to handle sequential data, since financial data often exhibit sequential dependencies, such as time series data for stock prices or historical transaction data. Within the financial industry, researchers have developed deep learning models for stock market forecasting [201], algorithmic trading [203], credit risk assessment [267], portfolio allocation [268], asset pricing

[269], and derivatives markets [270], among others and these models are intended to offer real-time operational solutions. In exchange rate prediction, Sun et al. [271] developed an ensemble deep learning technique known as LSTM-B by combining a bagging ensemble learning algorithm with a long-short term memory (LSTM) neural network to increase the profitability of exchange rate trading and produce accurate exchange rate forecasting results. In comparison to previous methodologies, the authors' estimates proved to be more accurate when they looked at the potential financial profitability of exchange rates between the US dollar (USD) and four other major currencies: GBP, JPY, EUR, and CNY.

The authors in [272] proposed a Bi-LSTM-BR technique, which combined Bagging Ridge (BR) regression with Bi-LSTM as base regressors. The pre-COVID-19 and COVID-19 exchange rates of 21 currencies against the USD were predicted using the Bi-LSTM BR, and experiments showed that the proposed method outperformed ML algorithms such as DT and SVM. However, exchange rate data can be noisy and subject to non-stationarity, which can pose challenges for predictive modelling. While bagging techniques can help mitigate the effects of noise to some extent, they may struggle to capture long-term trends or sudden shifts in the data distribution, leading to suboptimal performance. To address this, Wang et al. [273] presented an approach for one-day ahead of time exchange rate prediction that concurrently considers both supervised and unsupervised deep representation features to enhance Random Subspace. Two crucial phases in the SUDF-RS technique are feature extraction and model building. First, LSTM and deep belief networks, respectively, extract the supervised and unsupervised deep representation features. To produce high-quality feature subsets, an enhanced random subspace approach was created that integrates a random forest-based feature weighting mechanism. Then, the matching base learner is trained using each feature subset, and the final outputs are generated by averaging the outcomes of each base learner. Experiments on EUR/USD, GBP/USD and USD/JPY showed that improved accuracy was achieved using the model.

In stock market prediction, several deep learning architectures have been proposed in the literature. For instance, [274], conducted a comparative study between the ANN, SVR, RF and an LSTM model. As compared to the other models discussed in the study, the LSTM model outperformed the others in predicting the closing prices of iShares MSCI United Kingdom. Similarly, using stock market historical data and financial news, Cai et al. [275] used CNN and LSTM forecasting methods to generate seven predic-

75

tion models. The seven models were then combined into a single ensemble model in accordance with the ensemble learning approach to create an aggregated model. However, the accuracy of all the models' predictions was low. Gudelek et al. [276] proposed a CNN model which used a sliding window technique and created pictures by capturing daily snapshots within the window's bounds. With 72% accuracy, the model was able to forecast the prices for the following day and was able to generate 5 times the starting capital. In Eapen et al. [277], a CNN and Bi-LSTM model with numerous pipelines was proposed, utilising an SVM regressor model on the S&P 500 Grand Challenge dataset, and results showed enhanced prediction performance by over a factor of 6% compared to baseline models. As presented, deep learning has undeniably achieved state-of-the-art performance across various domains within finance. In [278], a multi-modal deep learning model is proposed for stock market trend prediction by integrating daily stock prices, technical indicators and sentiment from daily news headlines. The model architecture consists of BERT-based branch fine-tuned on financial news to capture textual sentiment and an LSTM branch that captures temporal patterns in the time series data, including stock prices and technical indicators. Both feature representations are combined through concatenation and then passed to a fully connected layer for predictions.

A study was conducted to analyze the performance of Prophet, LSTM and Transformer. Prophet is a parametric, additive regression model based on time series decomposition [279]. The LSTM model consists of two hidden layers, with a hidden size of 64 and a linear fully connected layer is used to generate the output. The Transformer model follows the original transformer architecture. The results show that transformer outperformed LSTM and Prophet in terms of forecasting accuracy, particularly for datasets with complex temporal dependencies. However, transformer models have high memory usage and quadratic complexity due to self-attention mechanism, which makes them inefficient for very long sequence time-series forecasting. To overcome the limitations, a StockFormer is proposed which is based on Informer [280]. Informer is a transformer that was designed for time-series forecasting, utilizing the ProbSparse self-attention mechanism which attends to only the most important queries for each key, reducing the computational complexity [281]. Furthermore, self-attention distilling technique is introduced to reduce redundancy and highlighting crucial information in the feature maps, helping the model to focus more on the relevant parts of the input sequence. In [282], a transformer-based model is proposed for stock trend prediction using limit

Table 11: Summary of state-of-the-art finance applications.

Ref.	Description	Results (Datasets and Metrics)
[273], 2023	This study introduces SUDF-RS technique for one-day-ahead exchange rate prediction by combining supervised and unsupervised deep representation features with an enhanced random subspace approach, leveraging LSTM and deep belief networks for feature extraction and using a random forest-based weighting mechanism to improve prediction accuracy	Data collected from investing.com RMSE: 0.448 (EUR/USD), 0.676 (GBP/USD), 0.374 (USD/JPY)
[278], 2024	A multi-modal deep learning model integrates stock prices, technical indicators, and news sentiment using a BERT-based branch for textual analysis and an LSTM branch for temporal patterns, combining their features for stock market trend prediction	Data collected using Yahoo Finance and EODHD API F1-Score: 0.45 (ATVI), 0.51 (AAPL), 0.48 (AMT), 0.53 (PLD), 0.50 (NDAQ), 0.45 (SCHW), 0.43 (BIO), 0.47 (JNJ)
[279], 2024	This study analyzes stock closing price prediction using LSTM, Prophet, and Transformer models	American Airlines Group Inc. MSE: 0.0085 (Transformer), 0.1972 (LSTM), 16.9321 (Prophet) Atlantic American-Life Insurance MSE: 0.0118 (Transformer), 0.0095 (LSTM), 17.8601 (Prophet)
[280], 2025	The study proposed a model based on Informer which utilizes ProbSparse self-attention and self-attention distilling to reduce the computation complexity	Combined data from Yahoo, alphavantage.co, alpaca.markets and polygon.io Percent profit: 1.7550
[282], 2025	The study proposed a transformer-based model with dual attention mechanisms and multilayer of perceptrons for feature mixing along feature and temporal dimensions	FI-2010 F1-score: 92.81 TSLA F1-score: 60.50 INTC F1-score: 80.15

order book data (LOB). The model consists of a series of transformer LOB (TLOB) blocks which leverage dual attention mechanisms to capture both spatial and temporal dependencies, allowing the model to adaptively focus on the market microstructure of the input data. Furthermore, a multilayer of perceptron LOB (MPLOB) is introduced, comprising two blocks of two fully connected layers which operate along the feature and temporal dimensions. Table 11 lists the summary of state-of-the-art finance applications.

4.3.4. Electrocardiogram (ECG) Classification

Disorders pertaining to the heart or blood vessels are collectively referred to as Cardiovascular Diseases (CVD) [283]. According to the American Heart Association's 2023 statistics, CVD has emerged as the leading cause of death

worldwide. In 2020, 19.05 million deaths were recorded from CVD globally, which signifies an increase of 18.71% from 2010, and it is believed that this number will rise to 23.6 million by 2030 [284]. Blood clots and vascular blockages caused by CVDs can cause myocardial infarction, stroke, or even death [283]. Generally, early diagnosis has been shown to reduce the mortality rate of CVDs, and Electrocardiogram (ECG) signals play a crucial role in diagnosing various cardiac abnormalities and monitoring heart health. However, ECG signal has characteristics of high noise and high complexity, making it time-consuming and labor-intensive to identify certain diseases using traditional methods. The traditional approach is tedious and requires the expertise of a medical specialist. Over the past decades, the task of Long-term ECG recording classification has been significantly facilitated for cardiologists through the adoption of computerized ECG recognition practices. Throughout this period, feature extraction methods have predominantly relied on manual techniques, encompassing diverse approaches such as wave shape functions [285], wavelet-based features [286], ECG morphology [287], hermite polynomials [288], and Karhunen-Loeve expansion of ECG morphology [289], among others. These extracted features are subsequently subjected to classification using various machine learning algorithms.

More recently, the advent of deep learning has revolutionized the field by enabling automatic feature learning directly from ECG signals. This advancement holds significant promise in the realm of automated ECG classification, offering clinicians a tool for swift and accurate diagnosis. Based on this, several deep learning architectures have been proposed for feature learning of ECG signals. For instance, Acharya et al. [290] developed a 9-layer CNN model to automatically identify five categories of heartbeats in ECG signals. A similar model was also developed in [291]. However, ECG signals often vary significantly in length, as they may contain different numbers of heartbeats. CNNs typically require fixed-length inputs, which may necessitate preprocessing steps such as padding or truncation, potentially losing important temporal information. For this reason, several architectures have leveraged RNN in ECG classification, as seen in [292], [293] and [294], among others. While RNNs are capable of handling sequential data, they also have limitations in capturing local patterns or short-term dependencies effectively. In ECG signals, local features such as specific waveforms or intervals can be crucial for classification. For this reason, recent works have proposed hybrid models which combine the strengths of both CNNs and RNNs to overcome some of these limitations [295].

Hybrid Models: The work of Rai et al. [296] developed a hybrid CNN-LSTM network to evaluate the optimum performing model for myocardial infarction detection using ECG signals. The authors then experimented on 123,998 ECG beats obtained from the PTB diagnostic database (PTBDB) and MIT-BIH arrhythmia database (MITDB), and the result showed that by combining the capabilities of both CNN and LSTM, improved classification accuracy can be achieved. Also, in [297], a CNN architecture was developed to extract morphological features from ECG signals. For the purpose of determining the degree of heart rate variability, another composite structure was designed using LSTM and a collection of manually created statistical features. Following that, a hybrid CNN-LSTM architecture is built using the two independent biomarkers to classify cardiovascular artery diseases, and experiments were carried out on two distinct datasets. The first is a partly noisy in-house dataset collected using an inexpensive ECG sensor, and the other is a corpus taken from the MIMIC II waveform dataset. The hybrid model proposed in the work achieved an overall classification accuracy of 88% and 93%, respectively, which surpasses the performance of standalone architectures.

An automated diagnosis method based on Deep CNN and LSTM architecture was presented in [298] to identify Congestive Heart Failure (CHF) from ECG signals. Specifically, CNN was used to extract deep features, and LSTM was employed to exploit the extracted features to achieve the CHF detection goal. The model was tested using real-time ECG signal datasets, and the results show that the AUC was 99.9%, the sensitivity was 99.31%, the specificity was 99.28%, the F-Score was 98.94%, and the accuracy was 99.52%. Similar work is reported in [299] whereby CNN is combined with LSTM for ECG classification. The hybrid model consists of three blocks of convolutional and maximum pooling layers with dropout followed by an LSTM with drop. Furthermore, Shapley is utilized to determine the contribution of each ECG sample on the prediction to improve model interpretability. However, since ECG signals can vary in length due to differences in recording durations or patient conditions. LSTMs are capable of handling variable-length sequences, but traditional CNNs typically require fixed-length inputs. Therefore, fusing these features effectively in a hybrid model can be challenging. Furthermore, Hybrid CNN-RNN models can be computationally intensive, especially when processing long ECG sequences or large datasets. For this reason, recent research works have proposed the use of attention mechanisms to reduce the computational burden by enabling the model to selectively at-

79

tend to informative features, focusing computational resources where they are most needed. Likewise, attention mechanisms can enable the model to attend to informative segments of the ECG signal, regardless of their length, allowing for more flexible processing of variable-length sequences.

Several researchers have leveraged attention mechanisms in standalone and hybrid architectures for improved performance. For instance, in the work of Chun-Yen et al. [300], CNN layers were used to extract main features, while LSTM and attention were included to enhance the model's feature learning capabilities. Experiments on a 12-lead KMUH ECG dataset showed that the model had high recognition rates in classifying normal and abnormal ECG signals, compared to hybrid models without attention mechanisms. Wang et al. [301] presented a 33-layer CNN architecture with non-local convolutional block attention module (NCBAM). To extract the spatial and channel information, preprocessed ECG signals were first fed into the CNN architecture. A non-local attention further captured long-range dependencies of representative features along spatial and channel axes. Similarly, a spatio-temporal attention-based convolutional recurrent neural network (STA-CRNN) was presented in [302] with the aim of concentrating on representative features in both the spatial and temporal dimensions. The CNN subnetwork, spatiotemporal attention modules, and RNN subnetwork formed the STA-CRNN and according to findings, the STA-CRNN model was able to classify eight different forms of arrhythmias and normal rhythm with an average F1 score of 0.835. In [303], the SE module is utilized in between convolutional layers and LSTM. This approach allows the model to focus on relevant channels before temporal dependencies are captured by the LSTM, improving the classification performance.

In [304], a guided spatial attention mechanism is introduced to incorporate domain knowledge into the model, enhancing ECG classification performance. The guided spatial attention mechanism has an autoencoder-like structure, where the feature maps are downsampled by the encoder and upsampled by the decoder. Additionally, skip connections are employed to improve information flow between the encoder and decoder. Furthermore, an attention loss term, based on the attention weights is introduced to jointly train the guided attention mechanisms. In [305], ECG signals are classified into sleep stages using deep learning. The approach is divided into three stages. In the first stage, a deep learning model consisting of convolutional layers and liquid time-constant network [306] is employed to compute kurtosis and skewness of the ECG signals. In the second stage, a deep learning

model with convolutional layers and SE attention is used to enhance the prediction of the minority sleep stage class. In the final stage, the outputs of the models are concatenated and classified into sleep stage categories. Combining hybrid deep learning models with attention mechanisms for ECG feature learning is a promising approach that has already shown potential in ECG feature learning, according to reviewed literature. Future research can further explore semi-supervised and self-supervised learning techniques to leverage large amounts of unlabeled ECG data. This could involve pre-training models on large-scale unlabeled datasets using self-supervised learning objectives.

Generative Models: Deep learning models have been leveraged in the generation of synthetic ECG signals to augment real signals, as seen in [307] where a GAN model was developed to generate ECG signals that correspond with available clinical data. The GAN model used two layers of BiLSTM for the generator and CNN for the discriminator, and trained using the 48 ECG recordings of different users from the MIT-BIH dataset. The authors then compared their model with a Recurrent neural network autoencoder (RNN-AE) model and a recurrent neural network variational autoencoder (RNN-VAE) model, and the results show that their model exhibited the fastest convergence of its loss function to zero. Similar work is reported in [308] in which GAN and autoencoder are employed to generate ECG signals to address the issue of imbalanced dataset as shown in Figure 32. The generator and encoder create synthetic and real low-dimensional ECG samples, respectively. The discriminator classifies the synthetic and real low-dimensional ECG samples as real or fake, while the decoder reconstructs the ECG signals from the synthetic low-dimensional samples. This approach further refines data transformation and reconstruction quality, generating more realistic ECG signals. In [309], cluster-based GAN is employed to improve the diversity and quality of the generated synthetic ECG signals. K-means algorithm is first employed to partition the ECG signals into k distinct groups. Then, the GAN is trained using the cluster centroids as initial reference points to guide the GAN in generating synthetic signals. Table 12 lists the summary of state-of-the-art ECG classification.

4.3.5. Electroencephalography (EEG) Classification

Three-dimensional scalp surface electrode readings provide a dynamic time series that is called Electroencephalogram (EEG) signal [310]. Brain
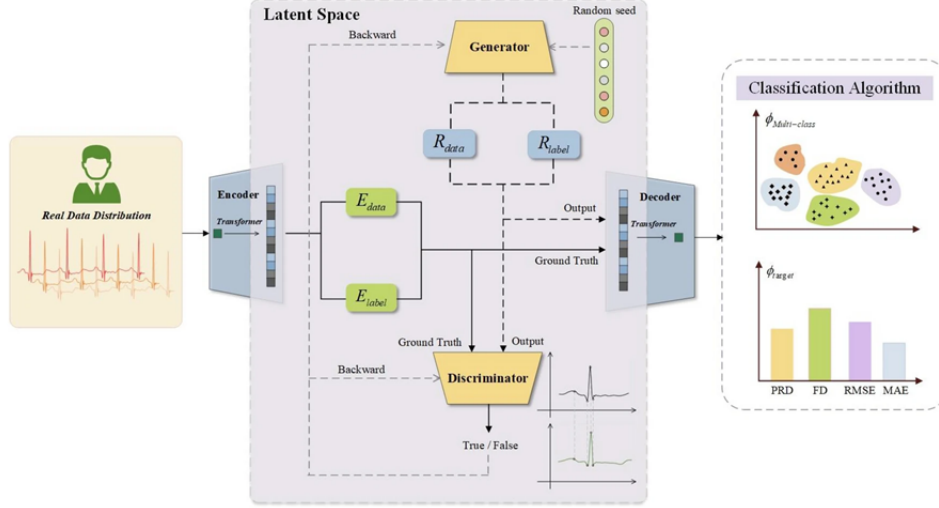
Figure 32: The architecture of CECG-GAN [308].

waves obtained from an EEG can effectively depict both the psychological and pathological states of a human. The human brain is acknowledged to be a fascinating and incredibly complicated structure. Numerous brain signals, including functional magnetic resonance imaging (fMRI), near-infrared spectroscopy (NIRS), electroencephalograms (EEGs), and functional near-infrared spectroscopy (fNIR), among others have been collected and used to study the brain [311]. Due to the EEG's non-invasive, affordable, accessible, and excellent temporal resolution characteristics, it has become the most utilized approach. However, the signal-to-noise ratio of EEG signal is low, meaning that sources with no task-relevant information frequently have a stronger effect on the EEG signal than those that do. These characteristics often make end-to-end feature learning for EEG data substantially more challenging [310]. Based on this, several methods have been leveraged for improved feature extraction in EEG signals across several domains including Motor imagery [312], anxiety disorder [313], epileptic seizure detection [314], sleep pattern analysis and disorder detection [315, 316], and Alzheimer's disease detection [317], and many others.

EEG Motor Imagery (MI) is a technique used to study brain activity associated with the imagination of movement. It involves recording electrical activity generated by the brain through electrodes placed on the scalp. MI tasks typically involve imagining performing a specific motor action, such as

82

Table 12: Summary of state-of-the-art ECG classification.

Ref.	Description	Results (Datasets and Metrics)
[303], 2024	A hybrid model consists of three blocks of convolutional layer, SE module and an LSTM	MIT-BIH ECG Accuracy: 98.5%
[299], 2024	A lightweight hybrid model consists of three blocks of convolutional and maximum pooling layers and an LSTM	MIT-BIH and LTAF Accuracy: 98.24%
[300], 2022	A hybrid model consists of three convolutional pipelines with different kernel size, two LSTM layers and an attention module	KMUH Accuracy: 96.02% CPSC-2018 Accuracy: 94.05%
[304], 2024	This study proposed a guided spatial attention mechanism to enhance ECG classification performance	CPSC2018 and PTB-XL Chapman Cinc2017 F1-score: 77.56% (STC), 88.53% (PC), 82.54% (WPW), 85.37% (AF)
[305], 2024	A novel three-stage approach for sleep stage classification extracts statistical features using a Feature Imitating Network, enhances minority sleep stage detection, and integrates outputs for five-class classification	MIT-BIH Polysomnographic Accuracy: 80.79%
[308], 2024	This study proposed a generative model by combining conditional GAN and autoencoder	CSPC2020 dataset F1-score: 0.9823 (Normal), 0.9824 (Premature ventricular), 0.9825 (Premature Supraventricular)
[309], 2024	This study introduces a generative model by combining GAN and K-means	MIT-BIH arrhythmia Accuracy: 99.66%

moving a hand or foot, without physically executing the movement, and has been leveraged in smart healthcare applications such as post-stroke rehabilitation and mobile assistive robots, among others [318]. Prior to the advent of deep learning, motor imagery EEG data are passed through various steps before classification using traditional ML techniques. Pre-processing, feature extraction, and classification are the three primary stages that traditional approaches usually take while processing MI-EEG signals. Pre-processing includes a number of operations, including signal filtering (choosing the most valuable frequency range for MI tasks), channel selection (identifying the most valuable EEG channels for MI tasks), signal normalization (normalizing each EEG channel around the time axis), and artefact removal (removing noise from MI-EEG signals). Independent component analysis (ICA) is the most often utilized technique for removing artefacts [319, 320, 321]. In contrast to the traditional approach, deep learning architectures can automatically extract complex features from raw MI-EEG data without the need for laborious feature extraction and pre-processing. Based on this, several deep learning architectures have been proposed for MI-EEG feature learning, as seen in [322], [323] and [324], among others. For instance [310], categorized MI-EEG signals using three CNNs with varying architectures, and the

number of convolutional layers varied from two layers to a five-layer deep ConvNet to a thirty-one-layer residual network.

