Advanced School on Applied Machine Learning | (SMR 3941)

27 May 2024 - 01 Jun 2024
ICTP, Trieste, Italy

P01 - ADOBBATI Federica

Super-resolution Convolutional Neural Networks for coastal ocean modelling

P02 - ARTOLA VELASCO Ander

Signature of warm dark matter in the cosmological density fields extracted using Machine Learning.

P03 - AYADI Asma

Lung Cancer Classification using Artificial Intelligent Models: Enhancing Precision and Performance

P04 - CALIFANO Federico

Evaluating Filled Rubber Viscoelasticity: A Comparative Analysis between NODEs and Classical Phenomenological Models

P05 - CANDANEDO Julio Jose

XFEL Autonomous Experimentation and Analysis using Bayesian Optimization

P06 - CATONI Josefina

Uncertainty representations in variational inference models of visual perception

P07 - DÍAZ FALOH Cristina

Pruning and generalization capacity of Restricted Boltzmann Machines

P08 - GUPTA Kanan

Nesterov acceleration despite very noisy gradients

P09 - JAFFARI Rabeea

Towards Artificial Intelligence (AI) in CO2 Electrochemical/Biochemical Reduction Catalysis

P10 - JIMOH Abdulganiyu

Deep Reinforcement Learning for Sound Source Localization in Animal Environments

P11 - KDOURI Lahoucine

Predicting activity in brain areas associated with emotion processing using multimodal behavioral signals

P12 - LOHAR Soonh Taj

Transforming Aspect-Based Sentiment Analysis with Fine Tuning BERT and Integrating XAI

P13 - MARZIK Guillermo

Artificial Intelligence acceleration of BNCT dose calculations

P14 - MENG Qingxin

A Q-LEARNING ALGORITHM FOR DISCRETE-TIME LINEAR-QUADRATIC CONTROL WITH RANDOM PARAMETERS OF UNKNOWN DISTRIBUTION: CONVERGENCE

P15 - MUSTAFA Rashed

Deep learning methods to identify facial expression

P16 - NAKACH Fatima-Zahrae

Bagging & Boosting ensembles based deep hybrid architecture for histopathological breast cancer classification

P17 - NWEMADJI TIAKO Arsene Gibbs

On the capacity of a quantum perceptron for storing biased patterns

P18 - PARDO ARAUJO Celsa

Isolated pulsar population synthesis with simulation-based inference

P19 - PASIANOTTO Isac

Gaussian processes for time-series data analysis: case of study for a large dataset

P20 - RUIZ SUAREZ Sofia Helena

HMM and extesions to classify time series data: an application in animal movement

P21 - SAINI Jaspal Kaur

DeepThink IoT: The Strength of Deep Learning in Internet of Things

P22 - SALAZAR Albin

A Kappa Model of Thermotaxis in Juvenile Honeybees

P23 - SHANKAR Anirudh

INFERENCE OF COSMOLOGY FROM THE MORPHOLOGICAL STRUCTURE OF DARK MATTER HALO AND GALAXY DISTRIBUTIONS USING NEURAL NETWORKS

P24 - SIMOU Mohamed Rabii

AI for Spatial Rural-Urban Transformation

P25 - SINGH Kunal Santoshkumar

Refining the Giants: A Comprehensive Review of Fine-Tuning Strategies for Large Language Models

P26 - LI Weikai

Fast Online Few-Shot Object Detection via Prototype Fusion

P01

Super-Resolution Convolutional Neural Networks for coastal ocean modelling

Federica Adobbati[1], **Lorenzo Bonin**[2], **Fabio Giordano**[1,2], **Stefano Querin**[1], **Gianpiero Cossarini**[1] **and Luca Manzoni**[1,2]

[1]*National Institute of Oceanography and Applied Geophysics - OGS*
[2]*University of Trieste*

We present a method based on Convolutional Neural Networks (CNNs) for the coastal super-resolution task of ocean modeling products (both physical and biogeochemical), applied to a marginal sea of the Mediterranean, the northern Adriatic Sea.

CNNs have been successfully used to solve image related tasks, such as super-resolution problems, where the neural network is trained to improve the resolution of a generic digital image. We adapted the super-resolution task to resolve images that are 3D model output, including masked areas (i.e. land points). We trained the super-resolution CNN and tested with two sets of model output at different resolutions: the low resolution ($\sim 4km$) Mediterranean Copernicus Marine Service (CMS) reanalysis, and the high- resolution ($\sim 750m$) reanalysis of the northern Adriatic Sea.

The resolution of CMS data barely resolve all the smaller-scale processes, coastal features, and variability (e.g., mesoscale eddies and meanders, river plumes, fronts) that characterize the sub-regional areas of the basin. The high-resolution ($\sim 750m$) reanalysis was performed in the framework of the CADEAU project [1], covering the period from 2006 to 2017, and tackling both physical and biogeochemical aspects. The resulting high-resolution dataset consists of 876, 5-day average, data points (i.e., 3D model output) for each variable.

Neural networks can be trained to reproduce the coastal patterns learned from such a dataset, starting from the corresponding coarse-resolution CMS products.

We tested different architectures: a simple CNN [2] and a U-net [3], and different options in the selection of the input data: univariate vs multivariate, with vs without details on boundary conditions (e.g., river inputs). We evaluated their performances against standard interpolation methods. Results are promising and showing that:

1. the level of variability observed in the high resolution model is efficiently introduced
2. biases are corrected
3. realistic spatial patterns in front of river mouths are reproduced.

The use of information on river boundaries proved to be important, as regional (global) models with low resolution rarely contain realistic runoff data. Among the list of physical and biogechemical variables, alkalinity and dissolved inorganic carbon perform better, and the use of multivariate approach does not increase the accuracy of the reconstruction.

This work is part of the broader iNEST PNRR project, and contributes to the goal of developing a digital twin of the northern Adriatic Sea.

[1] A. Bruschi, I. Lisi, R. De Angelis, S. Querin, G. Cossarini, V. Di Biagio, S. Salon, C. Solidoro, D. Fassina, S. Ancona, C. Silvestri, J. Environ. Manag. **293**, 112878 (2021).
[2] C. Dong, C. Loy, K. He, X. Tang, IEEE PAMI **38**, 295 (2015).
[3] O. Ronneberger, P. Fischer, T. Brox, MICCAI 2015 **18**, 234 (2015).

P02

Signature of warm dark matter in the cosmological density fields extracted using Machine Learning

Ander Artola[1]

[1] *Heidelberg University, Germany*

The precise nature of Dark Matter (DM) remains an unsolved puzzle in Physics, with various candidates distinguished by their free-streaming velocities, which dictate the scale at which the gravitational clustering of matter is inhibited by the motion of DM particles. While Cold Dark Matter effectively describes many observational features of the Universe, Warm Dark Matter (WDM) models emerge as alternatives at smaller scales. Significant efforts have been made in the literature to constrain the mass of WDM using Lyman-α forest flux statistics, such as the 1D Power Spectrum. This work explores the potential of using Neural Networks to directly extract the intergalactic medium baryon density fields from the Lyman-α forest of high-redshift quasars [Nayak *et al.*(2023)]. Inference on WDM models is carried out by leveraging the Sherwood-Relics simulation suite [Bolton *et al.*(2017)] to generate mock spectra. The networks, trained on Ly-α forest spectra extracted from Sherwood simulations, can successfully recover IGM conditions, even when the spectra are contaminated with realistic noise and instrumental effects. This work shows that the inference predictions on untrained WDM models can recover their masses within 2σ. Applying this Machine Learning approach to high-resolution quasar spectra can potentially provide tighter constraints on the nature of DM, clearly outperforming single summary statistics, such as the Power Spectrum. [Villasenor *et al.*(2023)]

[Bolton *et al.*(2017)] Bolton, J.S., Puchwein, E., Sijacki, D., Haehnelt, M.G., Kim, T.-S., Meiksin, A., and, ...: 2017, *Monthly Notices of the Royal Astronomical Society* **464**, 897. doi:10.1093/mnras/stw2397.

[Nayak *et al.*(2023)] Nayak, P., Walther, M., Gruen, D., and Adiraju, S.: 2023, *arXiv e-prints*, arXiv:2311.02167. doi:10.48550/arXiv.2311.02167.

[Villasenor *et al.*(2023)] Villasenor, B., Robertson, B., Madau, P., and Schneider, E.: 2023, *Physical Review D* **108**, 023502. doi:10.1103/PhysRevD.108.023502.