In Dai et al. [325], the authors proposed an approach for classifying MI-EEG signals which blend variational autoencoder with CNN architecture. The VAE decoder was used to fit the Gaussian distribution of EEG signals, and the time, frequency, and channel information from the EEG signal were combined to create a novel representation of input, and the proposed CNN-VAE method was optimized for the input. Experiments showed that by combining both deep learning architectures, improved features were learnt, which led to a high classification performance on the BCI Competition IV dataset 2b. Li et al. [326] employed optimal wavelet packet transform (OWPT) for the generation of feature vectors from MI-EEG signals. These vectors were then utilized to train an LSTM network, which demonstrated satisfactory performance on dataset III from the BCI Competition 2003. However, the model has an excessively intricate structure. To address this, Feng et al. [327] introduced a technique that merges continuous wavelet transform (CWT) with a simplified convolutional neural network to enhance the accuracy of recognizing MI-EEG signals. By employing CWT, MI-EEG signals were converted into time-frequency image representations. Subsequently, these image representations were fed into the SCNN for feature extraction and classification. Experiments on the BCI Competition IV Dataset 2b demonstrate that, on average, the classification accuracy across nine subjects reached 83.2%. However, the computational complexity of the model was quite high, due to the processing of time-frequency image representations. The conversion of MI-EEG signals into time-frequency images using CWT requires significant computational resources.

The authors in [328] introduced a classification framework based on Long Short-Term Memory (LSTM) to improve the accuracy of classifying four-class motor imagery signals from EEG. The authors used sliding window technique to capture time-varying EEG signal data, and employed an overlapping-band-based Filter Bank Common Spatial Patterns (FBCSP) method to extract subject-specific spatial features. Experiments on the BCI Competition IV dataset 2a, showed that their model achieved an average accuracy of 97%, compared to existing methods. Also, in the classification of Alzheimer's disease, Zhao et al. [329] employed a deep learning network to analyze EEG data. The deep learning model was evaluated on a dataset that consisted of fifteen (15) patients with clinically confirmed Alzheimer's disease and fifteen (15) healthy individuals, and results showed that improved features

were learnt and compared the results to the traditional methods. This has prompted the use of deep learning in Alzheimer's disease detection, as seen in [330], where the authors used CNN for diagnosing Alzheimer's Disease. To address challenges posed by limited data and overfitting in deep learning models designed for Alzheimer's Disease detection, the authors explored the use of overlapping sliding windows to augment the EEG data collected from 100 subjects (comprising 49 AD patients, 37 mild cognitive impairment patients, and 14 healthy controls subjects). After assembling the augmented dataset, a modified Deep Pyramid Convolutional Neural Network (DPCNN) was used to classify the enhanced EEG signals.

Hybrid Models: In [331], the authors developed three deep learning architectures (CNN, LSTM, and hybrid CNN-LSTM), with each model chosen for its effectiveness in handling the intricate characteristics of EEG data for epilepsy detection. Each architecture offers distinct advantages, with CNN excelling in spatial feature extraction, LSTM in capturing temporal dynamics, and the hybrid model combining these strengths. The CNN model, consisting of 31 layers, attained the highest accuracy, achieving 91% on the first benchmark dataset and 82% on the second dataset using a 30-second threshold, selected for its clinical significance. In the work of Abdulwahhab et al. [332], EEG waves' time-frequency image and raw EEG waves served as input elements for CNN and LSTM models. Two signal processing methods, namely Short-Time Fourier Transform (STFT) and CWT, were employed to generate spectrogram and scalogram images, sized at 77×75 and 32×32, respectively. The experimental findings demonstrated detection accuracies of 99.57% and 99.26% for CNN inputs using CWT Scalograms on the Bonn University dataset and 99.57% and 97.12% using STFT spectrograms on the CHB-MIT dataset. Similarly, in emotion recognition, several deep learning models have been leveraged with EEG signals. For instance, in [333], a subject-independent emotion recognition model was proposed, which utilizes Variational Mode Decomposition (VMD) for feature extraction and DNN as the classifier. Evaluation against the benchmark DEAP dataset demonstrates superior performance of this approach compared to other techniques in subject-independent emotion recognition from EEG signals.

Some researchers have also combined EEG signals with facial expression and speech in emotion recognition, as seen in [334], [335], and [336], among others. In [335], a multi-modal emotion recognition framework is proposed, utilizing three deep learning models to extract features from facial expressions, speech and EEG signals. To process EEG signals, the authors designed

85

a tree-like LSTM model that extracts temporal features at different stages to capture multiscale feature representations. The output of each model is then weighted to generate the final predictions. In [336], the feature maps extracted by the deep learning models are combined through concatenation, and then they are fed into a fully connected classifier to generate the predictions. For EEG signals, two deep learning models are designed: one for extracting local features and the other for extracting global features. Both models are implemented using convolutional layers. However, EEG signals can vary significantly across individuals, making it challenging to generalize models across different subjects.

Generative Models: GANs have been employed to enhance EEG signal classification. In [337], a conditional GAN model is proposed to overcome the limited number of EEG samples. To improve data generation quality, the discriminator is integrated with SE attention and a cropped training strategy is employed to leverage the entire spectrum of the EEG signals. In [338], a Wasserstein GAN model is employed to enhance the spatial resolution of EEG signals for EEG classification. The model takes low-resolution signals and generates the corresponding high-resolution signals while simulataneously interpolating the missing channels. EEG signals are often corrupted with noise and artifacts. DHCT-GAN is a dual-branch hybrid model designed to generate denoised EEG signals by processing both clean and noisy/artifact-corrupted signals [339]. The generator consists of dual-branch feature learning pipelines, where one accepts clean signals while the other handles noisy/artifact signals to learn feature representations. The feature maps are then fused using two fully connected layers with a tanh activation function to generate denoised EEG signals. Both feature learning branches are identical, consisting of convolutional layers with local and global self-attention mechanisms, while the discriminators are implemented using convolutional and fully connected layers. Table 13 lists the summary of state-of-the-art EEG classification.

4.4. Robotics

Deep learning is a crucial component in modern robotics, enabling machines to perceive and interpret their surroundings. By leveraging sensors and cameras, robots can identify objects, navigate environments and interact with humans more effectively.

86

Table 13: Summary of state-of-the-art EEG classification.

Ref.	Description	Results (Datasets and Metrics)
[331], 2024	A hybrid CNN-LSTM model consists of a convolutional layer to extract local features and two LSTM layers for temporal modeling	UPenn and Mayo Clinic's Seizure Detection Challenge Accuracy: 89% F1-score (avg.): 87%
[335], 2024	A tree-like LSTM to capture multiscale temporal feature representations from EEG signals	MAHNOB-HCI Accuracy: 0.9450 (valence), 0.7628 (arousal)
[336], 2023	Parallel convolutional pipelines for extracting local and global features from EEG signals	DEAP Accuracy: 85.72% (valence), 87.97% (arousal) MAHNO B-HCI Accuracy: 85.98% (valence), 85.23% (arousal)
[332], 2024	A framework comprises two separated deep learning-based models using a 2D Convolutional Neural Network and an LSTM to process EEG signal images and normalized EEG signals	Bonn Accuracy: 99.75% CHB-MIT Accuracy: 97.12%
[337], 2024	A conditional GAN integrated with SE attention in the discriminator and employing a cropped training strategy for improved data generation	BCI Competition IV 2a Accuracy: 81.3% BCI Competition IV 2b Accuracy: 90.3%
[338], 2025	A Wasserstein GAN model is used to enhance the spatial resolution of EEG signals for classification by generating high-resolution signals from low-resolution inputs while simultaneously interpolating missing channels, addressing noise and artifact contamination	Dataset V (Berlin BCI) Accuracy: 83.88% (scale 2), 82.00% (scale 4)
[339], 2025	A dual-branch hybrid model that denoises EEG signals by separately learning features from clean and noisy inputs, fusing them through fully connected layers, and leveraging convolutional layers with self-attention mechanisms	EEGdenoiseNet(EEG, EOG, EMG), MIT-BIH Arrhythmia (ECG) and semi-simulated EEG/EOG η (artifact reduction): 82.35% (EMG), 91.80% (EOG), 88.97% (ECG), 82.07% (EOG+EMG)

4.4.1. Object Identification

Deep learning plays a crucial role in object identification, allowing robots to recognize, classify, and differentiate between various objects in real-time. In [340], an intelligent system, integrating deep learning with robotic kinematic control is proposed for waste classification. The system consists of a robotic arm with a gripper and a camera placed in front of the robot to capture real-time video. YOLO model is trained to recognize and localize objects, allowing the system to determine the location, pick them up and classify them. A similar system is reported in [341], where a robotic arm is used to pick up objects on a conveyor and a camera is mounted above the conveyor to capture real-time video. The waste sorting process is divided into two tasks: localization of the waste objects and their classification. The object localization is performed using Segment Anything architecture (SAM) [342] which is based on vision transformer. Several lightweight SAM models are proposed to solve the localization task efficiently while minimizing computational requirements. For classification, object images are cropped using SAM's output masks and classified by a MobileNetV2.

A deep learning-based method is proposed to detect tile edges and corners for tile-paving mobile robotic systems [343]. The camera is mounted on a tile-grabing platform that is parallel to the ground and approximately 10 to 20 cm above it. The system utilizes a YOLOv8 for semantic segmentation with rule-based post-processing techniques. An unmanned ground vehicle-based is proposed for automated structural damage inspection [344]. The robotic system utilizes LiDAR and a camera, integrating them via Robot Operating System for efficient control and data processing. The images are processed for the damage detection and segmentation using an improved YOLOv7 model, while the point cloud data is processed for real-time localization and 3D mapping using an enhanced Keep It Small and Simple-Iterative Closest Point algorithm. A similar system is reported in [345], where object detection and tracking approach is proposed to fuse point cloud and visual data. The approach leverage YOLOv8 real-time object detection capabilities, while LiDAR provides accurate 3D spatial information and distance measurements, enhancing perception reliability. A fusion method aligns LiDAR data with camera images, enabling accurate 3D bounding boxes and object tracking. Table 14 lists the summary of state-of-the-art robot object identification.

Table 14: Summary of state-of-the-art robot object identification.

Ref.	Hardware platform	Model	Task
[340], 2025	Robotic arm, AI Stereo ZED 2 camera, gripper, Intel Xeon E5 series CPU @ 2.40 GHz	YOLOv4	Waste classification
[341], 2025	PC workstation, industrial camera, AMD Threadripper 3970X 32 cores CPU @ 3.79 GHz, 128 GB RAM, two Titan RTX 24 GB GPUs	Segment Anything architecture (SAM)	Waste sorting
[342], 2023	Mobile robot, camera, Nvidia RTX 3060 GPU	YOLOv8	Tile edges and corners detection
[344], 2025	Husky Unmanned ground vehicle, Intel Realsense camera, Velodyne VLP-16 LiDAR, Intel i7 CPU	YOLOv7	Structural damage inspection
[345], 2024	Agilex Scout Mini robot, Velodyne LiDAR, Intel D435 camera	YOLOv8	Object tracking

4.4.2. Path Extraction and Navigation

Deep learning models have been applied for robot navigation systems. In [346], LiDAR and a camera are to capture point cloud and visual data for a comprehensive understanding and navigation in the environments. The study experiments with Faster R-CNN, YOLOv5 and YOLOv8 for object detection. Additionally, the paper compares the performance of these systems in various real-world environments, showcasing their potential to enhance autonomous navigation. Mobile robot navigation system is proposed utilizing semantic segmentation to determine the drivable paths [347]. First, Deeplabv3 [348] model with a ResNet-50 backbone is applied to segment scene images to extract drivable area. A perspective transformation then maps the segmented images to real-world space. Following the transformation, the image is divided into grids to determine the optimal path while avoiding obstacles. Finally, a PID controller guides the robot along the smoothed path to ensure accurate navigation in the environment.

In [349], a method for orchard robot navigation utilizing a modified YOLOv8 model is proposed. The model locates vine tree trunks and identifies the center points of tree trunks at the lower end of the detection boxes. The least squares method is then applied to fit reference lines on both sides of the trunk. Furthermore, a multiscale attention mechanism is incorporated into the model to prioritize relevant features, enhancing the model performance. In [350], deep learning is employed for navigation path extraction for agricultural robots. The authors propose Single-Stage Navigation Path Extraction Network (NPENet) to predict the road centerline in a single stage, eliminating the need for complex multi-stage processes such as line detection

89

Table 15: Summary of state-of-the-art path extraction and navigation.

Ref.	Hardware platform	Model	Task
[346], 2024	Mobile robot, 2MP Logitech webcam, LiDAR, Raspberry Pi 4, 4 GB RAM	YOLOv8	Semantic navigation
[347], 2024	Mobile robot, camera, Nvidia Jetson Nano	Deeplabv3+ (ResNet50 backbone)	Path planning
[349], 2024	Mobile robot, Intel RealSense D455 camera, Intel Core i7 10870H CPU @ 2.20 GHz, Nvidia GeForce GTX 1650 Ti GPU	YOLOv8	Navigation line extraction
[350], 2025	Mobile robot, Intel RealSense cameras, Nvidia Jetson Xavier	NPENet	Navigation path extraction

and segmentation. The proposed model uses residual blocks for feature extraction which includes batch normalization, leaky ReLU and convolutional layers, generating three key outputs: detecting if the robot is on the road, estimating the navigation line's angle and predicting the vanishing point of the road. These outputs enable the robot to determine its trajectory and real-time navigation in orchard environments. Table 15 lists the summary of path extraction and navigation.

4.4.3. Human-Robot Interaction

Deep learning has significantly advanced human-robot interaction, particularly through the integration of large language models (LLMs) designed for natural language processing. Despite their extensive knowledge, LLMs have limitations, as they can sometimes generate inaccurate information, a phenomenon known as hallucination [351]. To address this issue, LLMs need to be connected to the physical world by providing real-world contextual information to the models, enabling them to generate accurate responses that are relevant to the situation. This process is known as LLM grounding. In [352], the authors explore the use of LLMs in human-robot interaction, focusing on collaborative planning between humans and robots. The LLM is guided to generate outputs within the predefined set of robot skills using a few-shot prompting, while vision models such as SAM-CLiP and ViLD are used for object identification within the scene, enabling the robot to perform actions in the real-world. By leveraging LLMs, robots can understand and process natural language instructions and execute the corresponding actions, enabling efficient collaboration with human users. LLMs need to be aligned

with human intentions to ensure effective communication and interaction between humans and robots. In [353], a 3D scene graph is employed to represent objects and environments captured by 3D instance segmentation and 2D image classification. Given the 3D scene graph, a complex task is decomposed using LLM through step-by-step reasoning by progressively retrieving relevant nodes from the graph. During the task solving, human intent is automatically identified, and the system generates responses accordingly.

In [354], a modular system architecture that can be easily extended is proposed to integrate LLM models for human-robot conversation and collaboration. Figure 33 shows the system architecture, where the chat manager is the central to the system that coordinates the state and inference of the LLM model. The robot perception such as recognizing objects and human gesture is handled separately by different deep learning models while robot actions are operated by speech generation model and actuators for emotion expression, gaze control and arm movement. This approach allows the robot to engage in open-ended conversation and collaborate naturally with humans. Grounding LLMs can be challenging due to robot's multi-modal perception system. Furthermore, the incoming data has different sampling rate which makes data alignment difficult, causing the robotic system to miss valuable information. In [355], a framework is proposed for processing multi-modal inputs to generate coherent narratives about robot experiences including its internal status, current observation, planning and critical alert. The framework is divided into three stages: key event selection, experience summarization and narration generation modules. First, multi-modal data across three categories: environment, internal and planning are aligned by sampling at a fixed rate and associating each frame at the nearest timestamp. Based on the aligned data, key events are selected by detecting changes in optical flow, RGB images and joint states using heuristic and normalization techniques to identify moment of interest. Using the detected events, the robotic data is converted into natural language summaries, and an LLM generates user-friendly explanation by filtering repetitive and irrelevant information. Table 16 lists the summary of human-robot interaction.

5. Research Challenges

Despite the success of deep learning in various applications, there are still fundamental challenges faced by researchers and practitioners. This section
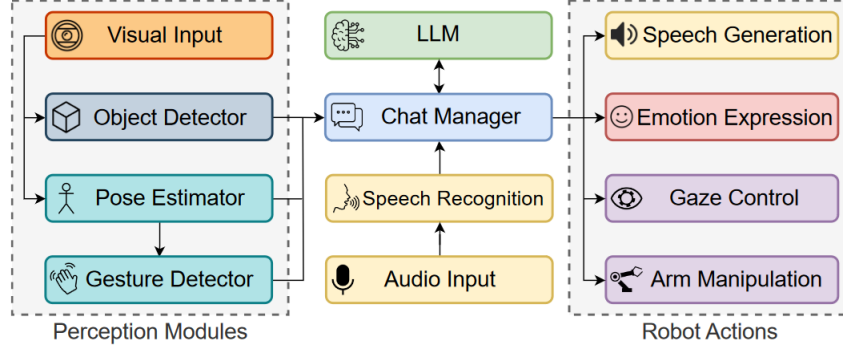
Figure 33: A grounded chat architecture [354].

Table 16: Summary of state-of-the-art human-robot interaction.

Ref.	Hardware platform	Model	Task
[352], 2025	Niryo Ned robot and Intel RealSense D435i camera	LLM: GPT-3.5, Object detection: SAM-CLiP and ViLD	Robot planning
[353], 2024	N.A.	LLM: GPT-4 turbo	Human collaboration
[354], 2024	Neuro-Inspired COLlaborator (NICOL) robot	LLM: GPT, Vicuna, Mistral Human detection: YOLOX Pose estimation: HRNet Object detection: ViLD	Conversation and collaboration
[355], 2024	MStretch SE3 robot, Intel RealSense D435i and D405	Object segmentation: YOLO World	Robot behavior announcement

discusses the challenges ethical, technical and domain-specific perspectives.

5.1. Ethical Issues

Interpretability and explainability is crucial for building trust and understanding how predictive models make decisions, especially in high-stake applications such as healthcare and medical image analysis [356]. However, as deep learning models become more intricate and complex with numerous layers, subnetworks and a large number of parameters, the models are often perceived as a "black box" and difficult to explain in terms of decision-making processes. Therefore, it is crucial for the researchers to focus on methods that provide insights into how a deep learning model performs the prediction and how its decisions are influenced by the input data, making it more transparent and trustworthy. Numerous methods have been proposed for interpreting and explaining the decisions of deep learning models, which can be categorized into visualization (feature attribution), model distillation and intrinsic

(explainable by itself). Visualization methods involve the use of scientific visualization such as saliency maps or heatmaps to express the explanation by highlighting the degree of association between the inputs and the predictions [357]. The heatmaps identify the saliency of the input features influencing the model's predictions. The visualization approach is simple and intuitive and can be applied to tabular data and image data. Furthermore, it can be used to identify and debug issues in deep learning models, leading to improved performance and robustness.

Model distillation is an approach to approximating a complex model by fitting a simpler model using the training set. The simpler model is built typically using a simpler or interpretable algorithm such as linear regression, decision tree or rule-based methods [358]. In this approach, the simpler model is trained to resemble the predictive behavior of the complex model. Then, the simpler model may serve as the proxy or surrogate model for explaining the complex model. Model distillation can be used together with visualization to further enhance the interpretability of the complex model [359]. Model distillation seeks explanations of the models that were never designed to be explainable. Ideally, the explanation of a deep learning model's prediction should be included as part of the model output, or the explanation can be derived from the architecture of the model. This is because an intrinsic model can learn not only the mapping between the input and output, but also generate an explanation of the prediction that is faithful to the model's behavior. Attention mechanisms are the key to this approach, providing a form of attention weights that can be used to explain why the model made a particular decision [360]. Another type of intrinsic approach is to train the model to simultaneously perform the prediction task and generate the explanation for its predictions [361]. This "additional task" can be in the form of a text explanation or model prototype which embeds the semantic meaning of the prediction. However, the intrinsic approach is more difficult to apply because the user needs additional knowledge and understanding of the model's architecture and inner workings.

Deep learning models are increasingly being deployed in making high-stake decision including recruitment [362], criminal justice [363] and credit scoring [364]. There are several advantages of deep learning-based systems in which, unlike humans, machines are able to process vast amounts of data and applications quickly and consistently. However, deep learning-based systems have the risk of being prone to biases present in the data used for training which can lead to unfairness and injustice. Numerous efforts have been made

to mitigate this issue which can be categorized into modelling bias detection and modelling bias mitigation. Detection of modelling bias refers to the process of identifying and quantifying biases that may present in predictive models. This approach involves the use of statistical analysis, fairness metrics, counterfactual testing and human review to detect bias in the models. For instance, visualization-based methods such as attribution maps are used to indicate which regions are significant to the predictions [365]. This in turn can be used to detect and quantify bias using metrics such as Relevance Mass Accuracy, Relevance Rank Accuracy and Area over the perturbation curve (AOPC). In [366], two modules are presented for estimating bias in predictive models. The first module utilizes an unsupervised deep neural network with a custom loss function to generate hidden representation of the input data called bias vectors, revealing the underlying bias of each feature. The second module combines these bias vectors into a single vector representing the bias estimation of each feature, achieved by aggregating them using the absolute averaging operation.

Bias mitigation refers to the process of reducing the presence of bias in predictive models, which can be done in three stages. The first stage combats bias by modifying the training data, either relabeling the labels or perturbing the feature values [367, 368]. The second stage addresses bias during the training of the model by applying regularization terms to the loss function to penalize discrimination. In [369], a loss function based on bias parity score (BPS) is introduced to measure the degree of similarity of a statistical measure such as accuracy across different subgroups. The BPS term is added to the loss function as a regularizer to the original prediction task. The last stage mitigates bias after the predictive models have been successfully trained. This stage applies post-processing approaches such as reinforcement learning to obtain a fairer model [370]. For instance, the detection of minority classes is rewarded to prevent bias towards the majority class. This allows the model to generalize well across different patient demographics.

5.2. Technical Issues

Even though deep learning architectures have achieved state-of-the-art across various computer vision tasks, they often come with large model parameters [371]. The architecture and complexity of a deep learning network determine the number of model parameters. The deeper the network, the larger the number of model parameters. However, deep learning models

with large parameters often suffer limitations when deploying on end devices. For instance, a deep learning model developed for security monitoring by analyzing video data using 3D-CNNs might suffer deployment issues when deploying such models on low-resourced systems like smartphones or small-scale IoT devices. Model training and inference for deep learning models with large parameters demands substantial processing power. As the number of parameters increases, so does the computational complexity, resulting in longer model training duration and more hardware needs. Furthermore, large parameter sizes translate to increased memory requirements, limiting their deployment on end devices. This is because these end devices often have battery, processor, or memory capacity limitations. To address these challenges, it is important to develop sophisticated but lightweight architectures that can achieve state-of-the-art with few model parameters. Such lightweight models will be characterized by their ability to deliver competitive performance while mitigating computational complexity and memory requirements, making them well-suited for deployment on resource-constrained devices. An approach would be to develop novel lightweight plug-and-play modules that can be plugged to a few layered deep learning architectures to improve feature learning without incurring additional model complexity. Other approaches could involve leveraging model compression techniques to reduce the size and computational complexity of deep learning models. Researchers can focus on improving pruning methods [372], which can identify and eliminate redundant parameters or connections, thereby reducing the model's footprint without compromising performance. Furthermore, quantization techniques [373] can be further explored to reduce the precision of weights and activations, therefore, enabling efficient representation with lower memory requirements. Also, knowledge distillation techniques [374] can be further investigated to facilitate the transfer of knowledge from a complex teacher model to a simpler student model, therefore, enabling compact yet effective representations.

Deep learning training involves complex processes that require efficient optimization to ensure fast convergence and resource management. As models grow in size and complexity, challenges such as slow convergence, vanishing gradients, and computational limitations become more pronounced. Researchers have explored various methods to overcome these issues, proposing innovative approaches like predicting parameter change and incorporating them into training, thus reducing training time and improving the model performance [375]. Researchers also leverage prior experience to optimize parameter adjustments to reduce forward and back propagation steps, thus

reducing computational costs and making training more efficient [376]. Others have proposed alternative training algorithms such as alternating minimization [377] and random search based on the annealing method [378]. Overall, the researchers aim to accelerate learning and improve efficiency either by decomposing the optimization problem into sequential sub-problems or by refining the search space based on parameter value ranges.

Adversarial attacks and defense mechanisms in deep learning represent a critical area of research and development, particularly as deep learning models become increasingly integrated into various applications. Adversarial attacks involve the deliberate manipulation of input data to mislead or deceive deep learning models, leading to incorrect predictions or behavior [379]. Szegedy et al. [380] was the first to identify this intriguing shortcoming of deep neural networks in image classification. They showed that even with their great accuracy, deep learning models are surprisingly vulnerable to adversarial attacks that take the form of tiny image changes that are (almost) invisible to human vision systems. A neural network classifier may radically alter its prediction about an image as a result of such an attack. Furthermore, such a model can indicate high confidence in wrong predictions, which can be catastrophic for deep learning models deployed in medical or security fields, among many others. In generative models, several studies have investigated how adversarial attacks affect autoencoders and GANs, as seen in Tabacof et al. [381] where a method to manipulate input images in a way that deceives variational autoencoders into reconstructing an entirely different image was introduced. In recent times, the focus of adversarial attack research has been on images, but studies have shown that adversarial attacks are not limited to image data; they can also affect other types of data such as text, signals, audio, and video [382, 383, 384].

5.3. Domain-specific Issues

Building and employing deep learning models face several challenges. The training of deep learning requires a large number of instances (examples) to achieve high accuracy and generalization [385]. Furthermore, the complexity of deep neural networks may lead to overfitting, where the model performs well on training data but fails to generalize on new, unseen data. This phenomenon frequently arises when the models are trained on insufficient data, highlighting the importance of diverse and extensive datasets. However, the data collection is time-consuming and often require domain experts, specialized training and standardization [386]. Moreover, this process is prone to

error and has the risk of introducing biases into the dataset which can significantly impact the performance of the trained model. One of the approaches to address this issue is transfer learning. Transfer learning involves the use of a deep learning model (known as pre-trained model) that is trained on a large dataset for solving a specific task (with a small dataset) [387]. The pre-trained model serves as a basis for the model training by fine-tuning the weights of the pre-trained model and adapting it to the new prediction task. This approach helps to mitigate the lack of training data in the target domain. Furthermore, transfer learning reduces computational resources required to train the model and helps faster convergence.

Another approach that can be employed to address the lack of data is data augmentation. Data augmentation is a convenient method that increases the number of instances by performing transformation functions on the existing instances without changing the labels [388]. In the domain of computer vision, image transformation such as rotation, translation and cropping. However, it is important to consider the output of the transformation because the resultant may not represent the actual data. For example, flipping or adding noise to a signal may introduce distortion or changing the characteristics (trend, seasonality and cyclic variations) of the signal. Thus, careful consideration must be given to ensure that the generated instances still accurately represent the underlying patterns present in the data. Data augmentation can also be realized by generating synthetic data to supplement the training set. Synthetic data is artificially created data that resembles real data but is generated using statistical methods or deep generative models [389, 390]. The generated data can complement the less-diverse, limited datasets, providing a broader range of examples for the model to learn from. However, generating synthetic data that accurately reflects the characteristics of the real-world data is challenging. Careful consideration must be given to the choice of models and parameters used to ensure the synthetic data is realistic and representative of the real-world data.

Data annotation is expensive and time-consuming. The problem is exacerbated when the data is of low quality corrupted with noise which may lead to bad, resulting in unreliable training data. One of the approaches addresses this issue is active learning. Active learning is an approach where the learning algorithm selectively queries the most informative data points for labeling. The aim is to improve predictive model efficiency with fewer training data, thus reducing the overall cost of data annotation. The approach involves training a model using an initial training set, and then a

subset of unlabeled data is selected for labeling by external annotators. The newly labeled instances are appended to the training set for retraining or fine-tuning. This process is repeated until the model performance reaches a desired threshold or the labeling budget has exhausted. The core principle behind active learning is the query strategy or, essentially how to select the most informative data for labeling that will be beneficial to the model training. In general, the query may select the most ambiguous instances based on the model predictions, instances that will have a significant impact on the model performance, or instances that cover the distribution of the entire feature space [391]. The dynamic and sequential nature of real-world applications presents different challenges, which led to the development of online active learning. This approach was introduced to address specific issues, including data streams, concept drift and environmental changes [392]. Recently, researchers have investigated the use of LLMs for data annotation. The study presents methods for generating annotations, assessing annotations and utilizing annotations [393].

Self-supervised learning is another approach that was introduced to mitigate the issue of limited labeled datasets. In self-supervised learning, the model learns from unlabeled data by generating its own labels from the input data [394]. This approach essentially creates a pretext task which the model solves without requiring manual annotations or labeled datasets. By solving the pretext task, the model leverages the underlying structure in the data and learn useful representations that can be used for specific tasks. For example, in computer vision, a model might predict the speed [395] or repeated scenes [396] in a video. In natural language processing, it could predict missing words in sentences to learn the dependencies between words [397]. The key advantage of self-supervised learning is that it allows the model to take advantage of large amounts of unlabeled data, which is often much more abundant and cheaper to collect than labeled data. The learned representations can then be used for downstream tasks like classification, regression, or segmentation.

6. Summary and Future Directions

We have discussed the state-of-the-art applications and challenges of deep learning in computer vision, natural language processing and time series analysis. In this section, we summarize the advancements made which can be

grouped into model architecture, contextual enhancement, and loss function and optimization. Finally, we discuss possible future works in these domains.

Model Architecture: The recent advancements in computer vision focus on vision transformers (ViT) for scalable representation learning [21]. In image classification and image segmentation, ViTs are utilized to capture global context and relationships between different parts of an image, while in object detection, ViTs are used to streamline the detection process by eliminating the traditional components like anchor boxes and non-maximum suppression [126]. However, ViTs lack the ability to exploit local spatial features and struggle with hyperparameter sensitivity and performance on smaller datasets. To address this issue, researchers have developed hybrid deep learning architectures such as conformer [107] and MaxViT [108] that combine both transformers and convolutional neural networks to capture both local and global features. Similar trend is observed in modern natural language processing whereby the basis of the deep learning models is transformer architectures such as BERT [162] and its variants to achieve remarkable accuracy in tasks such as text classification, machine translation and text generation.

While recent advancements in computer vision and natural language processing have predominantly relied on transformer architectures, the field of time series analysis focuses on a different trajectory, where transformers research is lagging due to fundamental challenges in data structure, sequence length and dataset size and availability. Notably, areas such as HAR and financial prediction have seen significant progress through the development of hybrid architectures that do not depend on transformers. Instead, these advancements leverage recurrent neural networks and convolutional neural networks, often incorporating attention mechanisms. The hybrid architectures allow the models to capture both local features and temporal dependencies in the data, which is crucial for time series analysis [222, 223]. The hybrid attention mechanisms improve the feature extraction process by dynamically weighting important channel, spatial and temporal features, enhancing model performance [224, 225, 226]. GAN models are used to address data scarcity in HAR, where the generator consists of convolutional layers [233, 234], with LSTM [231] while the discriminator consists of convolutional layers.

Similar trends can be observed in ECG and EEG classification, where convolutional and recurrent neural networks are commonly used to build hybrid models [299, 331], with attention mechanisms [303, 300]. Generative models such as variational autoencoder [325] and generative adversarial network

99

are used to generate data to address challenges posed by limited datasets [308, 309, 337], poor quality [338], noisy signals [339], and overfitting [307]. The generative approach employs GAN only or with encoder-decoder [308], where the generator consists of convolutional layers with attention mechanisms or bidirectional LSTM, while the discriminator consists of convolutional layers with attention mechanisms.

Contextual Enhancement: Recent advancements in text classification focus on capturing contextual information utilizing approaches such as joint embeddings of labels [169] and words as well as aspect-aware methods that enhance feature extraction [171]. Conversely, in machine translation, although transformers have proven effective, they often struggle to capture nuanced language features and contexts. To address this issue, various strategies have been introduced such as concatenating contextual sentences and context-sensitive training [179, 180]. In text generation, additional contexts are provided to improve both the quality and diversity of the generated sentences such as incorporating the category context [194] and style context [195] into text generation process.

In image generation, the researchers focus on improving the quality and relevance of generated images by enhancing the alignment between input text and images, while also addressing issues like fuzzy shapes and diversity through innovative deep learning architectures such as incorporating attention mechanisms and fusion modules into the model. Furthermore, recent models like DF-GAN [151] and DMF-GAN [152] showcase a shift towards single-stage generators with regularization strategies that maintain details while enhancing diversity. In text generation, BERT and its variants are the basis of the deep learning models. The research focuses on hierarchical architectures, capturing long text dependencies [193] and leveraging domain knowledge through label sequence [196] and conversation sequence [190] to bridge semantic gaps. Overall, the advancements show a clear trend toward improving contextual awareness and feature representation.

LLMs are increasingly being leveraged in robotics for improving human-robot interaction. The main challenge of integrating LLMs is bridging the gap between robots and the physical world to provide real-world contexts, a process known as LLM grounding. The concept of LLM grounding refers to how machines make sense of abstract language such as words and ideas by connecting them to objects they can actually experience or sense in the real world. To this end, sensors such as cameras and microphones, along with object detection, pose estimation and speech recognition methods, are used

for semantic and perceptual grounding. The robot data are streamed are at varying sampling rates, so they need to be aligned and carefully selected to identify key events [355]. The detected objects and their relationships are then modeled using 3D scene graph for scene understanding [353]. Central to the grounding system is a module that collates and constructs the text prompts which are sent to the LLM [354]. The text prompts are typically formulated by integrating a set of predefined robot skills that can be executed [352].

Loss Function and Optimization: Researchers are increasingly focused on refining loss functions to improve classification accuracy. For instance, in image classification, some studies introduce additive terms to the cross-entropy loss to reward well-classified instances [99], while others propose asymmetric polynomial loss functions that prioritize positive instances to tackle class imbalance [100] and adaptive loss function that dynamically adjust the weights assigned to class-level components based on model performance [398]. Another approach to address class imbalanced is data augmentation through generating virtual samples [399], oversampling with target-aware autoencoders for estimating target values for new features [400]. In time series analysis, similar approach has been proposed such as penalizing misclassification of minority classes [401], maximizing the minimum recall of the classes [402]. Improved training strategies have been proposed to address class imbalanced data such as iteratively selecting the most informative instances [403] and contrastive learning to keep the instances of each fine-grained clusters away from the minority class [404].

Future Directions: Looking ahead, the future of deep learning presents numerous opportunities for growth and innovation. Future models could explore non-sequential hybrid architectures of transformer and convolutional neural networks to leverage the strengths of both approaches, enhancing performance in image classification, object detection and image segmentation. Furthermore, researchers could investigate simpler or lightweight architectures and new training schemes to address the long training time associated with transformers. Future research in image generation may explore advanced frameworks or architectures that further integrate semantic understanding, perhaps by employing hierarchical attention modules and/or fusion modules to capture both local and global features more effectively. Additionally, incorporating unsupervised learning or self-supervised learning approaches could reduce reliance on labeled datasets, allowing models to learn from a diverse range of inputs.

In the area of natural language processing, future research in text classification and machine translation could explore the integration of external knowledge bases and domain-specific embeddings may further improve context understanding and label alignment. Additionally, refining context incorporation methods such as dynamic context updating during translation may enhance the model performance. In text generation, advancing towards more interactive and adaptive systems that can maintain context over extended dialogues and narratives is crucial. Furthermore, frameworks that can leverage multi-modalities (text, audio, and visual) for richer contextual understanding may facilitate the development of more advanced applications across these domains. Future research can also focus on exploring adversarial attacks in these domains and developing tailored defense mechanisms. Furthermore, researchers can further investigate the practical implications of adversarial attacks in real-world scenarios, such as in autonomous vehicles and medical imaging. Understanding the potential impact of adversarial attacks in these applications can inform the development of more robust and secure systems.

Real-time processing capabilities are paramount, future work can develop efficient models or algorithms for real-time processing of HAR, speech recognition, ECG and EEG signals. This could involve optimizing existing architectures and leveraging hardware acceleration techniques to enable real-time inference on resource-constrained devices such as wearable sensors and implantable devices. Furthermore, future models could explore lightweight transformers to enhance feature representation for time series analysis. Due to the sensitive nature of financial research, future work can focus on enhancing the interpretability of deep learning models in financial predictions. Researchers should explore techniques to explain the predictions of models, to improve trust and understanding of model decisions, which is essential for adoption in HAR, speech recognition and finance.

Future research in data generation for time series analysis and pervasive computing should focus on developing more advanced techniques to address data scarcity and improve model generalization. In HAR, frameworks that can generate diverse signals for subjects of different ages, genders, as well as physical abilities, may facilitate the development of models that can generalize across different populations and real-world scenarios [405]. Moreover, the generated data should capture sensor noise, variations in user behavior, and diverse environmental conditions. In finance, synthetic market data should model complex dependencies, volatility patterns and anomalies to improve trend and risk predictions. For EEG and ECG classification, researchers

102

could explore deep generative models that can create physiologically meaningful synthetic signals while preserving individual variability. Future models could explore methods for adapting or personalizing models to account for inter-subject variability and improve performance on individual subjects. Furthermore, EEG electrodes cover only a fraction of the brain's surface, resulting in limited coverage of neural activity. Deep learning models could investigate strategies to infer activity from unobserved brain regions or integrate information from multiple modalities to provide more comprehensive coverage. These areas can still be further explored.