P03

Lung Cancer Classification using Artificial Intelligent Models: Enhancing Precision and Performance

Asma Ayadi [1,2*], Najeh Ahmed,[1] and Imen Hammami [1,2]

[1] *Tunisian Center for Nuclear Sciences and Technology, Technopark Sidi Thabet 2020, Tunisia*

[2] *Research Laboratory on Energy and Matter for Nuclear Science Development (LR16CNSTN02), Ministery of Higher education and Research, Technopark Sidi Thabet 2020, Tunisia.*

Corresponding author:
ayadiasma@hotmail.com

Background: Lung cancer poses a substantial global health challenge, imposing a significant burden on healthcare systems. Timely detection is paramount for improving patient prognosis and treatment outcomes. The integration of artificial intelligence (AI) into medical imaging processes offers a promising avenue to enhance diagnostic accuracy and efficiency in lung cancer screening. **Objectives:** This study aims to propose a diagnostic aid system for the early detection of lung cancer from CT images, leveraging shape and texture attributes to achieve effective lung nodule classification. Given the pivotal role of early detection in lung cancer outcomes, the integration of AI tools enhances the potential for accurate and timely diagnoses. **Material and Methods:** The proposed methodology consists of three key steps. Firstly, a semantic segmentation step is implemented using the U-Net convolutional neural network. Subsequently, an extraction and selection step of attributes takes place, which are then utilized in the classification step. This classification step is based on a CNN convolutional neural network, ensuring a comprehensive and effective approach to lung nodule detection.
Results: The obtained results demonstrate a high level of accuracy in both the segmentation and classification of lung nodules. The U-Net algorithm for segmentation achieved an impressive accuracy of 99.16% and a Dice Coefficient (DSC) of 88.44%. The classification results for distinguishing nodules from non-nodules in terms of ROI regions achieved an accuracy of 90.36%. Further classifications were performed to distinguish between subsolid nodules (solid or non-solid) with an accuracy of 91.89% and the malignancy of nodules with an accuracy of 91.54%. These promising outcomes open avenues for potential advancements in the early detection and diagnosis of lung cancer using CT images, highlighting the valuable contribution of AI tools in the realm of lung cancer screening. **Conclusion:** In conclusion, this study introduces a robust diagnostic aid system demonstrating high accuracy in the early detection of lung cancer from CT images. The integration of AI, particularly deep learning techniques, proves instrumental in achieving enhanced precision in lung nodule classification. These results underscore the potential of the proposed system as a valuable tool for clinicians and healthcare professionals, contributing to improved healthcare outcomes and paving the way for further advancements in the field of lung cancer diagnosis and treatment.

Keywords: Lung cancer; CT images; Deep learning; Semantic segmentation; Lung nodule classification; U-Net convolutional neural network; CNN convolutional neural network.

P04

Evaluating Filled Rubber Viscoelasticity: A Comparative Analysis between NODEs and Classical Phenomenological Models

Federico Califano[1], **Jacopo Ciambella**[1,2]

[1]*Department of Mechanical and Aerospace Engineering, Sapienza University of Rome, Rome, 00184, Italy, federico.califano@uniroma1.it*
[2]*Department of Structural and Geotechnical Engineering, Sapienza University of Rome, Rome, 00184, Italy, jacopo.ciambella@uniroma1.it*

Recent literature reveals an escalating trend of incorporating data-driven methodologies in the modeling of visco-elasto-plastic materials, especially of deep neural networks (DNNs). These networks are broadly classified into three categories: black box NNs, NNs that incorporate physics in a weak manner, and NNs that enforce physics strongly. In our study, we employ the Neural Ordinary Differential Equations (NODEs), a specialized variant of DNNs. In 2018, Chen et al. introduced NODEs, which utilizes a general multi-layer perceptron (MLP) as the driving component on the right-hand side of a system of ordinary differential equations (ODEs) [1]. This method notably integrates the time-step scaling of dynamics, an element not present in recurrent neural networks (RNNs). NODEs were integrated into the flow of internal states, following the Coleman-Gurtin internal state variable theory [2]. In addition, the development of automatically convex data-driven creep potential functions, utilizing NODEs and focusing strongly on the enforcement of physics, was successfully carried out [3]. It has been demonstrated that the creep potential proposed by Reese and Gonvindjee [4], represents a specific instance within this methodology. Nevertheless, it remains to be established whether it effectively captures the deformation-enhanced shear thinning of the creep potential, a key aspect for accurately describing the Payne effect observed in filled rubber [5]. In this presentation we will assess the capability of NODEs in characterizing the behavior of filled rubber under varying frequencies and strain amplitudes.

[1] Chen, R. T. Q. et al. Neural ordinary differential equations. In Proc. of the 32nd Int. Conf. on NIPS, pp. 6572–6583, Red Hook, NY, USA, 2018. Curran Associates Inc.

[2] Jones, R. E. et al. A neural ordinary differential equation framework for modeling inelastic stress response via internal state variables. J. Mach. Learn. Model. Comput., 3(3):1–35, 2022.

[3] Taç, V. et al. Data-driven anisotropic finite viscoelasticity using neural ordinary differential equations. Comput. Methods Appl. Mech. Eng., 411:116046, 2023.

[4] Reese, S. and Govindjee, S. A theory of finite viscoelasticity and numerical aspects. Int. J. Solids Struct., 35(26):3455–3482, 1998.

[5] Califano, F., and Ciambella, J. Viscoplastic simple shear at finite strains. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2023. https://doi.org/10.1098/rspa.2023.0603

P05

XFEL Autonomous Experimentation and Analysis using Bayesian Optimization

Julio Candanedo[1]

[1]*(Presenting author underlined) Department of Physics, University of Wisconsin-Milwaukee, 53211, United States*

In the last decade, two areas in physics that underwent rapid development were XFELs and Machine-Learning. Naturally, one would inquire if these methods could be used together. It is known that ultrafast time-series data-analysis, after a data-collection experiment, can be achieved using Manifold-learning algorithms such as Diffusion-Map and Nonlinear-Laplacian-Spectrum-Analysis. However, modern XFEL's introduce an additional difficulty; a large amount of generated data, much of which might not be useful. This is especially true, with the advent of rapid 1 Million snapshots-per-second (1 MHz) rate of LCLS-II, 1 kHz snapshots-per-second in the case of CXFEL, and the 20 kHz rate of the EuXFEL. Therefore the need for real-time, during the experiment, data-analysis and autonomous-experimentation to alter experimental conditions to form a complete data-set are desirable. Here we use Bayesian-Optimization techniques to solve this issue within the manifold-learning; and the Gaussian-Process paradigm, building on the work of Marcus Noack. In this work, we compare our results to random and low-discrepancy sampling, to provide a benchmark on future work, on several classes of physically motivated Random-Continuous-Functions. These include a continuous landscape of spectra (generated by Random-Matrices) and diffraction-patterns, as a black-box ground-truth to test our Bayesian-Optimization algorithm.

P06

Uncertainty representations in variational inference models of visual perception

Josefina Catoni[1], **Enzo Ferrante**[1], **Diego H. Milone**[1] **and Rodrigo Echeveste**[1]

[1]*Research Institute for Signals, Systems and Computational Intelligence sinc(i), CONICET-UNL, Santa Fe, Argentina*

Bayes rule provides an optimal way to perform inference in probabilistic scenarios [1], and it is hence a natural tool to understand perception in the context of uncertainty. Indeed, increasing evidence indicates the brain is able to represent and operate with probability distributions to (approximately) perform probabilistic inference in several scenarios [2, 3, 4]. While exact Bayesian inference is often intractable, a popular choice to approximate the process is variational inference [5]. Variational Autoencoders (VAEs) [6] are a useful tool to learn internal probabilistic representations in an unsupervised fashion. This procedure can be useful when modeling an inference process where the generative model is unknown, since in VAEs the encoder and the decoder are simultaneously learned from the data. The encoder or inference model maps inputs to the parameters of latent distributions, and the decoder or generative model maps latent representations back to the original input space. This architecture provides a means to model inference in the cortex by learning from the statistics of stimuli. Indeed, previous work has shown that classical receptive fields emerge when training sparse VAEs [7].
Here we went beyond receptive fields and studied the properties of the posterior distributions of those VAEs, finding a counterintuitive behavior. While the signal mean and signal variance in the latent representations increase with the contrast of the images, as expected since the images and orientations present in them become more and more distinguishable, the reported uncertainty (noise variance) grows. This is counterintuitive as the uncertainty would be expected to decrease as contrast increases, with a blank zero contrast image being maximally uninformative. Taking inspiration from the Gaussian Scale Mixture (GSM) model [8], we incorporate a global multiplicative contrast variable to the generative model of the VAE. The GSM has been shown to capture basic properties of natural image statistics [8], and has been used as a model of cortical visual processing[9, 10]. The hope with this explicit multiplicative variable is to capture the explaining-away phenomenon observed in the GSM. We hence call this model explaining-away VAE (EA-VAE). Our model fixes the aforementioned problems showing decreasing uncertainty with contrast. Importantly, posteriors converge the prior for zero contrast, which in turn matches the average posterior.
Finally, to test whether the EA-VAE had a broader scope of applications than natural images, we studied VAEs for the classic MNIST dataset, as well as for Chest X-Ray images, showing that our model consistently produces better uncertainty estimates, in different conditions, such as interpolation between known examples, and even out of distribution detection.