Future research in robot object identification and navigation will focus on enhancing real-time perception, generalization across diverse environments, and increasing robustness to occlusions and dynamic obstacles. Furthermore, researchers can explore multimodal learning, integrating vision with LiDAR and tactile sensing to improve scene understanding. Advances in self-supervised learning and few-shot learning could enable robots to identify and navigate complex environments with minimal labelled data. LLMs will play a crucial role in human-robot interactions, facilitating natural language understanding, intent recognition and context-aware decision-making [406]. Researchers should optimize LLMs for real-time processing on robots and improve their grounding in physical environments. Additionally, future work can explore a deep learning-based navigation with LLMs to develop more autonomous, assistive and socially intelligent robots.

The future of AI lies beyond deep learning, moving towards a conscious intelligent system that mimics human cognition [407]. Achieving this requires a significant advancement in hardware and software architectures, enabling systems to process information with greater efficiency, adaptability and reasoning capabilities. Beyond mere pattern recognition, future of AI will need to integrate symbolic reasoning, causal inference and common sense to truly emulate human thought [408, 409, 410]. Future research will likely focus on hybrid AI models that combine deep learning with symbolic AI, enabling machines to understand abstract concepts, reason about cause and effect and apply knowledge across domains. Quantum computing, with its unique property of entanglement could exponentially accelerate deep learning training and inference [411, 412]. Future models could explore quantum neural networks which could process high dimensional data more effectively and discover patterns that are not possible by classical methods. Quantum-based optimizers could also solve problems that are infeasible for classical graphical processing units or tensor processing units. A more radical approach is

neuromorphic computing that uses specialized hardware to mimic the structure and function of the human brain, enabling low-power inference, parallel processing and real-time learning and adaptation [413, 414]. This results in autonomous AI systems that can solve complex tasks requiring quick response time and adapt to changing situations. Finally, multi-modal intelligence enables AI to process and integrate multiple types of sensory input, including vision, speech and touch, providing a more holistic view of the world, similar to human perception [415].

7. Conclusion

Deep learning has become the prominent data-driven approach in various state-of-the-art applications. Its importance lies in its ability to revolutionize many aspects of research and industries and tackle complex problems which were once impossible to overcome. Numerous surveys have been published on deep learning, reviewing the concepts, model architectures and applications. However, the studies do not discuss the emerging trends in the state-of-the-art applications of deep learning and emphasize the important traits and elements in the models. This paper presents a structured and comprehensive survey of deep learning, focusing on the latest trends and advancements in state-of-the-art applications such as computer vision, natural language processing, time series analysis and pervasive computing, and robotics. It explores key elements and traits in modern deep learning models, highlighting their significance in addressing complex challenges across diverse domains. The discussion also covers future research directions in advancing these fields. Furthermore, this paper presents a comprehensive review of the deep learning fundamentals, which is essential for understanding the core principles behind modern deep learning models. The survey finishes by discussing the critical challenges and future directions in deep learning.

Acknowledgements

Statements and Declarations

- **Funding** This work was supported by the Ministry of Higher Education Malaysia (FRGS/1/2023/ICT02/USM/02/2).

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

- **Ethics approval** Not applicable.

- **Availability of Data and Materials** No data was used for the research described in the article.

Appendix text.

References

[1] S. Shamshirband, M. Fathi, A. Dehzangi, A. T. Chronopoulos, H. Alinejad-Rokny, A review on deep learning approaches in healthcare systems: Taxonomies, challenges, and open issues 113 103627. `doi:10.1016/j.jbi.2020.103627`.
URL `https://www.sciencedirect.com/science/article/pii/S1532046420302550`

[2] J. Wang, Y. Ma, L. Zhang, R. X. Gao, D. Wu, Deep learning for smart manufacturing: Methods and applications 48 144–156. `doi:10.1016/j.jmsy.2018.01.003`.
URL `https://www.sciencedirect.com/science/article/pii/S0278612518300037`

[3] H. A. Pierson, M. S. Gashler, Deep learning in robotics: a review of recent research 31 (16) 821–835, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01691864.2017.1365009. `doi:10.1080/01691864.2017.1365009`.
URL `https://doi.org/10.1080/01691864.2017.1365009`

[4] P. Dixit, S. Silakari, Deep Learning Algorithms for Cybersecurity Applications: A Technological and Status Review 39 100317. `doi:10.1016/j.cosrev.2020.100317`.
URL `https://www.sciencedirect.com/science/article/pii/S1574013720304172`

[5] AlphaFold (Feb. 2025).
    URL https://deepmind.google/technologies/alphafold/

[6] S. Dong, P. Wang, K. Abbas, A survey on deep learning and its applications 40 100379, publisher: Elsevier.

[7] T. Talaei Khoei, H. Ould Slimane, N. Kaabouch, Deep learning: Systematic review, models, challenges, and research directions 35 (31) 23103–23124, publisher: Springer.

[8] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaria, M. A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions 8 1–74, publisher: Springer.

[9] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. Awwal, V. K. Asari, A state-of-the-art survey on deep learning theory and architectures 8 (3) 292, publisher: Multidisciplinary Digital Publishing Institute.

[10] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, S. S. Iyengar, A survey on deep learning: Algorithms, techniques, and applications 51 (5) 1–36, publisher: ACM New York, NY, USA.

[11] I. H. Sarker, Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions 2 (6) 420, publisher: Springer.

[12] Y. LeCun, Y. Bengio, G. Hinton, Deep learning 521 (7553) 436–444, number: 7553 Publisher: Nature Publishing Group. doi:10.1038/nature14539.
    URL https://www.nature.com/articles/nature14539

[13] poloclub/cnn-explainer, original-date: 2019-11-03T23:15:24Z (Mar. 2025).
    URL https://github.com/poloclub/cnn-explainer

[14] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-Excitation Networks, arXiv:1709.01507 [cs] version: 4. doi:10.48550/arXiv.1709.

01507.
URL http://arxiv.org/abs/1709.01507

[15] Z. Gao, J. Xie, Q. Wang, P. Li, Global Second-order Pooling Convolutional Networks, arXiv:1811.12006 [cs] version: 2. doi:10.48550/arXiv.1811.12006.
URL http://arxiv.org/abs/1811.12006

[16] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks, arXiv:1910.03151 [cs]. doi:10.48550/arXiv.1910.03151.
URL http://arxiv.org/abs/1910.03151

[17] Z. Liu, L. Wang, W. Wu, C. Qian, T. Lu, TAM: Temporal Adaptive Module for Video Recognition, arXiv:2005.06803 [cs]. doi:10.48550/arXiv.2005.06803.
URL http://arxiv.org/abs/2005.06803

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, arXiv:1706.03762 [cs]. doi:10.48550/arXiv.1706.03762.
URL http://arxiv.org/abs/1706.03762

[19] poloclub/transformer-explainer, original-date: 2024-05-16T21:42:10Z (Mar. 2025).
URL https://github.com/poloclub/transformer-explainer

[20] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, D. Rueckert, Attention U-Net: Learning Where to Look for the Pancreas, arXiv:1804.03999 [cs]. doi:10.48550/arXiv.1804.03999.
URL http://arxiv.org/abs/1804.03999

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, arXiv:2010.11929 [cs]. doi:10.48550/arXiv.2010.11929.
URL http://arxiv.org/abs/2010.11929

[22] M.-H. Guo, Z.-N. Liu, T.-J. Mu, S.-M. Hu, Beyond Self-Attention: External Attention Using Two Linear Layers for Visual Tasks 45 (5) 5436–5447, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. `doi:10.1109/TPAMI.2022.3211006`.
URL `https://ieeexplore.ieee.org/document/9912362`

[23] Y. A. LeCun, L. Bottou, G. B. Orr, K.-R. Müller, Efficient BackProp, Springer Berlin Heidelberg, pp. 9–48. `doi:10.1007/978-3-642-35289-8_3`.
URL `https://doi.org/10.1007/978-3-642-35289-8_3`

[24] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, in: A Field Guide to Dynamical Recurrent Networks, A field guide to dynamical recurrent neural networks. IEEE Press In.

[25] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, R. Garcia, Incorporating Second-Order Functional Knowledge for Better Option Pricing, in: Advances in Neural Information Processing Systems, Vol. 13, MIT Press.
URL `https://papers.nips.cc/paper_files/paper/2000/hash/44968aece94f667e4095002d140b5896-Abstract.html`

[26] X. Glorot, A. Bordes, Y. Bengio, Deep Sparse Rectifier Neural Networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, pp. 315–323, iSSN: 1938-7228.
URL `https://proceedings.mlr.press/v15/glorot11a.html`

[27] A. L. Maas, A. Y. Hannun, A. Y. Ng, et al., Rectifier nonlinearities improve neural network acoustic models, in: Proc. icml, Vol. 30, Atlanta, GA, p. 3, issue: 1.

[28] D. Misra, Mish: A Self Regularized Non-Monotonic Activation Function, arXiv:1908.08681 [cs, stat]. `doi:10.48550/arXiv.1908.08681`.
URL `http://arxiv.org/abs/1908.08681`

[29] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),

arXiv:1511.07289 [cs]. `doi:10.48550/arXiv.1511.07289`.
URL `http://arxiv.org/abs/1511.07289`

[30] Q. Wang, Y. Ma, K. Zhao, Y. Tian, A Comprehensive Survey of Loss Functions in Machine Learning 9 (2) 187–212. `doi:10.1007/s40745-020-00253-5`.
URL `https://doi.org/10.1007/s40745-020-00253-5`

[31] D. S. a. S. Carter, Tensorflow — Neural Network Playground.
URL `http://playground.tensorflow.org`

[32] J. Guo, AI Notes: Initializing neural networks.
URL `https://www.deeplearning.ai/ai-notes/initialization/`

[33] N. Qian, On the momentum term in gradient descent learning algorithms 12 (1) 145–151. `doi:10.1016/S0893-6080(98)00116-6`.
URL `https://www.sciencedirect.com/science/article/pii/S0893608098001166`

[34] J. Duchi, E. Hazan, Y. Singer, Adaptive Subgradient Methods for Online Learning and Stochastic Optimization 12 (61) 2121–2159.
URL `http://jmlr.org/papers/v12/duchi11a.html`

[35] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv:1412.6980 [cs]. `doi:10.48550/arXiv.1412.6980`.
URL `http://arxiv.org/abs/1412.6980`

[36] L. Prechelt, Early Stopping — But When?, in: G. Montavon, G. B. Orr, K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade: Second Edition, Lecture Notes in Computer Science, Springer, pp. 53–67. `doi:10.1007/978-3-642-35289-8_5`.
URL `https://doi.org/10.1007/978-3-642-35289-8_5`

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting 15 (56) 1929–1958.
URL `http://jmlr.org/papers/v15/srivastava14a.html`

[38] S. Park, N. Kwak, Analysis on the Dropout Effect in Convolutional Neural Networks, in: S.-H. Lai, V. Lepetit, K. Nishino, Y. Sato (Eds.), Computer Vision – ACCV 2016, Lecture Notes in Computer

Science, Springer International Publishing, pp. 189–204. `doi:10.1007/` `978-3-319-54184-6_12`.

[39] Z. Liu, Z. Xu, J. Jin, Z. Shen, T. Darrell, Dropout reduces underfitting, in: Proceedings of the 40th International Conference on Machine Learning, Vol. 202 of ICML'23, JMLR.org, pp. 22233–22248.

[40] R. Tibshirani, Regression Shrinkage and Selection via the Lasso 58 (1) 267–288, publisher: [Royal Statistical Society, Wiley].
URL `https://www.jstor.org/stable/2346178`

[41] H. Zou, T. Hastie, Regularization and Variable Selection Via the Elastic Net 67 (2) 301–320. `doi:10.1111/j.1467-9868.2005.00503.x`.
URL `https://doi.org/10.1111/j.1467-9868.2005.00503.x`

[42] K. Nakamura, B.-W. Hong, Adaptive Weight Decay for Deep Neural Networks 7 118857–118865, conference Name: IEEE Access. `doi:10.` `1109/ACCESS.2019.2937139`.
URL `https://ieeexplore.ieee.org/document/8811458`

[43] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, JMLR.org, pp. 448–456.

[44] J. Ba, J. Kiros, G. E. Hinton, Layer Normalization.
URL `https://www.semanticscholar.org/` `paper/Layer-Normalization-Ba-Kiros/` `97fb4e3d45bb098e27e0071448b6152217bd35a5`

[45] G. Cybenko, Approximation by superpositions of a sigmoidal function 2 (4) 303–314. `doi:10.1007/BF02551274`.
URL `https://doi.org/10.1007/BF02551274`

[46] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators 2 (5) 359–366. `doi:10.1016/0893-6080(89)90020-8`.
URL `https://www.sciencedirect.com/science/article/pii/` `0893608089900208`

[47] B. Widrow, D. E. Rumelhart, M. A. Lehr, Neural networks: applications in industry, business and science 37 (3) 93–105. `doi:10.1145/175247.175257`.
URL `https://dl.acm.org/doi/10.1145/175247.175257`

[48] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory 9 (8) 1735–1780. `doi:10.1162/neco.1997.9.8.1735`.
URL `https://doi.org/10.1162/neco.1997.9.8.1735`

[49] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, in: A. Moschitti, B. Pang, W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 1724–1734. `doi:10.3115/v1/D14-1179`.
URL `https://aclanthology.org/D14-1179`

[50] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, W.-c. Woo, Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, arXiv:1506.04214 [cs] version: 2. `doi:10.48550/arXiv.1506.04214`.
URL `http://arxiv.org/abs/1506.04214`

[51] G. Van Houdt, C. Mosquera, G. Nápoles, A review on the long short-term memory model, Artificial Intelligence Review 53 (8) (2020) 5929–5955. `doi:10.1007/s10462-020-09838-1`.
URL `https://doi.org/10.1007/s10462-020-09838-1`

[52] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: Advances in Neural Information Processing Systems, Vol. 25, Curran Associates, Inc.
URL `https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html`

[53] M. D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, arXiv:1311.2901 [cs] version: 3. `doi:10.48550/arXiv.1311.2901`.
URL `http://arxiv.org/abs/1311.2901`

[54] M. Lin, Q. Chen, S. Yan, Network In Network, arXiv:1312.4400 [cs] version: 3. doi:10.48550/arXiv.1312.4400.
URL http://arxiv.org/abs/1312.4400

[55] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv:1409.1556 [cs] version: 6. doi:10.48550/arXiv.1409.1556.
URL http://arxiv.org/abs/1409.1556

[56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper with Convolutions, arXiv:1409.4842 [cs] version: 1. doi:10.48550/arXiv.1409.4842.
URL http://arxiv.org/abs/1409.4842

[57] R. K. Srivastava, K. Greff, J. Schmidhuber, Highway Networks, arXiv:1505.00387 [cs]. doi:10.48550/arXiv.1505.00387.
URL http://arxiv.org/abs/1505.00387

[58] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, arXiv:1512.03385 [cs]. doi:10.48550/arXiv.1512.03385.
URL http://arxiv.org/abs/1512.03385

[59] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708.

[60] S. Zagoruyko, N. Komodakis, Wide Residual Networks, arXiv:1605.07146 [cs] version: 4. doi:10.48550/arXiv.1605.07146.
URL http://arxiv.org/abs/1605.07146

[61] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated Residual Transformations for Deep Neural Networks, arXiv:1611.05431 [cs]. doi:10.48550/arXiv.1611.05431.
URL http://arxiv.org/abs/1611.05431

[62] ImageNet.
URL https://www.image-net.org/

[63] CIFAR-10 and CIFAR-100 datasets.
URL https://www.cs.toronto.edu/~kriz/cifar.html

[64] H. Chougrad, H. Zouaki, O. Alheyane, Deep Convolutional Neural Networks for breast cancer screening, Computer Methods and Programs in Biomedicine 157 (2018) 19–30. `doi:10.1016/j.cmpb.2018.01.011`.
URL `https://www.sciencedirect.com/science/article/pii/S0169260717301451`

[65] M. Zhao, Y. Wei, Y. Lu, K. K. L. Wong, A novel U-Net approach to segment the cardiac chamber in magnetic resonance images with ghost artifacts, Computer Methods and Programs in Biomedicine 196 (2020) 105623. `doi:10.1016/j.cmpb.2020.105623`.
URL `https://www.sciencedirect.com/science/article/pii/S0169260720314565`

[66] N. Cinar, A. Ozcan, M. Kaya, A hybrid DenseNet121-UNet model for brain tumor segmentation from MR Images, Biomedical Signal Processing and Control 76 (2022) 103647. `doi:10.1016/j.bspc.2022.103647`.
URL `https://www.sciencedirect.com/science/article/pii/S1746809422001690`

[67] X. Sun, J. Fu, B. Wei, Z. Li, Y. Li, N. Wang, A Self-Attentional ResNet-LightGBM Model for IoT-Enabled Voice Liveness Detection, IEEE Internet of Things Journal 10 (9) (2023) 8257–8270, conference Name: IEEE Internet of Things Journal. `doi:10.1109/JIOT.2022.3230992`.
URL `https://ieeexplore.ieee.org/document/9996089`

[68] L. Dong, C. Wang, G. Yang, Z. Huang, Z. Zhang, C. Li, An Improved ResNet-1d with Channel Attention for Tool Wear Monitor in Smart Manufacturing, Sensors 23 (3) (2023) 1240, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/s23031240`.
URL `https://www.mdpi.com/1424-8220/23/3/1240`

[69] Y. Hou, Z. Wu, X. Cai, T. Zhu, The application of improved densenet algorithm in accurate image recognition, Scientific Reports 14 (1) (2024) 8645, publisher: Nature Publishing Group. `doi:10.1038/s41598-024-58421-z`.
URL `https://www.nature.com/articles/s41598-024-58421-z`

[70] H. Narotamo, M. Dias, R. Santos, A. V. Carreiro, H. Gamboa, M. Silveira, Deep learning for ECG classification: A comparative

study of 1D and 2D representations and multimodal fusion approaches, Biomedical Signal Processing and Control 93 (2024) 106141. `doi:10.1016/j.bspc.2024.106141`.
URL `https://www.sciencedirect.com/science/article/pii/S174680942400199X`

[71] C.-J. Zhang, Yuan-Lu, F.-Q. Tang, H.-P. Cai, Y.-F. Qian, Chao-Wang, Heart failure classification using deep learning to extract spatiotemporal features from ECG, BMC Medical Informatics and Decision Making 24 (1) (2024) 17. `doi:10.1186/s12911-024-02415-4`.
URL `https://doi.org/10.1186/s12911-024-02415-4`

[72] M. H. Mohd Noor, S. Y. Tan, M. N. Ab Wahab, Deep Temporal Conv-LSTM for Activity Recognition, Neural Processing Letters 54 (5) (2022) 4027–4049. `doi:10.1007/s11063-022-10799-5`.
URL `https://doi.org/10.1007/s11063-022-10799-5`

[73] G. E. Hinton, A Practical Guide to Training Restricted Boltzmann Machines, in: G. Montavon, G. B. Orr, K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade: Second Edition, Lecture Notes in Computer Science, Springer, pp. 599–619. `doi:10.1007/978-3-642-35289-8_32`.
URL `https://doi.org/10.1007/978-3-642-35289-8_32`

[74] G. E. Hinton, S. Osindero, Y.-W. Teh, A Fast Learning Algorithm for Deep Belief Nets 18 (7) 1527–1554. `doi:10.1162/neco.2006.18.7.1527`.
URL `https://doi.org/10.1162/neco.2006.18.7.1527`

[75] G. E. Hinton, Deep belief networks 4 (5) 5947. `doi:10.4249/scholarpedia.5947`.
URL `http://www.scholarpedia.org/article/Deep_belief_networks`

[76] J. Romero, J. P. Olson, A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data 2 (4) 045001, publisher: IOP Publishing. `doi:10.1088/2058-9565/aa8072`.
URL `https://dx.doi.org/10.1088/2058-9565/aa8072`

[77] X. Li, T. Zhang, X. Zhao, Z. Yi, Guided autoencoder for dimensionality reduction of pedestrian features 50 (12) 4557–4567. `doi:10.1007/s10489-020-01813-1`.
URL `https://doi.org/10.1007/s10489-020-01813-1`

[78] M. H. Mohd Noor, Feature learning using convolutional denoising autoencoder for activity recognition 33 (17) 10909–10922. `doi:10.1007/s00521-020-05638-4`.
URL `https://doi.org/10.1007/s00521-020-05638-4`

[79] P. Li, Y. Pei, J. Li, A comprehensive survey on design and application of autoencoder in deep learning 138 110176. `doi:10.1016/j.asoc.2023.110176`.
URL `https://www.sciencedirect.com/science/article/pii/S1568494623001941`

[80] A. Ng, et al., Sparse autoencoder 72 (2011) 1–19.

[81] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, X. Glorot, Higher Order Contractive Auto-Encoder, in: D. Gunopulos, T. Hofmann, D. Malerba, M. Vazirgiannis (Eds.), Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science, Springer, pp. 645–660. `doi:10.1007/978-3-642-23783-6_41`.

[82] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th international conference on Machine learning, ICML '08, Association for Computing Machinery, pp. 1096–1103. `doi:10.1145/1390156.1390294`.
URL `https://doi.org/10.1145/1390156.1390294`

[83] M. Chen, Z. Xu, K. Q. Weinberger, F. Sha, Marginalized denoising autoencoders for domain adaptation, in: Proceedings of the 29th International Coference on International Conference on Machine Learning, ICML'12, Omnipress, pp. 1627–1634.

[84] D. P. Kingma, M. Welling, Auto-encoding variational bayes.

[85] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets 27.

[86] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, in: International conference on machine learning, PMLR, pp. 2642–2651.

[87] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, arXiv:1606.03657 [cs, stat] version: 1. doi:10.48550/arXiv.1606.03657.
URL http://arxiv.org/abs/1606.03657

[88] L. Weng, From GAN to WGAN, arXiv:1904.08994 [cs, stat]. doi: 10.48550/arXiv.1904.08994.
URL http://arxiv.org/abs/1904.08994

[89] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive Growing of GANs for Improved Quality, Stability, and Variation, arXiv:1710.10196 [cs, stat] version: 3. doi:10.48550/arXiv.1710.10196.
URL http://arxiv.org/abs/1710.10196

[90] T. Karras, S. Laine, T. Aila, A Style-Based Generator Architecture for Generative Adversarial Networks, arXiv:1812.04948 [cs, stat] version: 3. doi:10.48550/arXiv.1812.04948.
URL http://arxiv.org/abs/1812.04948

[91] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[92] D. W. Otter, J. R. Medina, J. K. Kalita, A survey of the usages of deep learning for natural language processing 32 (2) 604–624. doi: 10.1109/TNNLS.2020.2979670.

[93] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, J. Dean, A guide to deep learning in healthcare 25 (1) 24–29.

[94] M. Soori, B. Arezoo, R. Dastres, Artificial intelligence, machine learning and deep learning in advanced robotics, a review.

[95] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, B. Navarro-Colorado, et al., A systematic review of deep learning approaches to educational data mining 2019.

[96] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition 86 (11) 2278–2324, conference Name: Proceedings of the IEEE. doi:10.1109/5.726791.
URL https://ieeexplore.ieee.org/document/726791

[97] P. Simard, D. Steinkraus, J. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., pp. 958–963. doi:10.1109/ICDAR.2003.1227801.
URL https://ieeexplore.ieee.org/document/1227801

[98] M. Matsugu, K. Mori, Y. Mitari, Y. Kaneda, Subject independent facial expression recognition with robust face detection using a convolutional neural network 16 (5-6) 555–559. doi:10.1016/S0893-6080(03)00115-1.

[99] G. Zhao, W. Yang, X. Ren, L. Li, Y. Wu, X. Sun, Well-classified examples are underestimated in classification with deep neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, pp. 9180–9189, issue: 8.

[100] Y. Huang, J. Qi, X. Wang, Z. Lin, Asymmetric Polynomial Loss for Multi-Label Classification, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 1–5.

[101] W. Park, I. Park, S. Kim, J. Ryu, Robust Asymmetric Loss for Multi-Label Long-Tailed Learning, pp. 2711–2720.
URL https://openaccess.thecvf.com/content/ICCV2023W/CVAMD/html/Park_Robust_Asymmetric_Loss_for_Multi-Label_Long-Tailed_Learning_ICCVW_2023_paper.html

[102] L. Nanni, A. Loreggia, L. Barcellona, S. Ghidoni, Building Ensemble of Deep Networks: Convolutional Networks and Transformers 11 124962–124974, conference Name: IEEE Access. doi:10.1109/ACCESS.2023.

3330442.
URL https://ieeexplore.ieee.org/document/10309107

[103] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, et al., Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, in: International Conference on Machine Learning, PMLR, pp. 23965–23998.

[104] C. Loh, S. Han, S. Sudalairaj, R. Dangovski, K. Xu, F. Wenzel, M. Soljacic, A. Srivastava, Multi-Symmetry Ensembles: Improving Diversity and Generalization via Opposing Symmetries.

[105] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, R. Girshick, Early convolutions help transformers see better 34 30392–30400.

[106] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, CvT: Introducing Convolutions to Vision Transformers, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 22–31, iSSN: 2380-7504. doi:10.1109/ICCV48922.2021.00009.
URL https://ieeexplore.ieee.org/document/9710031?denied=

[107] Z. Peng, Z. Guo, W. Huang, Y. Wang, L. Xie, J. Jiao, Q. Tian, Q. Ye, Conformer: Local Features Coupling Global Representations for Recognition and Detection 45 (8) 9454–9468, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. doi:10.1109/TPAMI.2023.3243048.
URL https://ieeexplore.ieee.org/document/10040235

[108] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, Y. Li, MaxViT: Multi-axis Vision Transformer, in: S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Lecture Notes in Computer Science, Springer Nature Switzerland, pp. 459–479. doi:10.1007/978-3-031-20053-3_27.

[109] Y. Ma, R. Wang, M. Zong, W. Ji, Y. Wang, B. Ye, Convolutional transformer network for fine-grained action recognition 569 127027. doi:10.1016/j.neucom.2023.127027.
URL https://www.sciencedirect.com/science/article/pii/S0925231223011505

[110] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, pp. 580–587.
URL https://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html

[111] R. Girshick, Fast R-CNN, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448, iSSN: 2380-7504. doi:10.1109/ICCV.2015.169.
URL https://ieeexplore.ieee.org/document/7410526

[112] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in: Advances in Neural Information Processing Systems, Vol. 28, Curran Associates, Inc.
URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[113] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, pp. 779–788.
URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

[114] J. Redmon, A. Farhadi, Yolov3: An incremental improvement.

[115] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection.

[116] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single Shot MultiBox Detector, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Lecture Notes in Computer Science, Springer International Publishing, pp. 21–37. doi:10.1007/978-3-319-46448-0_2.

[117] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal Loss for Dense Object Detection 42 (2) 318–327, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. doi:10.1109/TPAMI.2018.2858826.
URL https://ieeexplore.ieee.org/document/8417976

[118] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125.

[119] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10781–10790.

[120] M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, pp. 6105–6114.

[121] N. Bodla, B. Singh, R. Chellappa, L. S. Davis, Soft-NMS–improving object detection with one line of code, in: Proceedings of the IEEE international conference on computer vision, pp. 5561–5569.

[122] S. Liu, D. Huang, Y. Wang, Adaptive NMS: Refining pedestrian detection in a crowd, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 6459–6468.

[123] A. Husham Al-Badri, N. Azman Ismail, K. Al-Dulaimi, G. Ahmed Salman, M. Sah Hj Salam, Adaptive Non-Maximum Suppression for improving performance of *Rumex* detection, Expert Systems with Applications 219 (2023) 119634. `doi:10.1016/j.eswa.2023.119634`.
URL `https://www.sciencedirect.com/science/article/pii/S0957417423001355`

[124] H. Jiang, X. Zhang, S. Xiang, Non-Maximum Suppression Guided Label Assignment for Object Detection in Crowd Scenes, IEEE Transactions on Multimedia 26 (2024) 2207–2218, conference Name: IEEE Transactions on Multimedia. `doi:10.1109/TMM.2023.3293333`.
URL `https://ieeexplore.ieee.org/document/10175568`

[125] S. Shi, Q. Fang, X. Xu, T. Zhao, Similarity Distance-Based Label Assignment for Tiny Object Detection, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2024, pp. 13711–13718, iSSN: 2153-0866. `doi:10.1109/IROS58592.2024.`

10801448.
URL https://ieeexplore.ieee.org/document/10801448

[126] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-End Object Detection with Transformers, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, Lecture Notes in Computer Science, Springer International Publishing, pp. 213–229. doi:10.1007/978-3-030-58452-8_13.

[127] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable DETR: Deformable transformers for end-to-end object detection.

[128] X. Dai, Y. Chen, J. Yang, P. Zhang, L. Yuan, L. Zhang, Dynamic DETR: End-to-End Object Detection With Dynamic Attention, pp. 2988–2997.
URL https://openaccess.thecvf.com/content/ICCV2021/html/Dai_Dynamic_DETR_End-to-End_Object_Detection_With_Dynamic_Attention_ICCV_2021_paper.html

[129] D. Jia, Y. Yuan, H. He, X. Wu, H. Yu, W. Lin, L. Sun, C. Zhang, H. Hu, DETRs with Hybrid Matching, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 19702–19712.

[130] L. Huang, K. Lu, G. Song, L. Wang, S. Liu, Y. Liu, H. Li, Teach-DETR: Better Training DETR With Teachers 45 (12) 15759–15771, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. doi:10.1109/TPAMI.2023.3319387.
URL https://ieeexplore.ieee.org/document/10264211

[131] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

[132] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: Proceedings of the IEEE international conference on computer vision, pp. 1520–1528.

[133] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation 39 (12) 2481–2495, publisher: IEEE.

[134] A. Chaurasia, E. Culurciello, Linknet: Exploiting encoder representations for efficient semantic segmentation, in: 2017 IEEE visual communications and image processing (VCIP), IEEE, pp. 1–4.

[135] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969.

[136] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8759–8768.

[137] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, H. Adam, MaskLab: Instance segmentation by refining object detection with semantic and direction features, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4013–4022.

[138] Q. Liu, Y. Dong, X. Li, Multi-stage context refinement network for semantic segmentation 535 53–63. `doi:10.1016/j.neucom.2023.03.006`.
URL `https://www.sciencedirect.com/science/article/pii/S0925231223002254`

[139] Y. Liu, Y. Chen, P. Lasang, Q. Sun, Covariance Attention for Semantic Segmentation 44 (4) 1805–1818, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. `doi:10.1109/TPAMI.2020.3026069`.
URL `https://ieeexplore.ieee.org/document/9206128`

[140] R. Strudel, R. Garcia, I. Laptev, C. Schmid, Segmenter: Transformer for semantic segmentation, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 7262–7272.

[141] A. Hatamizadeh, H. Yin, G. Heinrich, J. Kautz, P. Molchanov, Global context vision transformers, in: International Conference on Machine Learning, PMLR, pp. 12633–12646.

[142] H. Shi, M. Hayat, J. Cai, Transformer scale gate for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3051–3060.

[143] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks.

[144] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D. N. Metaxas, StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks, in: Proceedings of the IEEE international conference on computer vision, pp. 5907–5915.

[145] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D. N. Metaxas, StackGAN++: Realistic image synthesis with stacked generative adversarial networks 41 (8) 1947–1962, publisher: IEEE.

[146] Z. Zhang, Y. Xie, L. Yang, Photographic text-to-image synthesis with a hierarchically-nested adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6199–6208.

[147] M. Zhu, P. Pan, W. Chen, Y. Yang, DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5802–5810.

[148] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, X. He, AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1316–1324.

[149] J. Sun, Y. Zhou, B. Zhang, ResFPA-GAN: Text-to-image synthesis with generative adversarial network based on residual block feature pyramid attention, in: 2019 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO), IEEE, pp. 317–322.

[150] Y. Cai, X. Wang, Z. Yu, F. Li, P. Xu, Y. Li, L. Li, Dualattn-GAN: Text to Image Synthesis With Dual Attentional Generative Adversarial Network 7 183706–183716, conference Name: IEEE Access. `doi:10.1109/ACCESS.2019.2958864`.
URL `https://ieeexplore.ieee.org/document/8930532?denied=`

[151] M. Tao, H. Tang, F. Wu, X.-Y. Jing, B.-K. Bao, C. Xu, DF-GAN: A simple and effective baseline for text-to-image synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16515–16525.

[152] B. Yang, X. Xiang, W. Kong, J. Zhang, Y. Peng, DMF-GAN: Deep Multimodal Fusion Generative Adversarial Networks for Text-to-Image Synthesis 1–13Conference Name: IEEE Transactions on Multimedia. `doi:10.1109/TMM.2024.3358086`.
URL `https://ieeexplore.ieee.org/document/10413630`

[153] B. Jiang, W. Zeng, C. Yang, R. Wang, B. Zhang, DE-GAN: Text-to-image synthesis with dual and efficient fusion model 83 (8) 23839–23852. `doi:10.1007/s11042-023-16377-8`.
URL `https://doi.org/10.1007/s11042-023-16377-8`

[154] R. A. Calvo, H. A. Ceccatto, Intelligent document classification 4 (5) 411–420, publisher: IOS Press. `doi:10.3233/IDA-2000-4503`.
URL `https://content.iospress.com/articles/intelligent-data-analysis/ida00028`

[155] B. Yu, Z.-b. Xu, C.-h. Li, Latent semantic analysis for text categorization using neural network 21 (8) 900–904. `doi:10.1016/j.knosys.2008.03.045`.
URL `https://www.sciencedirect.com/science/article/pii/S0950705108000993`

[156] G. Arevian, Recurrent Neural Networks for Robust Real-World Text Classification, in: IEEE/WIC/ACM International Conference on Web Intelligence (WI'07), pp. 326–329. `doi:10.1109/WI.2007.126`.
URL `https://ieeexplore.ieee.org/abstract/document/4427112`

[157] J. Huang, Y. Feng, Optimization of Recurrent Neural Networks on Natural Language Processing, in: Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition, ICCPR '19, Association for Computing Machinery, pp. 39–45. `doi:10.1145/3373509.3373573`.
URL `https://dl.acm.org/doi/10.1145/3373509.3373573`

[158] R. Wang, Z. Li, J. Cao, T. Chen, L. Wang, Convolutional Recurrent Neural Networks for Text Classification, in: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–6, iSSN: 2161-4407. `doi:10.1109/IJCNN.2019.8852406`.
URL `https://ieeexplore.ieee.org/document/8852406`

[159] G. Liu, J. Guo, Bidirectional LSTM with attention mechanism and convolutional layer for text classification 337 325–338. `doi:10.1016/j.neucom.2019.01.078`.
URL `https://www.sciencedirect.com/science/article/pii/S0925231219301067`

[160] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding.

[161] W. Li, F. Qi, M. Tang, Z. Yu, Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification 387 63–77. `doi:10.1016/j.neucom.2020.01.006`.
URL `https://www.sciencedirect.com/science/article/pii/S0925231220300254`

[162] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding.

[163] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite bert for self-supervised learning of language representations.

[164] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized bert pretraining approach.

[165] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced bert with disentangled attention.

[166] B. Rodrawangpai, W. Daungjaiboon, Improving text classification with transformers and layer normalization 10 100403. `doi:10.1016/j.mlwa.2022.100403`.
URL `https://www.sciencedirect.com/science/article/pii/S2666827022000792`

[167] H. Murfi, Syamsyuriani, T. Gowandi, G. Ardaneswari, S. Nur-rohmah, BERT-based combination of convolutional and recurrent neural network for indonesian sentiment analysis 151 111112. `doi:10.1016/j.asoc.2023.111112`.
URL `https://www.sciencedirect.com/science/article/pii/S1568494623011304`

[168] S. Hao, P. Zhang, S. Liu, Y. Wang, Sentiment recognition and analysis method of official document text based on BERT–SVM model 35 (35) 24621–24632. `doi:10.1007/s00521-023-08226-4`.
URL `https://doi.org/10.1007/s00521-023-08226-4`

[169] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, L. Carin, Joint embedding of words and labels for text classification.

[170] Y. Yan, F. Liu, X. Zhuang, J. Ju, An R-Transformer_bilstm Model Based on Attention for Multi-label Text Classification 55 (2) 1293–1316. `doi:10.1007/s11063-022-10938-y`.
URL `https://doi.org/10.1007/s11063-022-10938-y`

[171] X. Zhu, Y. Zhu, L. Zhang, Y. Chen, A BERT-based multi-semantic learning model with aspect-aware enhancement for aspect polarity classification 53 (4) 4609–4623. `doi:10.1007/s10489-022-03702-1`.
URL `https://doi.org/10.1007/s10489-022-03702-1`

[172] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks 27.

[173] F. Stahlberg, Neural machine translation: A review 69 343–418.

[174] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate.

[175] M.-T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation.

[176] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation.

[177] L. Lupo, M. Dinarelli, L. Besacier, Encoding Sentence Position in Context-Aware Neural Machine Translation with Concatenation.

[178] E. Rippeth, M. Carpuat, K. Duh, M. Post, Improving Word Sense Disambiguation in Neural Machine Translation with Salient Document Context.

[179] X. Wu, Y. Xia, J. Zhu, L. Wu, S. Xie, T. Qin, A study of BERT for context-aware neural machine translation 111 (3) 917–935, publisher: Springer.

[180] D. Kim, Y. Baek, S. Yang, J. Choo, Towards Formality-Aware Neural Machine Translation by Leveraging Context Information, in: Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 7384–7392.

[181] A. M. Gezmu, A. Nürnberger, Transformers for Low-resource Neural Machine Translation., in: ICAART (1), pp. 459–466.

[182] A. Araabi, C. Monz, Optimizing Transformer for Low-Resource Neural Machine Translation, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, pp. 3429–3435. doi:10.18653/v1/2020.coling-main.304.
URL https://aclanthology.org/2020.coling-main.304

[183] L. S. Meetei, A. Singh, T. D. Singh, S. Bandyopadhyay, Do cues in a video help in handling rare words in a machine translation system under a low-resource setting? 3 100016, publisher: Elsevier.

[184] M. A. Faheem, K. T. Wassif, H. Bayomi, S. M. Abdou, Improving neural machine translation for low resource languages through non-parallel corpora: a case study of Egyptian dialect to modern standard Arabic translation, Scientific Reports 14 (1) (2024) 2265, publisher: Nature Publishing Group. doi:10.1038/s41598-023-51090-4.
URL https://www.nature.com/articles/s41598-023-51090-4

[185] B. Li, Y. Weng, F. Xia, H. Deng, Towards better Chinese-centric neural machine translation for low-resource languages 84 101566, publisher: Elsevier.

[186] Y.-p. Nie, Y. Han, J.-m. Huang, B. Jiao, A.-p. Li, Attention-based encoder-decoder model for answer selection in question answering 18 (4) 535–544, publisher: Springer.

[187] J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, X. Li, Neural generative question answering.

[188] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, D. Jurafsky, Deep reinforcement learning for dialogue generation.