[1] D.J.C. MacKay, Cambridge University Press (2003).
[2] A. Pouget et al., Nat. Neurosci. **16** (9), 1170-1178 (2013).
[3] J. Fiser et al., Trends Cogn. Sci. **14** (3), 119-130 (2010).
[4] W.J. Ma et al., Nat. Neurosci. **9** (11), 1432-1438 (2006).
[5] D.M. Blei et al., J. Am. Stat. Assoc. **112** (518), 859–877 (2017).
[6] D.P. Kingma, M. Welling, arXiv:1312.6114 (2022).
[7] F. Csikor et al., arXiv:2206.00436 5(2022).
[8] M.J. Wainwright, E. Simoncelli, Adv. Neural Inf. Process. Syst. **12** (1999).
[9] G. Orbán et al., Neuron **92** (2), 530-543 (2016).
[10] R. Echeveste et al., Nat. Neurosci. **23** (9), 1138-1149 (2020).

P07

Pruning and generalization capacity of Restricted Boltzmann Machines for

… Advanced School on Applied Machine Learning …

C. Díaz-Faloh [1] **, R. Mulet** [2]

[1] *Group of Complex Systems and Statistical Physics, Physics Faculty. University of Havana, Cuba*

[2] *Group of Complex Systems and Statistical Physics, Department of Theoretical Physics, Physics Faculty. University of Havana, Cuba*

A Restricted Boltzmann Machine (RBM) is a generative neural network consisting of one layer of visible neurons fully connected to a second layer of hidden neurons, with no connections within the same layer. They are commonly used for unsupervised learning, often serving as an initial or intermediate layer in deeper models. Our investigation focuses on RBMs to understand the impact of pruning on their generalization capacity—the ability of the model to generate realistic and diverse samples that capture the underlying patterns of the training data. Our study presents outcomes from extensive numerical simulations and preliminary analytical findings. The focus is on understanding how pruning influences the learning processes of RBMs and similar models, shedding light on optimizing their structure for improved efficiency maintaining robust generalization capabilities.

P08

Nesterov acceleration despite very noisy gradients

Kanan Gupta[1], **Stephan Wojtowytsch**[2], **and Jonathan Siegel**[2]

[1]*(Presenting author underlined) University of Pittsburgh*
[2]*Texas A&M University*

Momentum-based gradient descent methods use information gained along the trajectory, in addition to the local information from the gradient, in order to achieve an accelerated rate of convergence. These methods have been well-studied for convex optimization. Computing the gradient is often too expensive and it is approximated using stochastic gradient estimates in practice. However, there's a lack of theoretical analyses of accelerated methods in the setting of stochastic gradient descent, even for the simple case of convex functions. We address this gap with a novel descent algorithm which provably achieves the optimal convergence rate for convex optimization. While the objective functions in deep learning training are non-convex, they share many properties with convex functions. Empirical results show that our algorithm outperforms the existing variants of stochastic gradient descent with momentum for training of neural networks.

[1] Gupta K, Siegel J, Wojtowytsch S. Achieving acceleration despite very noisy gradients. arXiv preprint arXiv:2302.05515. 2023 Feb 10.

P09

Abstract template for the Advanced School on Applied Machine Learning

Rabeea Jaffari [1], Sulafa Abdalmageed Sadaldeen [2]

[1] *Software Engineering Department, Mehran University of Engineering and Technology, Pakistan*

[2] *Chemical Engineering Department, Universiti Teknologi PETRONAS, Malaysia*

The advent of industrial revolution has resulted in massive emissions of carbon dioxide (CO_2) leading to global warming [1]. An effective method to reduce the carbon footprint is the cyclic utilization of CO_2 via sustainable development practices. Photosynthesis is one such process, but it has become infeasible due to the frequent deforestations while the CO_2 emissions are still on a constant rise. Alternatively, artificial photosynthesis, or CO_2 electrochemical reduction (CO_2RR) is a reutilization process of converting CO_2 gases present in the environment to value-added products such as fuel, methane, ethylene, formic acid, CO, Hydrogen and so on, using electrical energy [2]. However, the electric potential required to carry out CO_2RR is extremely high and therefore, needs an efficient catalyst to reduce the reaction overpotential [3]. The discovery of viable catalysts for CO_2RR is extremely complex since the reaction mechanisms are still under investigations and due to the complexity of catalytic processes and the rigorous requirements of ideal catalysts; for example, that they are highly efficient, environmentally benign, stable under working conditions and made of earth-abundant elements. In practice, catalysts are discovered through trial-and-error approach paired with chemical intuition which is an extremely challenging and time-consuming task [4]. Traditional density functional theory (DFT) computations have been guiding the exploration of CO_2RR electrocatalysts so far, but it is still challenging to efficiently search for viable electrocatalysts in a large chemical space using only the traditional DFT computations [5].

The recent success of artificial intelligence (AI) in various research areas provides the motivation to utilize it for the prediction of viable electrocatalysts in this research [6, 7]. We propose to develop an AI-based framework to predict the optimal catalyst for the CO_2RR. The framework would involve an investigation of the electronic and geometric properties of the CO_2RR electrocatalysts to formulate a dataset which would be utilized to train a novel AI model for the prediction of the catalyst. The intended dataset would be collected and analysed via COSMO-RS software suite and the AI model would be developed via suitable machine learning or deep learning techniques. The resulting AI model for the prediction of CO_2RR electrocatalysts will be validated against various industry and AI benchmark applications.

This research would serve as a significant step towards the generation of value-added carbon products (fuel, methane, ethylene etc.) via renewable energy sources such as solar or wind power. Hence, it has the potential to guide the design of next-generation renewable energy materials and reduce the effect of global warming.

[1] T. J. Wallington, J. Srinivasan, O. J. Nielsen, and E. J. Highwood, "Greenhouse gases and global warming," Environmental and ecological chemistry, vol. 1, pp. 36-63, 2009.

[2] A. C. Benniston and A. Harriman, "Artificial photosynthesis," Materials Today, vol. 11, pp. 26-34, 2008.

[3] R. Kortlever, J. Shen, K. J. P. Schouten, F. Calle-Vallejo, and M. T. Koper, "Catalysts and reaction pathways for the electrochemical reduction of carbon dioxide," The journal of physical chemistry letters, vol. 6, pp. 4073-4082, 2015.

[4] Z. Li, S. Wang, and H. Xin, "Toward artificial intelligence in catalysis," Nature Catalysis, vol. 1, pp. 641-642, 2018.

[5] X. Zhang and Z. Zhou, "Perspective on Theoretical Models for CO2 Electrochemical Reduction," The Journal of Physical Chemistry C, vol. 126, pp. 3820-3829, 2022.

[6] V. Dignum, "AI is multidisciplinary," AI Matters, vol. 5, pp. 18-21, 2020.

[7] Y. K. Dwivedi, L. Hughes, E. Ismagilova, G. Aarts, C. Coombs, T. Crick, et al., "Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," International Journal of Information Management, vol. 57, p. 101994, 2021.A. Author, B. Coauthor, J. Sci. Res. **13**, 1357 (2012). [2] A. Author, B. Coauthor, J. Sci. Res. **17**, 7531 (2013).

P10

Deep Reinforcement Learning for Sound Source Localization in Animal Environments

Anonymous Author 1*
Anonymous Department 1
Anonymous University 1
Anonymous Address and Email 1

Anonymous Author 2
Anonymous Department 2
Anonymous University 2
Anonymous Address and Email 2

Anonymous Author 3
Anonymous Department 3
Anonymous University 3
Anonymous Address and Email 3

Abstract

Background: Sound source localization is a critical skill for animals to navigate their surroundings, detect potential threats, and locate prey [1]. In this study, we leverage Deep Reinforcement Learning (DRL) to address the challenge of localizing sound sources in natural animal environments [2]. Our approach involves training an agent to estimate the direction or location of sound sources using audio signals, while receiving rewards for achieving accurate localization [3]. **Methods:** To accomplish this, we employ a state-of-the-art DRL framework, enabling the agent to autonomously learn and adapt its localization strategies over time. The agent interacts with its environment, continually refining its sound source estimation based on audio cues. Through trial and error, the agent learns to optimize its actions, ultimately leading to more precise and reliable sound source localization. **Expected Results and Conclusion:** Our research has the potential to make significant contributions to the field of bioacoustics and animal behavior studies, as accurate sound source localization is crucial for understanding communication, predator-prey interactions, and overall animal ecology. Additionally, the application of DRL techniques to this problem domain highlights the potential for artificial intelligence to enhance our understanding of the sensory perception and decision-making processes of animals in complex natural environments.

References

[1] Rhinehart, T. A., Chronister, L. M., Devlin, T., & Kitzes, J. (2020). Acoustic localization of terrestrial wildlife: Current practices and future opportunities. Ecology and Evolution, 10(13), 6794-6818.

[2] Jekateryńczuk, G., & Piotrowski, Z. (2023). A Survey of Sound Source Localization and Detection Methods and Their Applications. Sensors, 24(1), 68.