[189] J. Wu, T. Mu, J. Thiyagalingam, J. Y. Goulermas, Building interactive sentence-aware representation based on generative language model for community question answering 389 93–107, publisher: Elsevier.

[190] Z. Li, C. Niu, F. Meng, Y. Feng, Q. Li, J. Zhou, Incremental transformer with deliberation decoder for document grounded conversations.

[191] S. Alrowili, K. Vijay-Shanker, BioM-transformers: building large biomedical language models with BERT, ALBERT and ELECTRA, in: Proceedings of the 20th workshop on biomedical language processing, pp. 221–227.

[192] S. Alrowili, K. Vijay-Shanker, Exploring Biomedical Question Answering with BioM-Transformers At BioASQ10B challenge: Findings and Techniques., in: CLEF (Working Notes), pp. 222–234.

[193] T. Ma, Q. Pan, H. Rong, Y. Qian, Y. Tian, N. Al-Nabhan, T-BERTSum: Topic-aware text summarization based on bert 9 (3) 879–890, publisher: IEEE.

[194] X. Li, K. Mao, F. Lin, Z. Feng, Feature-aware conditional GAN for category text generation, Neurocomputing 547 (2023) 126352. `doi:10.1016/j.neucom.2023.126352`.
URL `https://www.sciencedirect.com/science/article/pii/S0925231223004757`

[195] N. Kwon, Y. Yoo, B. Lee, Class conditioned text generation with style attention mechanism for embracing diversity, Applied Soft Computing 163 (2024) 111893. `doi:10.1016/j.asoc.2024.111893`.
URL `https://www.sciencedirect.com/science/article/pii/S1568494624006677`

[196] Q. Xie, J. A. Bishop, P. Tiwari, S. Ananiadou, Pre-trained language models with domain knowledge for biomedical extractive summarization 252 109460, publisher: Elsevier.

[197] O. Hajipoor, A. Nickabadi, M. M. Homayounpour, GPTGAN: Utilizing the GPT language model and GAN to enhance adversarial text generation, Neurocomputing 617 (2025) 128865. `doi:10.1016/j.neucom.2024.128865.`
URL `https://www.sciencedirect.com/science/article/pii/S0925231224016369`

[198] M. Weiser, The computer for the 21 st century 265 (3) 94–105.

[199] Y. He, S. Nazir, B. Nie, S. Khan, J. Zhang, Developing an efficient deep learning-based trusted model for pervasive computing using an lstm-based classification model 2020 1–6.

[200] A. O. Ige, M. H. M. Noor, A survey on unsupervised learning for wearable sensor-based activity recognition 109363.

[201] R. Singh, S. Srivastava, Stock prediction using deep learning 76 18569–18584.

[202] X. Zhang, Y. Han, W. Xu, Q. Wang, Hoba: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture 557 302–316.

[203] Y. Lei, Q. Peng, Y. Shen, Deep learning for algorithmic trading: enhancing macd strategy, in: Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence, pp. 51–57.

[204] H.-B. Zhang, Y.-X. Zhang, B. Zhong, Q. Lei, L. Yang, J.-X. Du, D.-S. Chen, A comprehensive survey of vision-based human action recognition methods 19 (5) 1005.

[205] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, H. Moon, Sensor-based and vision-based human activity recognition: A comprehensive survey 108 107561.

[206] W. Gao, L. Zhang, Q. Teng, J. He, H. Wu, Danhar: Dual attention network for multimodal human activity recognition using wearable sensors 111 107728.

[207] S. Gupta, Deep learning based human activity recognition (har) using wearable sensor data 1 (2) 100046.

[208] C. B. Erdas, S. Guney, Human activity recognition by using different deep learning approaches for wearable sensors 53 1795–1809.

[209] M. G. Ragab, S. J. Abdulkadir, N. Aziz, Random search one dimensional cnn for human activity recognition, in: 2020 International Conference on Computational Intelligence (ICCI), IEEE, pp. 86–91.

[210] K. Banjarey, S. P. Sahu, D. K. Dewangan, Human activity recognition using 1d convolutional neural network, in: Sentimental Analysis and Deep Learning: Proceedings of ICSADL 2021, Springer, pp. 691–702.

[211] A. M. A. Baraka, M. H. Mohd Noor, Similarity Segmentation Approach for Sensor-Based Activity Recognition, IEEE Sensors Journal 23 (17) (2023) 19704–19716, conference Name: IEEE Sensors Journal. `doi: 10.1109/JSEN.2023.3295778`.
URL `https://ieeexplore.ieee.org/document/10188611`

[212] A. Baraka, M. H. Mohd Noor, Deep similarity segmentation model for sensor-based activity recognition, Multimedia Tools and Applications (May 2024). `doi:10.1007/s11042-024-18933-2`.
URL `https://doi.org/10.1007/s11042-024-18933-2`

[213] M. M. H. Shuvo, N. Ahmed, K. Nouduri, K. Palaniappan, A hybrid approach for human activity recognition with support vector machine and 1d convolutional neural network, in: 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), IEEE, pp. 1–5.

[214] C. Han, L. Zhang, Y. Tang, W. Huang, F. Min, J. He, Human activity recognition using wearable sensors by heterogeneous convolutional neural networks 198 116764.

[215] A. O. Ige, M. H. M. Noor, Wsense: A robust feature learning module for lightweight human activity recognition.

[216] S. Deep, X. Zheng, Hybrid Model Featuring CNN and LSTM Architecture for Human Activity Recognition on Smartphone Sensor Data, in: 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2019, pp. 259–264,

iSSN: 2640-6721. `doi:10.1109/PDCAT46702.2019.00055`.
URL `https://ieeexplore.ieee.org/document/9029136`

[217] Y. J. Luwe, C. P. Lee, K. M. Lim, Wearable Sensor-Based Human Activity Recognition with Hybrid Deep Learning Model, Informatics 9 (3) (2022) 56, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/informatics9030056`.
URL `https://www.mdpi.com/2227-9709/9/3/56`

[218] L.-F. Shi, Z.-Y. Liu, K.-J. Zhou, Y. Shi, X. Jing, Novel Deep Learning Network for Gait Recognition Using Multimodal Inertial Sensors, Sensors 23 (2) (2023) 849, number: 2 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/s23020849`.
URL `https://www.mdpi.com/1424-8220/23/2/849`

[219] N. Dua, S. N. Singh, V. B. Semwal, S. K. Challa, Inception inspired CNN-GRU hybrid network for human activity recognition, Multimedia Tools and Applications 82 (4) (2023) 5369–5403. `doi:10.1007/s11042-021-11885-x`.
URL `https://doi.org/10.1007/s11042-021-11885-x`

[220] H. A. Imran, Q. Riaz, M. Hussain, H. Tahir, R. Arshad, Smart-Wearable Sensors and CNN-BiGRU Model: A Powerful Combination for Human Activity Recognition, IEEE Sensors Journal 24 (2) (2024) 1963–1974, conference Name: IEEE Sensors Journal. `doi:10.1109/JSEN.2023.3338264`.
URL `https://ieeexplore.ieee.org/document/10348511`

[221] O. Nafea, W. Abdul, G. Muhammad, M. Alsulaiman, Sensor-Based Human Activity Recognition with Spatio-Temporal Deep Learning, Sensors 21 (6) (2021) 2141, number: 6 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/s21062141`.
URL `https://www.mdpi.com/1424-8220/21/6/2141`

[222] Z. N. Khan, J. Ahmad, Attention induced multi-head convolutional neural network for human activity recognition, Applied Soft Computing 110 (2021) 107671. `doi:10.1016/j.asoc.2021.107671`.
URL `https://www.sciencedirect.com/science/article/pii/S1568494621005925`

[223] A. O. Ige, M. H. Mohd Noor, A deep local-temporal architecture with attention for lightweight human activity recognition, Applied Soft Computing 149 (2023) 110954. `doi:10.1016/j.asoc.2023.110954`.
URL `https://www.sciencedirect.com/science/article/pii/S1568494623009729`

[224] W. Gao, L. Zhang, Q. Teng, J. He, H. Wu, DanHAR: Dual Attention Network for multimodal human activity recognition using wearable sensors, Applied Soft Computing 111 (2021) 107728. `doi:10.1016/j.asoc.2021.107728`.
URL `https://www.sciencedirect.com/science/article/pii/S1568494621006499`

[225] S. Agac, O. D. Incel, Resource-efficient, sensor-based human activity recognition with lightweight deep models boosted with attention, Computers and Electrical Engineering 117 (2024) 109274. `doi:10.1016/j.compeleceng.2024.109274`.
URL `https://www.sciencedirect.com/science/article/pii/S0045790624002027`

[226] Y. Tang, L. Zhang, Q. Teng, F. Min, A. Song, Triple Cross-Domain Attention on Human Activity Recognition Using Wearable Sensors, IEEE Transactions on Emerging Topics in Computational Intelligence 6 (5) (2022) 1167–1176, conference Name: IEEE Transactions on Emerging Topics in Computational Intelligence. `doi:10.1109/TETCI.2021.3136642`.
URL `https://ieeexplore.ieee.org/document/9669910`

[227] D. Misra, T. Nalamada, A. U. Arasanipalai, Q. Hou, Rotate to Attend: Convolutional Triplet Attention Module, arXiv:2010.03045 [cs] (Nov. 2020). `doi:10.48550/arXiv.2010.03045`.
URL `http://arxiv.org/abs/2010.03045`

[228] D. Chen, S. Yongchareon, E. M.-K. Lai, J. Yu, Q. Z. Sheng, Y. Li, Transformer With Bidirectional GRU for Nonintrusive, Sensor-Based Activity Recognition in a Multiresident Environment, IEEE Internet of Things Journal 9 (23) (2022) 23716–23727, conference Name: IEEE Internet of Things Journal. `doi:10.1109/JIOT.2022.3190307`.
URL `https://ieeexplore.ieee.org/document/9826881`

[229] Y. Sun, X. Xu, X. Tian, L. Zhou, Y. Li, Efficient human activity recognition: A deep convolutional transformer-based contrastive self-supervised approach using wearable sensors, Engineering Applications of Artificial Intelligence 135 (2024) 108705. `doi:10.1016/j.engappai.2024.108705`.
URL `https://www.sciencedirect.com/science/article/pii/S0952197624008637`

[230] E. Lattanzi, L. Calisti, C. Contoli, Are Transformers a Useful Tool for Tiny devices in Human Activity Recognition?, in: Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence, ICAAI '24, Association for Computing Machinery, New York, NY, USA, 2025, pp. 339–344. `doi:10.1145/3704137.3704171`.
URL `https://dl.acm.org/doi/10.1145/3704137.3704171`

[231] M. H. Chan, M. H. M. Noor, A unified generative model using generative adversarial network for activity recognition, Journal of Ambient Intelligence and Humanized Computing 12 (7) (2021) 8119–8128. `doi:10.1007/s12652-020-02548-0`.
URL `https://doi.org/10.1007/s12652-020-02548-0`

[232] A. O. Jimale, M. H. Mohd Noor, Fully Connected Generative Adversarial Network for Human Activity Recognition, IEEE Access 10 (2022) 100257–100266, conference Name: IEEE Access. `doi:10.1109/ACCESS.2022.3206952`.
URL `https://ieeexplore.ieee.org/document/9893100`

[233] M. Lupión, F. Cruciani, I. Cleland, C. Nugent, P. M. Ortigosa, Data Augmentation for Human Activity Recognition With Generative Adversarial Networks, IEEE Journal of Biomedical and Health Informatics 28 (4) (2024) 2350–2361, conference Name: IEEE Journal of Biomedical and Health Informatics. `doi:10.1109/JBHI.2024.3364910`.
URL `https://ieeexplore.ieee.org/document/10432776`

[234] Z. Kia, M. Yadollahzaeh-Tabari, H. Motameni, Human activity recognition by body-worn sensor data using bi-directional generative adversarial networks and frequency analysis techniques, The Journal of Supercomputing 81 (1) (2024) 342. `doi:10.1007/s11227-024-06743-0`.
URL `https://doi.org/10.1007/s11227-024-06743-0`

[235] M. Mohammadzadeh, A. Ghadami, A. Taheri, S. Behzadipour, cGAN-based high dimensional IMU sensor data generation for enhanced human activity recognition in therapeutic activities, Biomedical Signal Processing and Control 103 (2025) 107476. `doi:10.1016/j.bspc.2024.107476`.
URL `https://www.sciencedirect.com/science/article/pii/S1746809424015349`

[236] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, K. Shaalan, Speech recognition using deep neural networks: A systematic review 7 19143–19165. `doi:10.1109/ACCESS.2019.2896880`.

[237] S. S. Tirumala, S. R. Shahamiri, A review on deep learning approaches in speaker identification, in: Proceedings of the 8th International Conference on Signal Processing Systems, Association for Computing Machinery, pp. 142–147. `doi:10.1145/3015166.3015210`.
URL `https://doi.org/10.1145/3015166.3015210`

[238] F. Ye, J. Yang, A deep neural network model for speaker identification 11 (8). `doi:10.3390/app11083603`.
URL `https://www.mdpi.com/2076-3417/11/8/3603`

[239] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, T. Alhussain, Speech emotion recognition using deep learning techniques: A review 7 117327–117345. `doi:10.1109/ACCESS.2019.2936124`.

[240] G. Singh, S. Sharma, V. Kumar, M. Kaur, M. Baz, M. Masud, et al., Spoken language identification using deep learning 2021.

[241] Y. Jiao, M. Tu, V. Berisha, J. M. Liss, Accent identification by combining deep neural networks and recurrent neural networks trained on long and short term features., in: Interspeech, pp. 2388–2392.

[242] H. A. Sánchez-Hevia, R. Gil-Pita, M. Utrilla-Manso, M. Rosa-Zurera, Age group classification and gender recognition from speech with temporal convolutional neural networks 81 (3) 3535–3552.

[243] A. A. Alnuaim, M. Zakariah, C. Shashidhar, W. A. Hatamleh, H. Tarazi, P. K. Shukla, R. Ratna, Speaker gender recognition based on deep neural networks and resnet50 2022 1–13.

[244] R. Srivastava, D. Pandey, Speech recognition using hmm and soft computing 51 1878–1883. `doi:https://doi.org/10.1016/j.matpr.2021.10.097`.

[245] J. Padmanabhan, M. J. J. Premkumar, Machine learning in automatic speech recognition: A survey 32 (4) 240–251. `doi:10.1080/02564602.2015.1010611`.

[246] A. Mukhamadiyev, I. Khujayarov, O. Djuraev, J. Cho, Automatic speech recognition method based on deep learning approaches for uzbek language 22 (10) 3683. `doi:https://doi.org/10.3390/s22103683`.

[247] X. Lu, S. Li, M. Fujimoto, Automatic speech recognition 21–38.

[248] C. Hema, F. P. Garcia Marquez, Emotional speech recognition using cnn and deep learning techniques 211 109492. `doi:https://doi.org/10.1016/j.apacoust.2023.109492`. URL `https://www.sciencedirect.com/science/article/pii/S0003682X23002906`

[249] A. Shewalkar, D. Nyavanandi, S. A. Ludwig, Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru 9 (4) 235–245.

[250] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, N. Jaitly, A comparison of sequence-to-sequence models for speech recognition., in: Interspeech, pp. 939–943.

[251] A. Graves, Sequence transduction with recurrent neural networks.

[252] W. Chan, N. Jaitly, Q. V. Le, O. Vinyals, Listen, attend and spell.

[253] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, S. Bengio, An online sequence-to-sequence model using partial conditioning 29.

[254] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, D. Eck, Online and linear-time attention by enforcing monotonic alignments, in: International conference on machine learning, PMLR, pp. 2837–2846.

[255] H. Sak, M. Shannon, K. Rao, F. Beaufays, Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping., in: Interspeech, Vol. 8, pp. 1298–1302.

[256] G. Szűcs, D. Huszti, Seq2seq deep learning method for summary gener-
ation by lstm with two-way encoder and beam search decoder, in: 2019
IEEE 17th International Symposium on Intelligent Systems and Infor-
matics (SISY), pp. 221–226. `doi:10.1109/SISY47553.2019.9111502`.

[257] Z. Li, J. Cai, S. He, H. Zhao, Seq2seq dependency parsing, in: Pro-
ceedings of the 27th International Conference on Computational Lin-
guistics, pp. 3203–3214.

[258] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen,
Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li,
J. Chorowski, M. Bacchiani, State-of-the-art speech recognition with
sequence-to-sequence models, in: 2018 IEEE International Conference
on Acoustics, Speech and Signal Processing (ICASSP), pp. 4774–4778.
`doi:10.1109/ICASSP.2018.8462105`.

[259] A. Baevski, H. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A
Framework for Self-Supervised Learning of Speech Representations,
arXiv:2006.11477 [cs] (Oct. 2020). `doi:10.48550/arXiv.2006.11477`.
URL `http://arxiv.org/abs/2006.11477`

[260] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey,
I. Sutskever, Robust Speech Recognition via Large-Scale Weak Supervi-
sion, arXiv:2212.04356 [eess] (Dec. 2022). `doi:10.48550/arXiv.2212.`
`04356`.
URL `http://arxiv.org/abs/2212.04356`

[261] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han,
S. Wang, Z. Zhang, Y. Wu, R. Pang, Conformer: Convolution-
augmented Transformer for Speech Recognition, arXiv:2005.08100
[eess] (May 2020). `doi:10.48550/arXiv.2005.08100`.
URL `http://arxiv.org/abs/2005.08100`

[262] A. Stooke, R. Prabhavalkar, K. C. Sim, P. M. Mengibar, Aligner-
Encoders: Self-Attention Transformers Can Be Self-Transducers,
arXiv:2502.05232 [cs] (Feb. 2025). `doi:10.48550/arXiv.2502.05232`.
URL `http://arxiv.org/abs/2502.05232`

[263] W. Zhang, T.-H. Zhang, C. Luo, H. Zhou, C. Yang, X. Qian, X.-C.
Yin, Breaking Through the Spike: Spike Window Decoding for Accel-
erated and Precise Automatic Speech Recognition, arXiv:2501.03257

[eess] (Jan. 2025). `doi:10.48550/arXiv.2501.03257`.
URL `http://arxiv.org/abs/2501.03257`

[264] P. Wang, H. V. hamme, Disentangled-Transformer: An Explainable End-to-End Automatic Speech Recognition Model with Speech Content-Context Separation, arXiv:2411.17846 [eess] (Nov. 2024). `doi: 10.48550/arXiv.2411.17846`.
URL `http://arxiv.org/abs/2411.17846`

[265] S. A. G. Shakhadri, K. KR, K. B. Angadi, Samba-ASR: State-Of-The-Art Speech Recognition Leveraging Structured State-Space Models, arXiv:2501.02832 [cs] (Jan. 2025). `doi:10.48550/arXiv.2501.02832`.
URL `http://arxiv.org/abs/2501.02832`

[266] A. M. Ozbayoglu, M. U. Gudelek, O. B. Sezer, Deep learning for financial applications: A survey 93 106384.

[267] F. Shen, X. Zhao, G. Kou, F. E. Alsaadi, A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique 98 106852.

[268] W. Wang, W. Li, N. Zhang, K. Liu, Portfolio formation with preselection using deep learning from long-term financial data 143 113042.

[269] L. Chen, M. Pelger, J. Zhu, Deep learning in asset pricing 70 (2) 714–750.

[270] M. Ahnouch, L. Elaachak, A. Ghadi, Model risk in financial derivatives and the transformative impact of deep learning: A systematic review, in: The Proceedings of the International Conference on Smart City Applications, Springer, pp. 155–165.