[3] Stowell, D. (2022). Computational bioacoustics with deep learning: a review and roadmap. PeerJ, 10, e13152.

P11

Predicting activity in brain areas associated with emotion processing using multimodal behavioral signals

Lahoucine Kdouri[1], Youssef Hmamouche[1], Amal El Fallah Seghrouchni[1], and Thierry Chaminade[2]

[1]*International Artificial Intelligence Center of Morocco, University Mohammed VI Polytechnique, Rabat, Morocco*
[2]*INT UMR 7289, CNRS, Aix Marseille Université, Marseille, France*

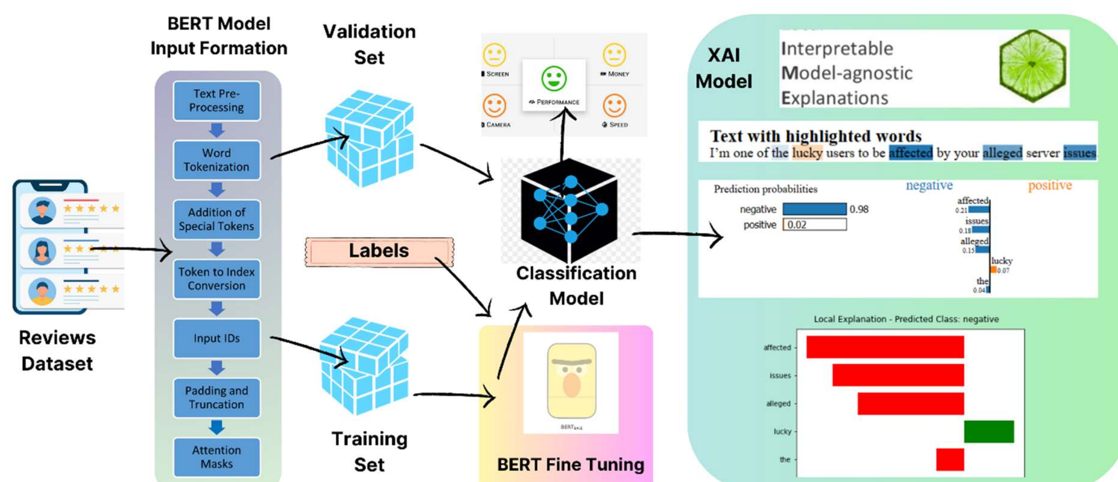
Modern artificial agents are expected to interact with humans and adaptively express emotions relying on multimodal social integration, both in perception and production, in a more realistic manner than present-day systems relying mainly on facial expressions. For such artificial agents, such as humanoid robots, to mimic human behavior including emotional responsiveness, the use of social cognitive neuroscience has become crucial. In this paper, we present a novel approach to understanding emotions, the integration of a multimodal deep learning network to predict brain activity in regions of interest related to emotion processing. The network takes as input two categories of signals recorded synchronously with brain activity during conversations with a human or a robot: behavioural (video and audio) and physiological (blood pulse). This approach allows us to (1) predict brain activity on the bases of multimodal behavioural and physiological signals, and to (2) compute the performance of our system according to the nature, human or robot, of the interlocutor and the localisation of the region of interest. The results obtained demonstrate that, although the proposed architecture is simple in design, it outperforms existing architectures. An ablation study evaluating subsets of the input modalities shows that local brain activity prediction was reduced when one or two modalities were not provided, but suggested that the visual information may be more dispensable than the others to reach the best predictions when participants interacted with another human. It also demonstrated the importance of the physiological data, the blood pulse, in predicting brain response, emphasising the relevance of somatic markers in relation to the central nervous system processing of social emotions.

Abstract

Soonh Taj [1]

[1] *Sukkur IBA University, Sindh Pakistan*

Requirement Elicitation (RE) is one of the most essential phase in software development process as it involves to understand the needs of users and stakeholders and turning them functional and non-functional requirements. One innovative way to gather requirements from users is to refer reviews given by user for a product or service and analyze it using NLP techniques like Aspect-based sentiment analysis (ABSA). ABSA of app reviews allows us to better understand the manner in which users experience and what they are seeking in terms of specific features of a system or product. This analysis of reviews helps the development team to elicit requirements much more effectively. Current literature on ABSA uses BERT based approaches which still lacks robustness in terms of performance and are totally black box in nature and are difficult to interpret. This work proposes an innovative approach to improve traditional BERT based ABSA with more robustness and better interpretability. The proposed approach is based on large pre-trained models like BERT and the integration of LIME an XAI (Explainable Artificial Intelligence). This approach is employed with overall goal to improve the requirements elicitation process by analysing 11323 app reviews from the benchmark dataset AWARE that is tailored specifically for the requirements elicitation process. This work fine-tunes the BERT model for the ABSA task and outperformed baseline and existing studies. This study also exploits LIME for results' explanations that helps to better visualize app review analysis.



P13

Artificial Intelligence acceleration of BNCT dose calculations

Guillermo Marzik[1,2], **M. Eugenia Capoulat**[1,2,3], **Andrés J. Kreiner**[1,2,3] **and Daniel Minsky**[1,2,3]

[1]*Gerencia de Investigación y Aplicaciones, Comisión Nacional de Energía Atómica*
[2]*Consejo Nacional de Actividades Científicas*
[3]*Escuela de Ciencia y Tecnología, Universidad Nacional de San Martín*

In the context of Boron Neutron Capture Therapy (BNCT), the impact of the neutron beam employed for patient irradiation extends beyond the neutron capture reaction with the 10B isotope associated with tumor cells. Various interactions involving neutrons or secondary photons with other elements could potentially affect healthy tissues. Hence, meticulous treatment planning is of paramount importance, ensuring that a given neutron beam configuration optimizes the probability of tumor control while minimizing adverse effects on healthy tissues. This optimization is achieved through the calculation of dose maps over tumor and healthy tissues. Traditionally, these dose maps are estimated using neutron transport simulations based on Monte Carlo methods, which, while accurate, entail a substantial computational cost and demand high computational power and long simulation times for convergence and low statistical errors [1]. This complexity poses challenges for comprehensive studies on optimal treatment configurations for individual patients and may impede the widespread adoption of this therapy in medical centers aiming to treat multiple patients daily.

This study introduces a novel approach leveraging a neural network model designed to expedite the convergence of Monte Carlo simulations. The primary objective is to shift the time-intensive aspect of simulations to the training of the neural network, a process performed only a limited number of times. The proposed model is built upon a variant of the U-Net architecture.

The input data comprises the CT scan of a patient, divided in 4 channels corresponding to the different materials present: air, bone, healthy tissue and tumor, along with 10^4 histories simulations of various dose components (boron dose and dose due to neutrons and photons in different tissues), as well as the error maps for each dose component. Given the relatively low number of histories, these simulations are susceptible to statistical noise due to the algorithm's incomplete convergence. The neural network is trained to predict 10^8 histories simulations (where convergence is achieved) for these dose components, which are then used to compute the final dose map. The L1 norm between the neural network estimations and noise-free simulations obtained through the traditional Monte Carlo approach serves as the cost function for parameter tuning.

The proposed system underwent training using 80 different beam positions across 200 different patients from the Cancer Image Archive [2], resulting in a total of 16000 training instances. For testing, an additional 2000 instances, corresponding to 25 patients, were employed. Importantly, none of the patient data from the testing set was used during the training phase, ensuring the model's generalization to unseen data. Simulations were made using MCNP6.1 [3] and patients were modeled as 24x24x24 voxel arrays, where each voxel has sides of 1 cm.

Across the testing set, 96.9% of the voxels in the 3D dose maps estimated by the proposed system exhibited absolute differences of less than 5% of the maximum dose calculated in dose maps based on 108 histories Monte Carlo simulations. For reference, only 61.4% of the voxels of the 10^4 histories dose maps fulfilled the same requirement. Furthermore, the proposed method achieved convergence with 10^4 less histories than the conventional approach, as inference time of the neural network is neglectable. This highlights its promise as a fast and reliable approach for designing treatment plans within the context of BNCT.

[1] Kumada, H., Takada, K. Treatment planning system and patient positioning for boron neutron capture therapy. Therapeutic Radiology And Oncology, 2 (2018).
[2] Shusharina, N., Bortfeld, T. Glioma Image Segmentation for Radiotherapy: RT targets, barriers to cancer spread, and organs at risk (GLIS-RT) [Data set]. The Cancer Imaging Archive (2021).
[3] Goorley, J.T. MCNP 6.1.1 Beta Release Notes. Los Alamos National Laboratory Tech. Rep. LA-UR-14-24680. Los Alamos, NM, USA (2014).

A Q-LEARNING ALGORITHM FOR DISCRETE-TIME LINEAR-QUADRATIC CONTROL WITH RANDOM PARAMETERS OF UNKNOWN DISTRIBUTION: CONVERGENCE

Kai Du [1] , **Qiingxin Meng**[2] , **and Fu Zhang** [3]

1 Shanghai Center for Mathematical Sciences, Fudan University, Shanghai 200438, China

2 Department of Mathematics, Huzhou University, Zhejiang 313000, China

3College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China

This paper studies an infinite horizon optimal control problem for discrete-time linear systems and quadratic criteria, both with random parameters which are independent and identically distributed with respect to time. A classical approach is to solve an algebraic Riccati equation that involves mathematical expectations and requires certain statistical information of the parameters. In this paper, we propose an iterative algorithm in the spirit of Q-learning for the situation where only one random sample of parameters emerges at each time step. The first theorem proves the equivalence of three properties: the convergence of the learning sequence, the well-posedness of the control problem, and the solvability of the algebraic Riccati equation. The second theorem shows that the adaptive feedback control in terms of the learning sequence stabilizes the system as long as the control problem is well-posed. Numerical examples are presented to illustrate our results.

Facial Expression Recognition using Extended CNN

Antika Saha, Rashed Mustafa[*] and Rezaul Karim
Computer Science and Engineering, University of Chittagong, Bangladesh
(antikasaha15@gmail.com, rashed.m@cu.ac.bd, pinnacle_of_success@yahoo.com)

***Corresponding Author: Rashed Mustafa**

ABSTRACT
Facial expression recognition is an important problem in the field of computer vision. Computer vision is an interdisciplinary scientific field that deals with how computers gain high-level understanding from digital images. The facial expression recognition process formation of three stages they are face detection, feature extraction, and recognizing expression. The idea of expression recognition is helpful for people with physically disabled like hard of hearing and dumb to identify human facial expressions through the help of image processing and computer vision. The system can identify seven several facial expressions: anger, disgust, fear, neutral, happy, sad, and surprise. In the end, the design and implementation of the system are explained. The proposed method is a custom Deep Convolutional Neural Network (DCNN) model with more CNN layers and ten-fold cross-validation which is used to train and test various facial expression images with Google Colab. This paper worked on Kaggle facial expression dataset. The better accuracy of the model acquired is 85.0%, precision 0.83, recall 0.83, and f1-score 0.83 on the testing dataset.

Keywords: Convolutional neural network, Confusion Matrix, Cross-validation, deep learning, Facial expression, Feature extraction.

INTRODUCTION
The face is an essential part of an individual's human body, which plays a vital role in the extraction of an individual's emotional state. The face is responsible for communicating thoughts and ideas as well as emotions. A facial expression is one or more movements or areas of the muscles below the skin of the face. Through facial expressions, people can express their emotions. Identifying facial expression exploration has the capability to give computers to realize human emotions like anger, disgust, fear, happiness, neutral, sadness, and surprise. In nonverbal communication, the expression of faces plays an important role, which defining the interaction between humans and animals.

Deep learning is one kind of machine learning and artificial intelligence (AI) that follow the path of people getting particular categories of wisdom. It's a crucial material of data science that contains data and prognostic modeling. It's highly helpful to the scientists of data tasked with assembling, resolving, and understanding massive quantities of data; deep learning creates this method quicker and lighter. Sometimes deep learning is alluded to as deep neural learning or deep neural networking. Neural networks penetrate distinctly various forms, along with recurrent neural networks, convolutional neural networks, and artificial neural networks, and each has benefits for specific use cases. After 1960 this subject became more universal when a list of popular emotions was determined and several systems were recommended. There are seven basic popular emotions for humans. These are anger, disgust, fear, happiness, neutral, sad, and surprise.

As computers come gradually extensive and their connection with user's changes, they demand new materials to achieve a response from their correlations with those users and reciprocate appropriately. At present, there are various opportunities to derive responses from users like pulsation, articulation, gesture, visual communication, etc. But some of those opportunities are intrusive to users or do not allow sufficient or exact responses in order for a system to be dependable. After reviewing ten years of publications depending on Facial Expression Recognition build up a schematic diagram of the growth of research, where the x-axis denote the year and the y-axis denotes the total number of publications. The graphical representation of the growth of research is in figure 1.

One humble choice that provides a rational amount of facial expressions. The expressions of the face have been deliberated as a great source of knowledge to control the accurate emotions of a character (Ekman, et al, 1997). Even before Charles Darwin guided "studies on how people recognize emotion in faces" (Jabr, 2010), venerable minds like Aristoteles, learned the significance of the expressions of faces (Russell, et al, 1997). Even so, it was before Paul Ekman managed cross-civilization analysis near the world that a set of common emotions like anger, disgust, fear, neutral, happiness, sad, and surprise; were finally approved (Bettadapura 1998, Siegman, 1978).

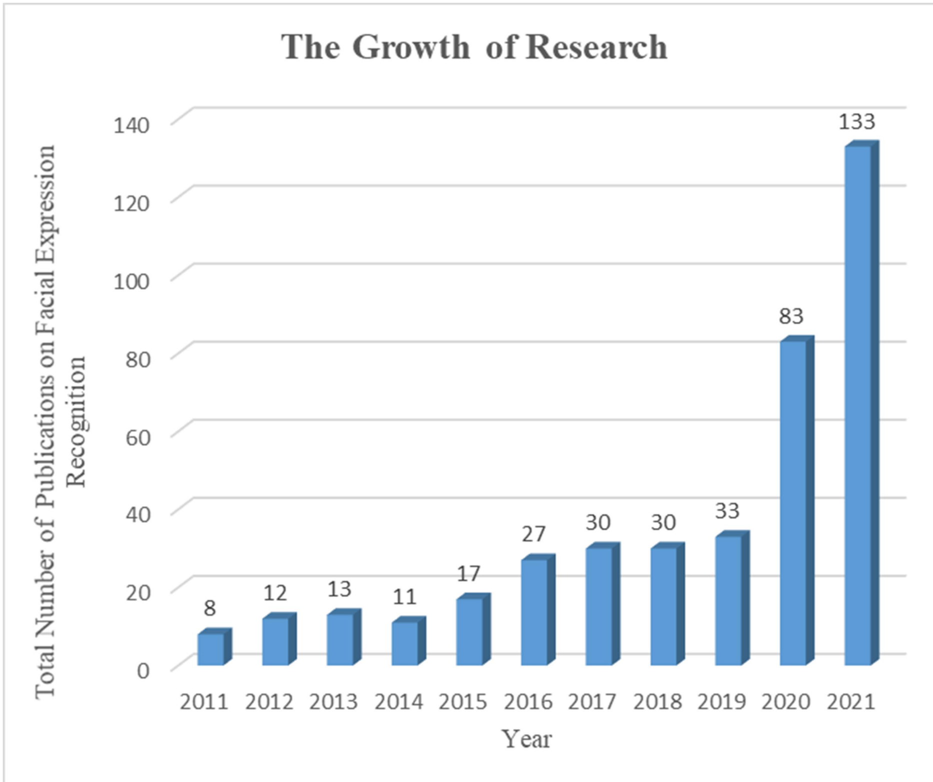


Figure 1: The graphical representation of the growth of research

Facial expressions can demonstrate individual emotions and show personal intentions in social situations. It can carry various emotional states and detect various physiological reactions. At present, facial expression recognition or facial expression based on the computer has motivated significant research work because of its capability to imitate human cipher skills. I have also been motivated to penetrate the advantages of the physically disabled like hard of hearing and dumb. But if the facial expression detection system can recognize their necessity by observing their facial expression then it becomes a lot lighter for them to make the associate facial expression detection system understand their demands. Currently, there is a vast amount of solutions available but still no consensus on what is the best solution when applied to images, and uncertainty.

LITERATURE REVIEW

This experiment has categorized six different facial emotions, which assemble into individual global expressions: anger, disgust, fear, happiness, sadness, and surprise. There is multiple research work that uses many techniques to identify facial expressions. Now is the time to discuss the research work on facial expression recognition and its limitations.

P. Liu, S. Han, Z. Meng, and Y. Tong (Liu. P, et al. 2014) proposed Boosted Deep Belief Network is operative for describing facial expression changes and appointed to form a boosted powerful classifier analytically by using a set of features. This framework can be learned from facial images' extremely complicated features. Using that process they could classify the seven facial expressions and the classification rate is 41%.

Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang (Emad Barsoum, et al, 2016), the deep convolutional neural networks (DCNN) estimate the efficiency of four different patterns to train emotion identification on crowd-sourced labels. Here also handle turbulent levels by using a deep convolutional neural

network (DCNN) for recognizing faces and crowdsourcing used to accumulate place of truth labels. Using that process they could classify the eight facial expressions and the highest accuracy is 85%.
Z. Meng, P. Liu, J. Cai, S. Han, and Y. Tong (Meng. Z, et al, 2017), proposed the identity-aware convolutional neural network (IACNN) process used training, expression, and recognition associated features are together assumed through a deep CNN scheme, which is collected from two duplicate CNN streams and prepared at the same time minimizing the categorization errors when maximizing the expression and recognize similarities. This method has been evaluated on two blatantly obtainable databases CK+ and MMI, the accuracies are 71.29% and 55.41%.
S. K. Lalitha and J. Aishwarya, (Lalitha. S. K, et al, 2021) proposed a raw convolutional neural network (CNN) classifier model. The output layer contains feature maps that reflect the knowledge of the image. Haar cascade classifier is used to identify the faces from the image. By using the proposed method and different types of datasets the average accuracy is 67%.
K. Chang, C. Chen and Y. Hung (Chang. K, et al, 2013) the diagram assesses the distinct predominance state of an expression depending on a single image. An effective descriptor spread covert, which is rendition invariant and can linearize deformation. For the ranking problem, this paper could not work on multiple image-based expression intensity estimations. Results describe that the diagram with dispersed change omits. The accuracy is 71.29%.
According to the proposed work, the facial expression recognition problem has two main perspectives: validity and ability. Ability is counted in terms of time complexity, computational complexity, and space complexity. The targets of this research are to make a proposed method that has high accuracy and extreme computational complications. Here, ready an accurate set of data is a huge challenge. Another challenge is the process of creating an accurate extended CNN method for the emotional recognition of face purposes. Following the complications of the problem is necessary to be composed deep learning; the quantity of input is different.

METHODOLOGY

Deep learning current approaches are used for increasing the processing power of detailed datasets with results that overcome customary methods (Kaiming et al, 2015). The usual pipeline of deep learning for facial expression recognition systems comprises a preprocessing stage of the input image, which is organized by face alignment, data augmentation, and face normalization process. Those processed data are then passed to a neural network, mainly Convolutional Neural Network (CNN), Deep Belief Network (DBN), Recurrent Neural Network (RNN), Generative Adversarial Network (GAN), or Deep Autoencoder (DAE). After the features are turned out by the neural network those images are passed into a classifier that will classify the image. The overview of the facial expression recognition system is illustrated in figure 2. The facial expression recognition system includes the major stages such as image preprocessing, feature extraction, and expression classification.

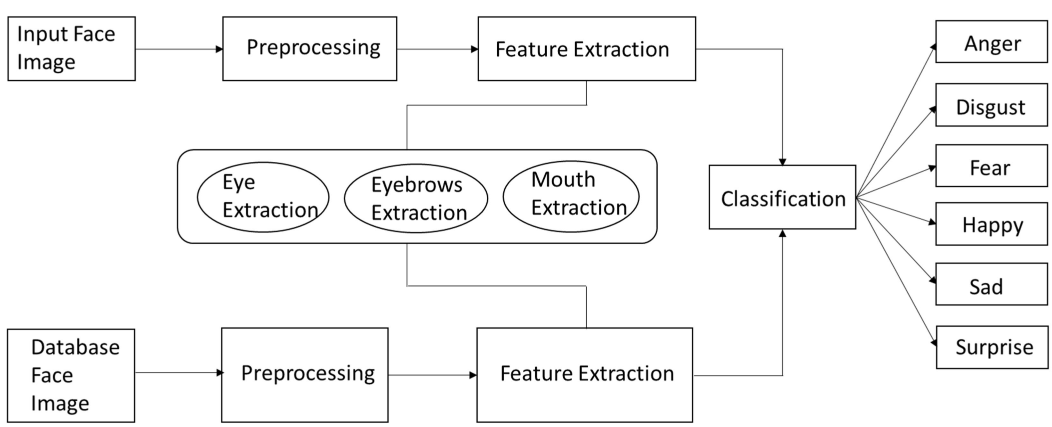


Figure 2: The architecture of the Facial Expression Recognition System

The progress and popularity of computer vision with deep learning are guided by the Convolutional Neural Network (CNN) algorithm. Convolutional Neural Network (CNN) can receive input images, train itself by learning filters to characterize distinct features between images and be capable to distinguish one image from another (Saha. S, 2018). The initial goal of Artificial Intelligence (AI) is to provide a set of algorithms used to resolve problems that humans can solve by instinct and spontaneously but this is a more challenging task for computers (Rosebrock. A, 2018). On the other hand, machine learning deals with the area of the lesson which delivers

computers to know except being detailed programmed (Samuel A. L, 1959).
The proposed Deep Convolutional Neural Network model accepts an input image of 48 × 48 pixels and methods of various Convolution, Max-pooling, and Fully-connected layers giving the ultimate output of the other seven classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. The flow diagram of DCNN is shown in figure 3.

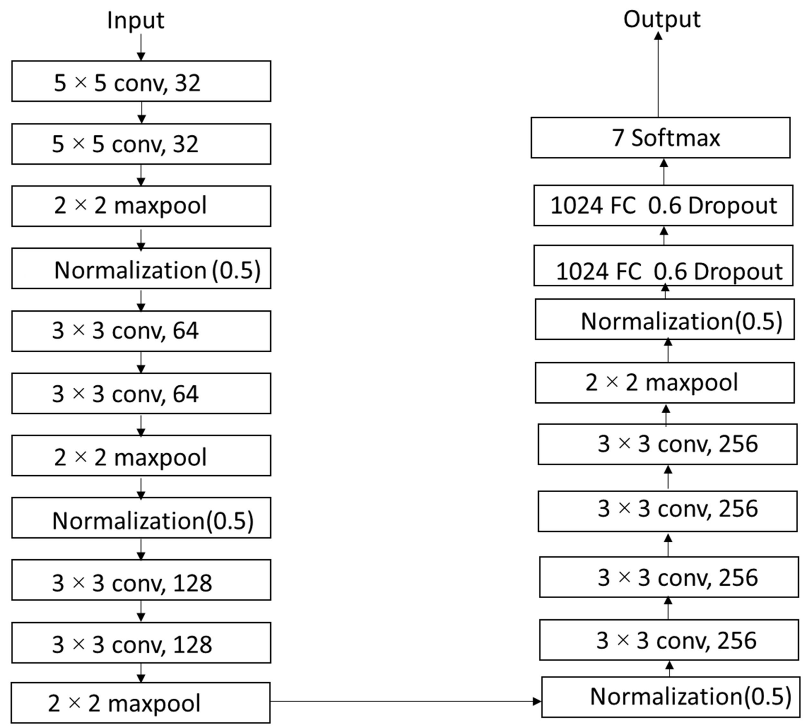


Figure 3: Flow Diagram of Deep Convolutional Neural Network

In this system is a DCNN proposed method with 23 layers for training and testing facial images. It has 10 convolution layers, from those two convolution layers, have a 5 × 5 size of the filter and the other convolution layers have a 3 × 3 size of the filter, and the pooling layers have a size of the pool is 2 × 2. After every convolution layer, an activation layer and batch normalization are included for gaining better accuracy. Every pooling layer is followed by the dropout layer for reducing the overfitting of the model. All the convolution layers are followed by two dense layers, every layer with 1024 hidden units, followed by a 60% dropout layer. The convolution-pooling batch is composed of convolution, activation, batch normalization, pooling, and dropout gradually. Every fully-connected layer is composed of fully connected, activation, batch normalization, pooling, and dropout layers gradually. For classifying the image into the seven individual expressions of faces use a classifier that is softmax.
Figure 4 shows the architecture of the implementation of a deep convolutional neural network. For the initial Convolution layer use the following equation:

$$B[i,j] = \sum_{k_1=0}^{2} \sum_{k_2=0}^{2} A[i + k_1, j + k_2] W_1[k_1, k_2] \qquad \text{------------------}4.1$$

In equation 4.1, W_1 is the filter, and $k_1 = 0:2$, $k_2 = 0:2$. Every convolution layer used the same zero padding at p = 1 and stride s = 1. The same padding outline is used to develop the design of the architecture of networks more proficiently.
This process uses a K fold cross-validation technique to assess vertical models by separating the main dataset into a training dataset and a test set to assess it in occurrence. In this, the main dataset is modified and randomly delivered among k identically sizes sub-datasets. The outputs of this process can be built to give a distinct calculation. All surveys are applied for training and validation and each survey is applied for validation only one time. Here, using 10-fold cross-validation techniques.