[271] S. Sun, S. Wang, Y. Wei, A new ensemble deep learning approach for exchange rates forecasting and trading 46 101160. `doi:https://doi.org/10.1016/j.aei.2020.101160`.
URL `https://www.sciencedirect.com/science/article/pii/S1474034620301312`

[272] M. Z. Abedin, M. H. Moon, M. K. Hassan, P. Hajek, Deep learning-based exchange rate prediction during the covid-19 pandemic 1–52.

[273] G. Wang, J. Ma, Y. Wang, T. Tao, G. Ren, H. Zhu, SUDF-RS: A new foreign exchange rate prediction method considering the complementarity of supervised and unsupervised deep representation features, Expert Systems with Applications 214 (2023) 119152. `doi:10.1016/j.eswa.2022.119152`.
URL `https://www.sciencedirect.com/science/article/pii/S0957417422021704`

[274] M. Nikou, G. Mansourfar, J. Bagherzadeh, Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms 26 (4) 164–174.

[275] S. Cai, X. Feng, Z. Deng, Z. Ming, Z. Shan, Financial news quantization and stock market forecast research based on cnn and lstm, in: Smart Computing and Communication: Third International Conference, SmartCom 2018, Tokyo, Japan, December 10–12, 2018, Proceedings 3, Springer, pp. 366–375.

[276] M. U. Gudelek, S. A. Boluk, A. M. Ozbayoglu, A deep learning based stock trading model with 2-d cnn trend detection, in: 2017 IEEE symposium series on computational intelligence (SSCI), IEEE, pp. 1–8.

[277] J. Eapen, D. Bein, A. Verma, Novel deep learning model with cnn and bi-directional lstm for improved stock market index prediction, in: 2019 IEEE 9th annual computing and communication workshop and conference (CCWC), IEEE, pp. 0264–0270.

[278] P. Chen, Z. Boukouvalas, R. Corizzo, A deep fusion model for stock market prediction with news headlines and time series data, Neural Computing and Applications 36 (34) (2024) 21229–21271. `doi:10.1007/s00521-024-10303-1`.
URL `https://doi.org/10.1007/s00521-024-10303-1`

[279] L. Mozaffari, J. Zhang, Predictive Modeling of Stock Prices Using Transformer Model, in: Proceedings of the 2024 9th International Conference on Machine Learning Technologies, ICMLT '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 41–48. `doi:10.1145/3674029.3674037`.
URL `https://dl.acm.org/doi/10.1145/3674029.3674037`

[280] S. Li, Z. B. Schulwol, R. Miikkulainen, Transformer Based Time-Series Forecasting for Stock, arXiv:2502.09625 [q-fin] (Jan. 2025). `doi:10.48550/arXiv.2502.09625`.
URL `http://arxiv.org/abs/2502.09625`

[281] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting, arXiv:2012.07436 [cs] (Mar. 2021). `doi:10.48550/arXiv.2012.07436`.
URL `http://arxiv.org/abs/2012.07436`

[282] L. Berti, G. Kasneci, TLOB: A Novel Transformer Model with Dual Attention for Stock Price Trend Prediction with Limit Order Book Data, arXiv:2502.15757 [q-fin] (Feb. 2025). `doi:10.48550/arXiv.2502.15757`.
URL `http://arxiv.org/abs/2502.15757`

[283] X. Liu, H. Wang, Z. Li, L. Qin, Deep learning in ecg diagnosis: A review 227 107187. `doi:https://doi.org/10.1016/j.knosys.2021.107187`.
URL `https://www.sciencedirect.com/science/article/pii/S0950705121004494`

[284] C. W. Tsao, A. W. Aday, Z. I. Almarzooq, C. A. Anderson, P. Arora, C. L. Avery, C. M. Baker-Smith, A. Z. Beaton, A. K. Boehme, A. E. Buxton, et al., Heart disease and stroke statistics—2023 update: a report from the american heart association 147 (8) e93–e621.

[285] M. Llamedo, J. P. Martínez, Heartbeat classification using feature selection driven by database generalization criteria 58 (3) 616–625. `doi:10.1109/TBME.2010.2068048`.

[286] S. M. Mathews, C. Kambhamettu, K. E. Barner, A novel application of deep learning for single-lead ecg classification 99 53–62. `doi:https://doi.org/10.1016/j.compbiomed.2018.05.013`.
URL `https://www.sciencedirect.com/science/article/pii/S0010482518301264`

[287] w. zhu wenliang wenliang, X. Chen, Y. Wang, L. Wang, Arrhythmia recognition and classification using ecg morphology and segment feature analysis 16 (1) 131–138. `doi:10.1109/TCBB.2018.2846611`.

[288] M. P. Desai, G. Caffarena, R. Jevtic, D. G. Márquez, A. Otero, A low-latency, low-power fpga implementation of ecg signal characterization using hermite polynomials 10 (19) 2324.

[289] P. Crippa, A. Curzi, L. Falaschetti, C. Turchetti, et al., Multi-class ecg beat classification based on a gaussian mixture model of karhunen-loève transform 16 (1) 2–1.

[290] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, R. San Tan, A deep convolutional neural network model to classify heartbeats 89 389–396.

[291] U. B. Baloglu, M. Talo, O. Yildirim, R. San Tan, U. R. Acharya, Classification of myocardial infarction with multi-lead ecg signals and deep cnn 122 23–30.

[292] S. Singh, S. K. Pandey, U. Pawar, R. R. Janghel, Classification of ecg arrhythmia using recurrent neural networks 132 1290–1297.

[293] E. Prabhakararao, S. Dandapat, Attentive rnn-based network to fuse 12-lead ecg and clinical features for improved myocardial infarction diagnosis 27 2029–2033.

[294] M. Wang, S. Rahardja, P. Fränti, S. Rahardja, Single-lead ecg recordings modeling for end-to-end recognition of atrial fibrillation with dual-path rnn 79 104067.

[295] S. Sowmya, D. Jose, Contemplate on ecg signals and classification of arrhythmia signals using cnn-lstm deep learning model 24 100558.

[296] H. M. Rai, K. Chatterjee, Hybrid cnn-lstm deep learning model and ensemble technique for automatic detection of myocardial infarction using big ecg data 52 (5) 5366–5384.

[297] R. Banerjee, A. Ghose, K. Muthana Mandana, A hybrid cnn-lstm architecture for detection of coronary artery disease from ecg, in: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. doi:10.1109/IJCNN48605.2020.9207044.

[298] S. Kusuma, K. R. Jothi, ECG signals-based automated diagnosis of congestive heart failure using Deep CNN and LSTM architecture,

Biocybernetics and Biomedical Engineering 42 (1) (2022) 247–257.
`doi:10.1016/j.bbe.2022.02.003`.
URL `https://www.sciencedirect.com/science/article/pii/S0208521622000067`

[299] N. Alamatsaz, L. Tabatabaei, M. Yazdchi, H. Payan, N. Alamatsaz, F. Nasimi, A lightweight hybrid CNN-LSTM explainable model for ECG-based arrhythmia detection, Biomedical Signal Processing and Control 90 (2024) 105884. `doi:10.1016/j.bspc.2023.105884`.
URL `https://www.sciencedirect.com/science/article/pii/S1746809423013174`

[300] C.-Y. Chen, Y.-T. Lin, S.-J. Lee, W.-C. Tsai, T.-C. Huang, Y.-H. Liu, M.-C. Cheng, C.-Y. Dai, Automated ECG classification based on 1D deep learning network, Methods 202 (2022) 127–135. `doi:10.1016/j.ymeth.2021.04.021`.
URL `https://www.sciencedirect.com/science/article/pii/S1046202321001134`

[301] J. Wang, X. Qiao, C. Liu, X. Wang, Y. Liu, L. Yao, H. Zhang, Automated ECG classification using a non-local convolutional block attention module, Computer Methods and Programs in Biomedicine 203 (2021) 106006. `doi:10.1016/j.cmpb.2021.106006`.
URL `https://www.sciencedirect.com/science/article/pii/S016926072100081X`

[302] J. Zhang, A. Liu, M. Gao, X. Chen, X. Zhang, X. Chen, ECG-based multi-class arrhythmia detection using spatio-temporal attention-based convolutional recurrent neural network, Artificial Intelligence in Medicine 106 (2020) 101856. `doi:10.1016/j.artmed.2020.101856`.
URL `https://www.sciencedirect.com/science/article/pii/S0933365719312606`

[303] A. Sun, W. Hong, J. Li, J. Mao, An Arrhythmia Classification Model Based on a CNN-LSTM-SE Algorithm, Sensors 24 (19) (2024) 6306, number: 19 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/s24196306`.
URL `https://www.mdpi.com/1424-8220/24/19/6306`

[304] Y. Huang, W. Liu, Z. Yin, S. Hu, M. Wang, W. Cai, ECG classification based on guided attention mechanism, Computer Methods and Programs in Biomedicine 257 (2024) 108454. `doi:10.1016/j.cmpb.2024.108454`.
URL `https://www.sciencedirect.com/science/article/pii/S0169260724004474`

[305] P. Aghaomidi, G. Wang, ECG-SleepNet: Deep Learning-Based Comprehensive Sleep Stage Classification Using ECG Signals, arXiv:2412.01929 [cs] (Dec. 2024). `doi:10.48550/arXiv.2412.01929`.
URL `http://arxiv.org/abs/2412.01929`

[306] R. Hasani, M. Lechner, A. Amini, D. Rus, R. Grosu, Liquid Time-constant Networks, Proceedings of the AAAI Conference on Artificial Intelligence 35 (9) (2021) 7657–7666, number: 9. `doi:10.1609/aaai.v35i9.16936`.
URL `https://ojs.aaai.org/index.php/AAAI/article/view/16936`

[307] F. Zhu, F. Ye, Y. Fu, Q. Liu, B. Shen, Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network, Scientific Reports 9 (1) (2019) 6734, publisher: Nature Publishing Group. `doi:10.1038/s41598-019-42516-z`.
URL `https://www.nature.com/articles/s41598-019-42516-z`

[308] Y. Yang, T. Lan, Y. Wang, F. Li, L. Liu, X. Huang, F. Gao, S. Jiang, Z. Zhang, X. Chen, Data imbalance in cardiac health diagnostics using CECG-GAN, Scientific Reports 14 (1) (2024) 14767, publisher: Nature Publishing Group. `doi:10.1038/s41598-024-65619-8`.
URL `https://www.nature.com/articles/s41598-024-65619-8`

[309] C. Msigwa, D. Bernard, J. Yun, IoT-driven wearable devices enhancing healthcare: ECG classification with cluster-based GAN and meta-features, Internet of Things 28 (2024) 101405. `doi:10.1016/j.iot.2024.101405`.
URL `https://www.sciencedirect.com/science/article/pii/S2542660524003469`

[310] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter,

W. Burgard, T. Ball, Deep learning with convolutional neural networks for eeg decoding and visualization 38 (11) 5391–5420. doi:https://doi.org/10.1002/hbm.23730.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23730

[311] Z. Gao, W. Dang, X. Wang, X. Hong, L. Hou, K. Ma, M. Perc, Complex networks and deep learning for eeg signal analysis 15 (3) 369–388.

[312] K. K. Ang, C. Guan, Eeg-based strategies to detect motor imagery for control and rehabilitation 25 (4) 392–401. doi:10.1109/TNSRE.2016.2646763.

[313] Z. Shen, G. Li, J. Fang, H. Zhong, J. Wang, Y. Sun, X. Shen, Aberrated multidimensional eeg characteristics in patients with generalized anxiety disorder: A machine-learning based analysis framework 22 (14) 5420.

[314] P. Boonyakitanont, A. Lek-Uthai, K. Chomtho, J. Songsiri, A review of feature extraction and performance evaluation in epileptic seizure detection using eeg 57 101702.

[315] M. Sharma, J. Tiwari, V. Patel, U. R. Acharya, Automated identification of sleep disorder types using triplet half-band filter and ensemble machine learning techniques with eeg signals 10 (13) 1531.

[316] F. Vaquerizo-Villar, G. C. Gutiérrez-Tobal, E. Calvo, D. Álvarez, L. Kheirandish-Gozal, F. Del Campo, D. Gozal, R. Hornero, An explainable deep-learning model to stage sleep states in children and propose novel eeg-related patterns in sleep apnea 165 107419.

[317] A. Modir, S. Shamekhi, P. Ghaderyan, A systematic review and methodological analysis of eeg-based biomarkers of alzheimer's disease 113274.

[318] H. Altaheri, G. Muhammad, M. Alsulaiman, S. U. Amin, G. A. Altuwaijri, W. Abdul, M. A. Bencherif, M. Faisal, Deep learning techniques for classification of electroencephalogram (eeg) motor imagery (mi) signals: A review 35 (20) 14681–14722.

[319] C. Brunner, M. Naeem, R. Leeb, B. Graimann, G. Pfurtscheller, Spatial filtering and selection of optimized components in four class motor imagery eeg data using independent components analysis 28 (8) 957–964.

[320] A. Delorme, T. Sejnowski, S. Makeig, Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis 34 (4) 1443–1449.

[321] A. Jafarifarmand, M. A. Badamchizadeh, Eeg artifacts handling in a real practical brain–computer interface controlled vehicle 27 (6) 1200–1208.

[322] R. Zhang, Q. Zong, L. Dou, X. Zhao, A novel hybrid deep learning scheme for four-class motor imagery classification 16 (6) 066004.

[323] S. Kumar, A. Sharma, K. Mamun, T. Tsunoda, A deep learning approach for motor imagery eeg signal classification, in: 2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), IEEE, pp. 34–39.

[324] N. Tibrewal, N. Leeuwis, M. Alimardani, Classification of motor imagery eeg using deep learning increases performance in inefficient bci users 17 (7) e0268880.

[325] M. Dai, D. Zheng, R. Na, S. Wang, S. Zhang, Eeg classification of motor imagery using a novel deep learning framework 19 (3) 551.

[326] M. Li, W. Zhu, M. Zhang, Y. Sun, Z. Wang, The novel recognition method with optimal wavelet packet and lstm based recurrent neural network, in: 2017 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 584–589. `doi:10.1109/ICMA.2017.8015882`.

[327] F. Li, F. He, F. Wang, D. Zhang, Y. Xia, X. Li, A novel simplified convolutional neural network classification algorithm of motor imagery eeg signals based on deep learning 10 (5) 1605.

[328] J. Hwang, S. Park, J. Chi, Improving Multi-Class Motor Imagery EEG Classification Using Overlapping Sliding Window and Deep

Learning Model, Electronics 12 (5) (2023) 1186, number: 5 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/electronics12051186`.
URL `https://www.mdpi.com/2079-9292/12/5/1186`

[329] Y. Zhao, L. He, Deep Learning in the EEG Diagnosis of Alzheimer's Disease, in: C. Jawahar, S. Shan (Eds.), Computer Vision - ACCV 2014 Workshops, Springer International Publishing, Cham, 2015, pp. 340–353. `doi:10.1007/978-3-319-16628-5_25`.

[330] W. Xia, R. Zhang, X. Zhang, M. Usman, A novel method for diagnosing Alzheimer's disease using deep pyramid CNN based on EEG signals, Heliyon 9 (4) (2023) e14858. `doi:10.1016/j.heliyon.2023.e14858`.
URL `https://www.sciencedirect.com/science/article/pii/S2405844023020650`

[331] A. T. Hermawan, I. A. E. Zaeni, A. P. Wibawa, G. Gunawan, W. H. Hendrawan, Y. Kristian, A Multi Representation Deep Learning Approach for Epileptic Seizure Detection, Journal of Robotics and Control (JRC) 5 (1) (2024) 187–204, number: 1. `doi:10.18196/jrc.v5i1.20870`.
URL `https://journal.umy.ac.id/index.php/jrc/article/view/20870`

[332] A. H. Abdulwahhab, A. H. Abdulaal, A. H. Thary Al-Ghrairi, A. A. Mohammed, M. Valizadeh, Detection of epileptic seizure using EEG signals analysis based on deep learning techniques, Chaos, Solitons & Fractals 181 (2024) 114700. `doi:10.1016/j.chaos.2024.114700`.
URL `https://www.sciencedirect.com/science/article/pii/S0960077924002522`

[333] P. Pandey, K. R. Seeja, Subject independent emotion recognition from EEG using VMD and deep learning, Journal of King Saud University - Computer and Information Sciences 34 (5) (2022) 1730–1738. `doi:10.1016/j.jksuci.2019.11.003`.
URL `https://www.sciencedirect.com/science/article/pii/S1319157819309991`

[334] A. Hassouneh, A. M. Mutawa, M. Murugappan, Development of a Real-Time Emotion Recognition System Using Facial Expressions

and EEG based on machine learning and deep neural network methods, Informatics in Medicine Unlocked 20 (2020) 100372. `doi:10.1016/j.imu.2020.100372`.
URL `https://www.sciencedirect.com/science/article/pii/S235291482030201X`

[335] J. Pan, W. Fang, Z. Zhang, B. Chen, Z. Zhang, S. Wang, Multimodal Emotion Recognition Based on Facial Expressions, Speech, and EEG, IEEE Open Journal of Engineering in Medicine and Biology 5 (2024) 396–403, conference Name: IEEE Open Journal of Engineering in Medicine and Biology. `doi:10.1109/OJEMB.2023.3240280`.
URL `https://ieeexplore.ieee.org/document/10026861`

[336] S. Wang, J. Qu, Y. Zhang, Y. Zhang, Multimodal Emotion Recognition From EEG Signals and Facial Expressions, IEEE Access 11 (2023) 33061–33068, conference Name: IEEE Access. `doi:10.1109/ACCESS.2023.3263670`.
URL `https://ieeexplore.ieee.org/document/10089483`

[337] J. Song, Q. Zhai, C. Wang, J. Liu, EEGGAN-Net: enhancing EEG signal classification through data augmentation, Frontiers in Human Neuroscience 18, publisher: Frontiers (Jun. 2024). `doi:10.3389/fnhum.2024.1430086`.
URL `https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2024.1430086/full`

[338] I. Corley, Y. Huang, Deep EEG Super-Resolution: Upsampling EEG Spatial Resolution with Generative Adversarial Networks, arXiv:2502.08803 [cs] (Feb. 2025). `doi:10.48550/arXiv.2502.08803`.
URL `http://arxiv.org/abs/2502.08803`

[339] Y. Cai, Z. Meng, D. Huang, DHCT-GAN: Improving EEG Signal Quality with a Dual-Branch Hybrid CNN–Transformer Network, Sensors 25 (1) (2025) 231, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/s25010231`.
URL `https://www.mdpi.com/1424-8220/25/1/231`

[340] H. T. N. Le, H. Q. T. Ngo, Application of the vision-based deep learning technique for waste classification using the robotic manipulation

system, International Journal of Cognitive Computing in Engineering 6 (2025) 391–400. `doi:10.1016/j.ijcce.2025.02.005`.
URL `https://www.sciencedirect.com/science/article/pii/S2666307425000154`

[341] A. M. Vukicevic, M. Petrovic, N. Jurisevic, M. Djapan, N. Knezevic, A. Novakovic, K. Jovanovic, Versatile waste sorting in small batch and flexible manufacturing industries using deep learning techniques, Scientific Reports 15 (1) (2025) 3756, publisher: Nature Publishing Group. `doi:10.1038/s41598-025-87226-x`.
URL `https://www.nature.com/articles/s41598-025-87226-x`

[342] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment Anything, arXiv:2304.02643 [cs] (Apr. 2023). `doi:10.48550/arXiv.2304.02643`.
URL `http://arxiv.org/abs/2304.02643`