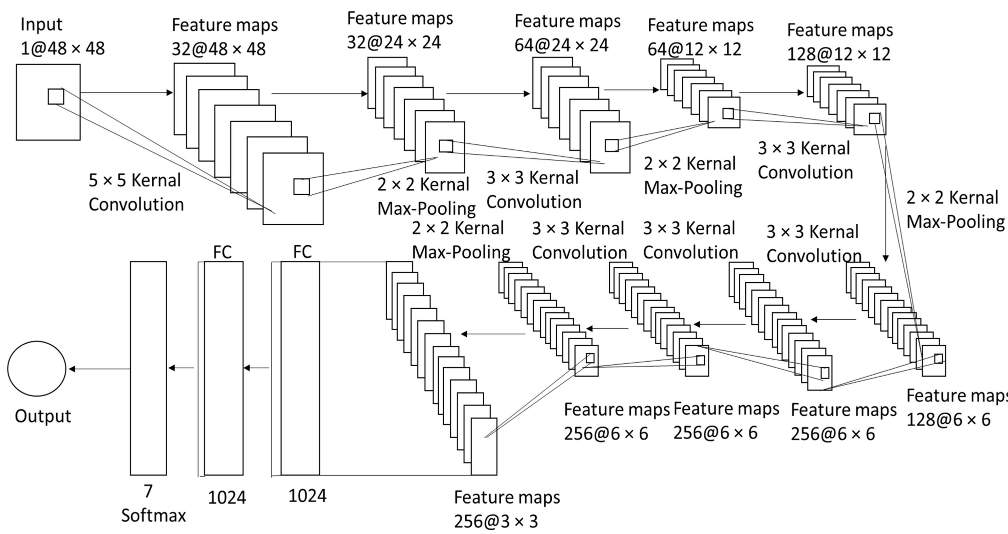


Figure 4: The architecture of the implementation of the Deep Convolutional Neural Network

As a batch normalization using an optimization algorithm that is Adam for finding better results. The results can be arranged in a Confusion Matrix that differs the predicted values from the true values. True positive meaning that data points are correct and false-negative meaning that they are incorrect. Accuracy is the fraction between the number of right predictions and the total number of predictions. It is applied for calculating the number of right predictions.

$$\text{Accuracy} = \frac{TP+T}{\text{total}} \quad \text{---} 4.2$$

Precision is the fraction between the number of right predictions and the total number of positive predictions. It is applied for calculating the ratio of right calculations between the positive ones.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+}} \quad \text{---} 4.3$$

The Recall is the fraction between the number of right predictions and the total number of all predictions. It is applied for calculating the cost of false positives in the model.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+}} \quad \text{---} 4.4$$

F1 score is the harmonic mean of precision and recall. It describes the number of correct predictions and the number of instances.

$$\text{F1} = \frac{2\,.\ \text{Precision}\,.\ \text{recall}}{\text{Precision+reca}} \quad \text{-----------------------------------} 4.5$$

DATASET EXPLORATION

The beginning of the experimental analysis of the proposed model uses different types of expression datasets. The size and color of human faces differ from each other. First, capturing the image using a camera or collecting the image from the internet. Then, the images are cropped in the face region and a face-alignment post-processing phase is directed. At last, the images are counted into similar communities of expressions. The dataset consists of a single .csv file bearing the column's emotion, pixels, and Usage. Every image illustrated in the 48 × 48 vectors in pixels is labeled with an encoded expression. The datasets are divided into training and validation datasets. The inconsistency is performed by using data augmentation processes or growing a cost-sensitive loss function during training.

Compared to other facial expression datasets, the FER dataset has more changes in the images, including facial repression, half faces, low-contrast images, and eyeglasses. The images were gathered using the Google image search API and resized as a result the face is high centered and holds about an equal amount of field in every image. The dataset is made of 28,709 training images, 3,589 validation images, and 3,589 test images with seven several emotions such as happiness, anger, fear, disgust, surprise, neutral, and sadness (Ian J, et al, 2013). The seven different expressions sample images from the FER dataset are shown in figure 5.

Figure 5: Seven different samples images from the FER2013 dataset

RESULTS

From the FER dataset, the expression of faces is available so the total number of samples of faces is 35887 and this dataset is used to appraise the proposed method. Compare to other evaluations, could not find any work which computes the cross-validation with the FER facial expression dataset. So now computes ten-fold cross-validation with the proposed model using StratifiedKFold tool on this FER dataset. The proposed model is built using the Sequential Keras backbone, which gives grouping a linear stack of layers into a Keras model. The proposed network is trained up to 200 epochs with 10-fold cross-validation where the datasets are divided into ten folds using StraightKFold machine learning tool by placing the exchanging parameter, optimizing the cross-entropy loss using Adam optimizer. Adam optimization is a stochastic gradient descent method that depends on the adaptive estimation of first-order and second-order moments. Using the Adam algorithm the method is computationally efficient, has little memory requirement, is invariant to diagonal rescaling of gradient, and is well suited for problems that are large in terms of data.

VGG19 (Simonyan, K, 2015) was one of the first architectures that used small kernel sizes and increased the depth of the network with 19 layers, which led to a reduction in the number of parameters. The training results illustrated in figure 6 are the accuracy and loss learning curves. By using the ten-fold cross-validation on this VGG19 model the validation accuracy is 62.43%.

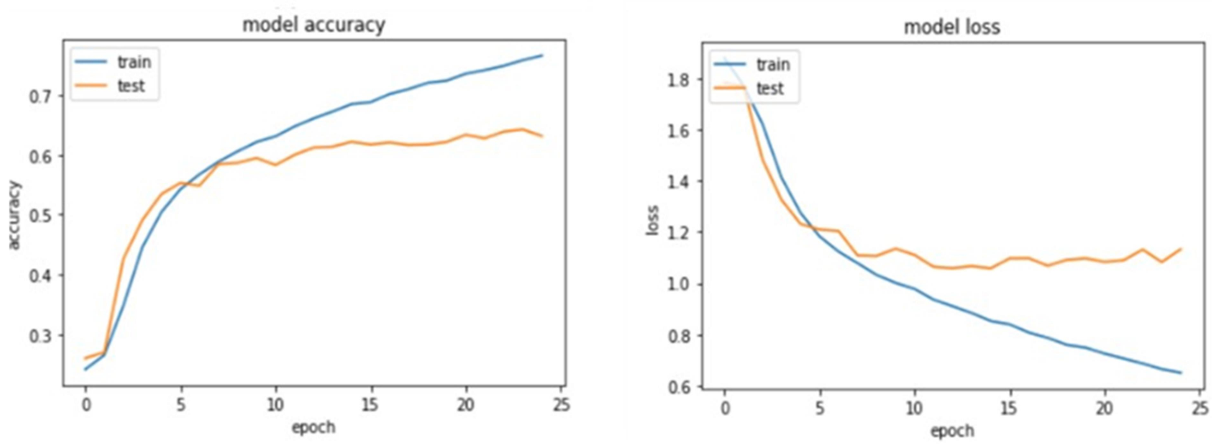
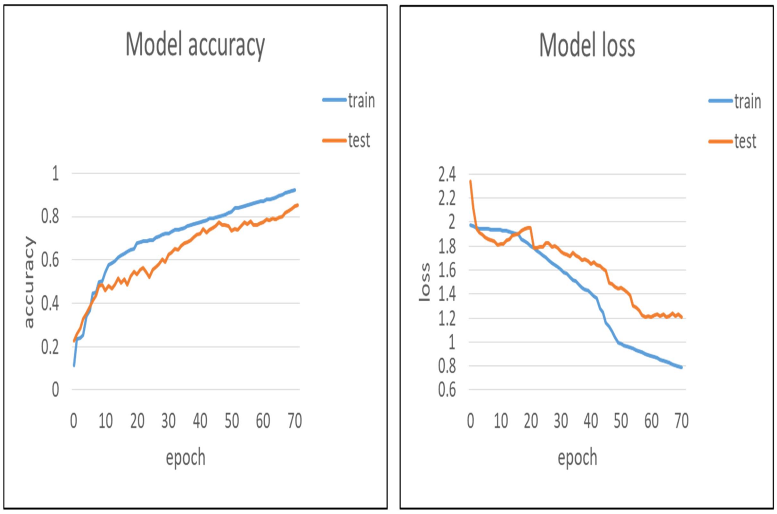


Figure 6: The VGG19 method accuracy, loss on the FER2013 dataset

By using the extended CNN proposed method and ten-fold cross-validation the validation accuracy is 85%. The facial expression recognition (FER) dataset given its size and all faces are aligned in the images. The accuracy and loss learning curves and the confusion matrix of the FER validation set are illustrated in figure 7.



Actual \ Predicted	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	80.0	3.0	5.0	3.0	7.0	2.0	0.0
Disgust	8.0	66.0	9.0	7.0	0.0	8.0	2.0
Fear	3.0	0.0	92.0	5.0	0.0	0.0	0.0
Happy	0.0	0.0	2.0	87.0	5.0	4.0	2.0
Neutral	4.0	0.0	5.0	3.0	82.0	4.0	2.0
Sad	4.0	2.5	1.5	0.0	5.0	85.0	2.0
Surprise	2.0	0.0	1.0	2.0	4.0	0.0	91.0

Figure 7: The proposed method accuracy, loss, and confusion matrix on the FER2013 dataset

According to the extended CNN model calculated the accuracy with Adam optimizers those easy to find by the result of precision, recall and f1-score of the dataset FER2013 validation set using ten-fold cross-validation. The classification report containing precision, recall, and F1-score for each class is shown in table 1.