[343] W. Liu, J. Chen, Z. Lyu, R. Feng, T. Hu, L. Deng, Automatic tile position and orientation detection combining deep-learning and rule-based computer vision algorithms, Automation in Construction 171 (2025) 106001. `doi:10.1016/j.autcon.2025.106001`.
URL `https://www.sciencedirect.com/science/article/pii/S092658052500041X`

[344] L. Ge, A. Sadhu, Deep learning-enhanced smart ground robotic system for automated structural damage inspection and mapping, Automation in Construction 170 (2025) 105951. `doi:10.1016/j.autcon.2024.105951`.
URL `https://www.sciencedirect.com/science/article/pii/S0926580524006873`

[345] Y. Dai, D. Kim, K. Lee, An Advanced Approach to Object Detection and Tracking in Robotics and Autonomous Vehicles Using YOLOv8 and LiDAR Data Fusion, Electronics 13 (12) (2024) 2250, number: 12 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/electronics13122250`.
URL `https://www.mdpi.com/2079-9292/13/12/2250`

[346] A. Alotaibi, H. Alatawi, A. Binnouh, L. Duwayriat, T. Alhmiedat, O. M. Alia, Deep Learning-Based Vision Systems for Robot Semantic Navigation: An Experimental Study, Technologies 12 (9) (2024) 157, number: 9 Publisher: Multidisciplinary Digital Publishing Institute. doi:10.3390/technologies12090157.
URL https://www.mdpi.com/2227-7080/12/9/157

[347] O. Mısır, Drivable path detection for a mobile robot with differential drive using a deep Learning based segmentation method for indoor navigation, PeerJ Computer Science 10 (2024) e2514, publisher: PeerJ Inc. doi:10.7717/peerj-cs.2514.
URL https://peerj.com/articles/cs-2514

[348] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking Atrous Convolution for Semantic Image Segmentation, arXiv:1706.05587 [cs] version: 3 (Dec. 2017). doi:10.48550/arXiv.1706.05587.
URL http://arxiv.org/abs/1706.05587

[349] Z. Cao, C. Gong, J. Meng, L. Liu, Y. Rao, W. Hou, Orchard Vision Navigation Line Extraction Based on YOLOv8-Trunk Detection, IEEE Access 12 (2024) 104126–104137, conference Name: IEEE Access. doi:10.1109/ACCESS.2024.3422422.
URL https://ieeexplore.ieee.org/document/10583867

[350] H. Liu, X. Zeng, Y. Shen, J. Xu, Z. Khan, A Single-Stage Navigation Path Extraction Network for agricultural robots in orchards, Computers and Electronics in Agriculture 229 (2025) 109687. doi:10.1016/j.compag.2024.109687.
URL https://www.sciencedirect.com/science/article/pii/S0168169924010780

[351] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, T. Liu, A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions, ACM Trans. Inf. Syst. 43 (2) (2025) 42:1–42:55. doi:10.1145/3703155.
URL https://doi.org/10.1145/3703155

[352] K. Asuzu, H. Singh, M. Idrissi, Human–robot interaction through joint robot planning with large language models, Intelligent Service Robotics

(Jan. 2025). `doi:10.1007/s11370-024-00570-1`.
URL `https://doi.org/10.1007/s11370-024-00570-1`

[353] Y. Chen, G. Zhang, Y. Zhang, H. Xu, P. Zhi, Q. Li, S. Huang, SYNERGAI: Perception Alignment for Human-Robot Collaboration, arXiv:2409.15684 [cs] (Sep. 2024). `doi:10.48550/arXiv.2409.15684`.
URL `http://arxiv.org/abs/2409.15684`

[354] P. Allgeuer, H. Ali, S. Wermter, When Robots Get Chatty: Grounding Multimodal Human-Robot Conversation and Collaboration, Vol. 15019, 2024, pp. 306–321, arXiv:2407.00518 [cs]. `doi:10.1007/978-3-031-72341-4_21`.
URL `http://arxiv.org/abs/2407.00518`

[355] Z. Wang, B. Liang, V. Dhat, Z. Brumbaugh, N. Walker, R. Krishna, M. Cakmak, I Can Tell What I am Doing: Toward Real-World Natural Language Grounding of Robot Experiences, arXiv:2411.12960 [cs] version: 1 (Nov. 2024). `doi:10.48550/arXiv.2411.12960`.
URL `http://arxiv.org/abs/2411.12960`

[356] S. Tonekaboni, S. Joshi, M. D. McCradden, A. Goldenberg, What clinicians want: contextualizing explainable machine learning for clinical end use, in: Machine learning for healthcare conference, PMLR, pp. 359–380.

[357] E. Tjoa, H. J. Khok, T. Chouhan, C. Guan, Enhancing the confidence of deep learning classifiers via interpretable saliency maps 562 126825, publisher: Elsevier.

[358] X. Li, Q. Shen, A hybrid framework based on knowledge distillation for explainable disease diagnosis 238 121844, publisher: Elsevier.

[359] C. Termritthikun, A. Umer, S. Suwanwimolkul, F. Xia, I. Lee, Explainable knowledge distillation for on-device chest x-ray classificationPublisher: IEEE.

[360] W. Xiong, Z. Xiong, Y. Cui, An Explainable Attention Network for Fine-Grained Ship Classification Using Remote-Sensing Images 60 1–14, conference Name: IEEE Transactions on Geoscience and Remote Sensing. `doi:10.1109/TGRS.2022.3162195`.
URL `https://ieeexplore.ieee.org/abstract/document/9741720`

[361] L. Fernandes, J. N. Fernandes, M. Calado, J. R. Pinto, R. Cerqueira, J. S. Cardoso, Intrinsic Explainability for End-to-End Object DetectionPublisher: IEEE.

[362] M. N. Freire, L. N. de Castro, e-Recruitment recommender systems: a systematic review 63 1–20, publisher: Springer.

[363] R. K. Dass, N. Petersen, M. Omori, T. R. Lave, U. Visser, Detecting racial inequalities in criminal justice: towards an equitable deep learning approach for generating and interpreting racial categories using mugshots 38 (2) 897–918, publisher: Springer.

[364] A. Gicić, D. Donko, A. Subasi, Intelligent credit scoring using deep learning methods 35 (9) e7637.

[365] N. Schaaf, O. de Mitri, H. B. Kim, A. Windberger, M. F. Huber, Towards measuring bias in image classification, in: Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part III 30, Springer, pp. 433–445.

[366] A. Giloni, E. Grolman, T. Hagemann, R. Fromm, S. Fischer, Y. Elovici, A. Shabtai, BENN: Bias Estimation Using a Deep Neural NetworkPublisher: IEEE.

[367] V. Iosifidis, T. N. H. Tran, E. Ntoutsi, Fairness-enhancing interventions in stream classification, in: Database and Expert Systems Applications: 30th International Conference, DEXA 2019, Linz, Austria, August 26–29, 2019, Proceedings, Part I 30, Springer, pp. 261–276.

[368] T. Kehrenberg, Z. Chen, N. Quadrianto, Tuning fairness by balancing target labels 3 33, publisher: Frontiers Media SA.

[369] B. Jain, M. Huber, R. Elmasri, Increasing Fairness in Predictions Using Bias Parity Score Based Loss Function Regularization 36. doi:10.32473/flairs.36.133311.
URL https://journals.flvc.org/FLAIRS/article/view/133311

[370] J. Yang, A. A. Soltan, D. W. Eyre, D. A. Clifton, Algorithmic fairness and bias mitigation for clinical machine learning with deep reinforcement learning 5 (8) 884–894, publisher: Nature Publishing Group UK London.

[371] M. A. K. Raiaan, S. Sakib, N. M. Fahad, A. A. Mamun, M. A. Rahman, S. Shatabda, M. S. H. Mukta, A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks, Decision Analytics Journal 11 (2024) 100470. `doi:10.1016/j.dajour.2024.100470`.
URL `https://www.sciencedirect.com/science/article/pii/S2772662224000742`

[372] L. Li, J. Zhu, M.-T. Sun, Deep learning based method for pruning deep neural networks, in: 2019 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 312–317. `doi:10.1109/ICMEW.2019.00-68`.

[373] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, X.-s. Hua, Quantization networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[374] S. Stanton, P. Izmailov, P. Kirichenko, A. A. Alemi, A. G. Wilson, Does knowledge distillation really work?, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., pp. 6906–6919.
URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/376c6b9ff3bedbbea56751a84fffc10c-Paper.pdf`

[375] H. Ying, M. Song, Y. Tang, S. Xiao, Z. Xiao, Enhancing deep neural network training efficiency and performance through linear prediction, Scientific Reports 14 (1) (2024) 15197, publisher: Nature Publishing Group. `doi:10.1038/s41598-024-65691-0`.
URL `https://www.nature.com/articles/s41598-024-65691-0`

[376] K. Wang, X. Huang, C. Tan, S.-M. Yiu, Z. Chen, X. Lei, Optimizing Neural Network Training: A Markov Chain Approach for Resource Conservation, IEEE Transactions on Artificial Intelligence (2024) 1–14Conference Name: IEEE Transactions on Artificial Intelligence. `doi:10.1109/TAI.2024.3413688`.
URL `https://ieeexplore.ieee.org/document/10557251`

[377] C. Yan, J. Xu, Q. Wang, Z. Peng, A Triple-Inertial Accelerated Alternating Optimization Method for Deep Learning Training,

arXiv:2503.08489 [cs] (Mar. 2025). `doi:10.48550/arXiv.2503.08489`.
URL `http://arxiv.org/abs/2503.08489`

[378] V. V. Krasnoproshin, V. V. Matskevich, Random Search in Neural Networks Training, Pattern Recognition and Image Analysis 34 (2) (2024) 309–316. `doi:10.1134/S105466182470010X`.
URL `https://doi.org/10.1134/S105466182470010X`

[379] N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: A survey 6 14410–14430. `doi:10.1109/ACCESS.2018.2807385`.

[380] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks.

[381] P. Tabacof, J. Tavares, E. Valle, Adversarial images for variational autoencoders.

[382] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, C. Li, Adversarial attacks on deep-learning models in natural language processing: A survey 11 (3) 1–41.

[383] L. Jiang, X. Ma, S. Chen, J. Bailey, Y.-G. Jiang, Black-box adversarial attacks on video recognition models, in: Proceedings of the 27th ACM International Conference on Multimedia, pp. 864–872.

[384] M. Esmaeilpour, P. Cardinal, A. L. Koerich, A robust approach for securing audio classification against adversarial attacks 15 2147–2159.

[385] A. R. Munappy, J. Bosch, H. H. Olsson, A. Arpteg, B. Brinne, Data management for production quality deep learning models: Challenges and solutions 191 111359. `doi:10.1016/j.jss.2022.111359`.
URL `https://www.sciencedirect.com/science/article/pii/S0164121222000905`

[386] A. R. Luca, T. F. Ursuleanu, L. Gheorghe, R. Grigorovici, S. Iancu, M. Hlusneac, A. Grigorovici, Impact of quality, type and volume of data used by deep learning models in the analysis of medical images 29 100911. `doi:10.1016/j.imu.2022.100911`.
URL `https://www.sciencedirect.com/science/article/pii/S2352914822000612`

[387] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning 109 (1) 43–76, publisher: IEEE.

[388] A. Mumuni, F. Mumuni, Data augmentation: A comprehensive survey of modern approaches 16 100258. `doi: 10.1016/j.array.2022.100258`.
URL `https://www.sciencedirect.com/science/article/pii/S2590005622000911`

[389] C. Hu, Z. Sun, C. Li, Y. Zhang, C. Xing, Survey of Time Series Data Generation in IoT 23 (15) 6976, number: 15 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/s23156976`.
URL `https://www.mdpi.com/1424-8220/23/15/6976`

[390] H. Murtaza, M. Ahmed, N. F. Khan, G. Murtaza, S. Zafar, A. Bano, Synthetic data generation: State of the art in health care domain 48 100546. `doi:10.1016/j.cosrev.2023.100546`.
URL `https://www.sciencedirect.com/science/article/pii/S1574013723000138`

[391] D. Li, Z. Wang, Y. Chen, R. Jiang, W. Ding, M. Okumura, A Survey on Deep Active Learning: Recent Advances and New Frontiers, arXiv:2405.00334 [cs] (Jul. 2024). `doi:10.48550/arXiv.2405.00334`.
URL `http://arxiv.org/abs/2405.00334`

[392] D. Cacciarelli, M. Kulahci, Active learning for data streams: a survey, Machine Learning 113 (1) (2024) 185–239. `doi:10.1007/s10994-023-06454-2`.
URL `https://doi.org/10.1007/s10994-023-06454-2`

[393] Z. Tan, D. Li, S. Wang, A. Beigi, B. Jiang, A. Bhattacharjee, M. Karami, J. Li, L. Cheng, H. Liu, Large Language Models for Data Annotation: A Survey, arXiv:2402.13446 [cs] version: 2 (Jun. 2024). `doi:10.48550/arXiv.2402.13446`.
URL `http://arxiv.org/abs/2402.13446`

[394] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, D. Tao, A Survey on Self-Supervised Learning: Algorithms, Applications, and Future

Trends, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (12) (2024) 9052–9071, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. `doi:10.1109/TPAMI.2024.3415112`.
URL `https://ieeexplore.ieee.org/document/10559458`

[395] H. Altabrawee, M. H. Mohd Noor, STCLR: Sparse Temporal Contrastive Learning for Video Representation, Neurocomputing 630 (2025) 129694. `doi:10.1016/j.neucom.2025.129694`.
URL `https://www.sciencedirect.com/science/article/pii/S0925231225003662`

[396] H. Altabrawee, M. H. Mohd Noor, Repeat and learn: Self-supervised visual representations learning by Repeated Scene Localization, Pattern Recognition 156 (2024) 110804. `doi:10.1016/j.patcog.2024.110804`.
URL `https://www.sciencedirect.com/science/article/pii/S0031320324005557`

[397] J. D. Lee, Q. Lei, N. Saunshi, J. Zhuo, Predicting what you already know helps: provable self-supervised learning, in: Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21, Curran Associates Inc., Red Hook, NY, USA, 2021, pp. 309–323.

[398] S. Maldonado, C. Vairetti, K. Jara, M. Carrasco, J. López, Owadapt: An adaptive loss function for deep learning using owa operators 280 111022.

[399] W. Zhu, O. Wu, N. Yang, Irda: Implicit data augmentation for deep imbalanced regression 120873.

[400] S. B. Belhaouari, A. Islam, K. Kassoul, A. Al-Fuqaha, A. Bouzerdoum, Oversampling techniques for imbalanced data in regression 252 124118.

[401] X. Wang, Y. Zhang, N. Bai, Q. Yu, Q. Wang, Class-imbalanced time series anomaly detection method based on cost-sensitive hybrid network 238 122192.

[402] J. Ircio, A. Lojo, U. Mori, S. Malinowski, J. A. Lozano, Minimum recall-based loss function for imbalanced time series classification 35 (10) 10024–10034.

[403] L. Moles, A. Andres, G. Echegaray, F. Boto, Exploring data augmentation and active learning benefits in imbalanced datasets 12 (12) 1898.

[404] Y. Zhu, S. Wang, H. Yu, W. Li, J. Tian, Sfpl: Sample-specific fine-grained prototype learning for imbalanced medical image classification 97 103281.

[405] A. O. Jimale, M. H. Mohd Noor, Subject variability in sensor-based activity recognition, Journal of Ambient Intelligence and Humanized Computing 14 (4) (2023) 3261–3274. doi:10.1007/s12652-021-03465-6.
URL https://doi.org/10.1007/s12652-021-03465-6

[406] C. Zhang, J. Chen, J. Li, Y. Peng, Z. Mao, Large language models for human–robot interaction: A review, Biomimetic Intelligence and Robotics 3 (4) (2023) 100131. doi:10.1016/j.birob.2023.100131.
URL https://www.sciencedirect.com/science/article/pii/S2667379723000451

[407] P. Butlin, R. Long, E. Elmoznino, Y. Bengio, J. Birch, A. Constant, G. Deane, S. M. Fleming, C. Frith, X. Ji, R. Kanai, C. Klein, G. Lindsay, M. Michel, L. Mudrik, M. A. K. Peters, E. Schwitzgebel, J. Simon, R. VanRullen, Consciousness in Artificial Intelligence: Insights from the Science of Consciousness, arXiv:2308.08708 [cs] (Aug. 2023). doi:10.48550/arXiv.2308.08708.
URL http://arxiv.org/abs/2308.08708

[408] G. Marra, S. Dumančić, R. Manhaeve, L. De Raedt, From statistical relational to neurosymbolic artificial intelligence: A survey, Artificial Intelligence 328 (2024) 104062. doi:10.1016/j.artint.2023.104062.
URL https://www.sciencedirect.com/science/article/pii/S0004370223002084

[409] B. C. Colelough, W. Regli, Neuro-Symbolic AI in 2024: A Systematic Review, arXiv:2501.05435 [cs] version: 1 (Jan. 2025). doi:10.48550/

arXiv.2501.05435.
URL http://arxiv.org/abs/2501.05435

[410] B. P. Bhuyan, A. Ramdane-Cherif, R. Tomar, T. P. Singh, Neuro-symbolic artificial intelligence: a survey, Neural Computing and Applications 36 (21) (2024) 12809–12844. doi:10.1007/s00521-024-09960-z.
URL https://doi.org/10.1007/s00521-024-09960-z

[411] H. Wu, J. Zhang, L. Wang, D. Li, D. Kong, Y. Han, A survey on quantum deep learning, The Journal of Supercomputing 81 (4) (2025) 564. doi:10.1007/s11227-025-07083-3.
URL https://doi.org/10.1007/s11227-025-07083-3

[412] M. Klusch, J. Lässig, D. Müssig, A. Macaluso, F. K. Wilhelm, Quantum Artificial Intelligence: A Brief Survey, KI - Künstliche Intelligenz 38 (4) (2024) 257–276. doi:10.1007/s13218-024-00871-8.
URL https://doi.org/10.1007/s13218-024-00871-8

[413] A. Shrestha, H. Fang, Z. Mei, D. P. Rider, Q. Wu, Q. Qiu, A Survey on Neuromorphic Computing: Models and Hardware, IEEE Circuits and Systems Magazine 22 (2) (2022) 6–35, conference Name: IEEE Circuits and Systems Magazine. doi:10.1109/MCAS.2022.3166331.
URL https://ieeexplore.ieee.org/document/9782767

[414] D. Kudithipudi, C. Schuman, C. M. Vineyard, T. Pandit, C. Merkel, R. Kubendran, J. B. Aimone, G. Orchard, C. Mayr, R. Benosman, J. Hays, C. Young, C. Bartolozzi, A. Majumdar, S. G. Cardwell, M. Payvand, S. Buckley, S. Kulkarni, H. A. Gonzalez, G. Cauwenberghs, C. S. Thakur, A. Subramoney, S. Furber, Neuromorphic computing at scale, Nature 637 (8047) (2025) 801–812, publisher: Nature Publishing Group. doi:10.1038/s41586-024-08253-8.
URL https://www.nature.com/articles/s41586-024-08253-8

[415] N. Fei, Z. Lu, Y. Gao, G. Yang, Y. Huo, J. Wen, H. Lu, R. Song, X. Gao, T. Xiang, H. Sun, J.-R. Wen, Towards artificial general intelligence via a multimodal foundation model, Nature Communications 13 (1) (2022) 3094, publisher: Nature Publishing Group. doi:10.1038/s41467-022-30761-2.
URL https://www.nature.com/articles/s41467-022-30761-2