Table 1: Precision, Recall, and F1-score of the FER2013 validation set

	precision(%)	recall(%)	f1-score(%)
Angry	72	80	75
Disgust	79	66	72
Fear	88	92	90
Happy	84	87	85
Neutral	75	82	78
Sad	87	85	86
Surprise	94	91	92
Macro average	83	83	83
Accuracy			**85**

DISCUSSION & CONCLUSION

After that, this experiment computes by splitting the dataset using 10-fold cross-validation. For getting better accuracy split the datasets into 10% training and 90% testing, 20% training and 80% testing, 30% training and 70% testing, 40% training and 60% testing, 50% training and 50% testing, and so on. The best validation accuracy with ten-fold cross-validation is 85% compared with the VGG19 model which gives the 62.43% accuracy value for recognizing the expression. Finally, the proposed extended CNN model with a ten-fold cross-validation process can classify facial expressions of humans i.e. happiness, anger, fear, disgust, neutral, sad, and surprise. Also, using the confusion matrix the model can be evaluated precision, recall, and f1- score. Using those different types of methods found the better accuracy of the model acquired is 85%, precision 0.83, recall 0.83, f1-score 0.83.

In this thesis, the target is to sketch an extension of the convolutional neural network to identify expression recognition of faces which helps the physically disabled like those hard of hearing and dumb. The direction of identifying the expression of faces is according to a sketch and improvement of a Convolutional Neural Network (CNN) able to forecast human emotions of faces. The CNN model form of ten convolutional layers arranged on the highest in all with the number of kernels redoubling in every obstacle. The extended CNN model learned on the FER2013 dataset, which makes with descriptions holding images of various lighting strikingly situations.

REFERENCES

Bettadapura, Vinay. (1998) "Face Expression Recognition and Analysis: The State Of The Art".

Chang. K, Chen. C and Hung. Y (2013), "Intensity Rank Estimation of Facial Expressions Based on a Single Image," *IEEE International Conference on Systems, Man, and Cybernetics*.

Ekman. P and Rosenberg. E (1997). "What the face reveals". New York: Oxford University Press.

Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang (2016). "Training deep networks for facial expression recognition with crowd-sourced label distribution". In Proceedings of the 18th ACM International Conference on Multimodal Interaction.

Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al (2013). "Challenges in representation learning: A report on three machine learning contests".

Jabr, Ferris. (2010) "The Evolution of Emotion: Charles Darwin's Little-Known Psychology Experiment".

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Deep residual learning for image recognition".

Lalitha. S. K, Aishwarya. Shivakumar. Srilekha. J. N. T and Kartheek. G. C. R (2021), "A Deep Learning Model for Face Expression Detection," *International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*.

Liu. P, Han. S, Meng. Z, and Tong. Y (2014), "Facial Expression Recognition via a Boosted Deep Belief Network," *IEEE Conference on Computer Vision and Pattern Recognition*.

Meng. Z, Liu. P, Cai. J, Han. S, and Tong. Y (2017), "Identity-Aware Convolutional Neural Network for Facial Expression Recognition," 12th IEEE International Conference on Automatic Face & Gesture Recognition.

Rosebrock. A (2018), "Deep Learning for Computer Vision with Python", 1.3.0. PyImageSearch.com.

Russell, James A, and José Miguel Fernández Dols (1997). "The Psychology of Facial Expression". Cambridge: Cambridge University Press.

Saha. S (2018), “A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way,” Towards Data Science.

Samuel A. L (1959), “Some studies in machine learning using the game of Checkers,” IBM J.

Siegman, Aron Wolfe, and Stanley Feldstein (1978). "Nonverbal Behavior and Communication. Facial Expression". Hillsdale, N.J.: L. Erlbaum Associates.

Simonyan, K. and Zisserman, A. (2015). "Very deep convolutional networks for large-scale image recognition".

P16

Bagging & Boosting ensembles based deep hybrid architecture for histopathological breast cancer classification

Fatima-Zahrae Nakach [1], **Hasnae Zerouaoui** [1], **and Ali Idri** [1,2]

[1]*Mohammed VI Polytechnic University (UM6P), Ben Guerir, Morocco*
[2] *ENSIAS, Mohammed V University, Rabat, Morocco*

Breast cancer is the most commonly diagnosed cancer type and one of the top leading causes of death in women worldwide [1]. Research indicates that most experienced pathologists can diagnose cancer with an accuracy of 79% while ML techniques may reach an accuracy of 91% [2]. However, using one single technique does not always guarantee a high level of accuracy under all circumstances since most of the ML techniques suffer from the problem of high variance or/and high bias, which motivates the use of ensemble learning techniques [3][4].
The main objective is to look into the effect of the type and number of base learners on the predictive capability of the bagging & boosting ensembles and to examine the impact of various feature extraction deep learning models on the performance of the ensembles.
The empirical evaluations used: four classification performance criteria (accuracy, recall, precision and F1-score), the 5-fold cross-validation, Scott Knott statistical test to select the best cluster of the outperforming models, and Borda Count voting system to rank the best-performing ones.
The results demonstrated that combining CNNs for feature extraction with bagging and boosting ensembles is an effective and a promising approach for the automatic classification of histopathological breast cancer images. The ensemble methods consistently perform well over all the MFs in terms of their accuracy, sensitivity, recall and F1-score: The best bagging ensemble achieved a mean accuracy value of 93.98%, and was constructed using 3 base learners, 200× as MF, MLP as a classifier, and DenseNet201 as a feature extractor. The best boosting ensemble achieved an accuracy value of 92.52% and it was constructed using XGBoost with 200 trees and Inception V3 as feature extractor. Overall, bagging ensembles tend to outperform boosting ensembles for each feature extractor and MF.

[1] Sung H, Ferlay J, Siegel RL, Laversanne M, Soerjomataram I, Jemal A, Bray F. Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. CA Cancer J Clin. 2021;71:209–49.
[2] Hamed G, Marey MAE-R, Amin SE-S, Tolba MF. Deep learning in breast cancer detection and classification. In: Hassanien A-E, Azar AT, Gaber T, Oliva D, Tolba FM, editors. Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020). Springer International Publishing, Cham; 2020. p. 322–33. [3] Nakach, FZ., Zerouaoui, H. Idri, A. Hybrid deep boosting ensembles for histopathological breast cancer classification. Health Technol. 12, 1043–1060 (2022). https://doi.org/10.1007/s12553-022-00709-z
[4] Nakach, F. Z., Idri, A., Zerouaoui, H. (2023). Deep Hybrid Bagging Ensembles for Classifying Histopathological Breast Cancer Images. In ICAART (2) (pp. 289-300).

P17

Although different architectures of quantum perceptrons have been recently put forward, the capabilities of such quantum devices versus their classical counterparts remain debated. Here, we consider random patterns and targets independently distributed with biased probabilities and investigate the storage capacity of a continuous quantum perceptron model that admits a classical limit, thus facilitating the comparison of performances. Such a more general context extends a previous study of the quantum storage capacity where using statistical mechanics techniques in the limit of a large number of inputs, it was proved that no quantum advantages are to be expected concerning the storage properties. This outcome is due to the fuzziness inevitably introduced by the intrinsic stochasticity of quantum devices. We strengthen such an indication by showing that the possibility of indefinitely enhancing the storage capacity for highly correlated patterns, as it occurs in a classical setting, is instead prevented at the quantum level.

P18

Isolated pulsar population synthesis with simulation-based inference

Celsa Pardo Araujo[1], **Vanessa Graber**[1,2], **Michele Ronchi**[2], **and Nanda Rea**

[1] *Institute of Space Sciences (CSIC-ICE), Campus UAB, Carrer de Can Magrans s/n, 08193, Barcelona, Spain*

[2] *Institut d'Estudis Espacials de Catalunya (IEEC), Carrer Gran Capita 2–4, 08034 Barcelona, Spain*

Many challenges in astrophysics involve the task of constraining free parameters of a physical model to match the observed reality. Often, these models are highly complex, making Bayesian inference through traditional approaches impractical due to intractable likelihoods. To overcome this issue we can use so-called simulation-based (or likelihood-free) inference (for a recent review see [1]). This approach is particularly powerful when combined with deep learning, wherein a neural network learns to map from the simulated data onto the posterior distribution of the underlying parameters ([2]). In this presentation, I will explore the possibility of using one of these techniques within the context of neutron star population synthesis. Neutron stars are highly compact objects, born in the core-collapse supernovae of massive stars. Neutron stars are characterized by their high magnetic fields and fast spinning. These extreme properties make them perfect laboratories to study fundamental physics of ultra-dense and strongly magnetized matter. However, although about a billion neutron stars are expected to exist in the Milky Way, observational constraints limit us to only observing a few thousand. Pulsar population synthesis bridges this gap by simulating the entire population and comparing it to the observed sample to constrain neutron-star physics (e.g. [3], [4] and [5]). I will demonstrate how we can constrain the physical properties of isolated Galactic neutron stars by combining population synthesis with simulation-based inference. For this purpose, we implement a population-synthesis framework able to simulate the stars' dynamical and magneto-rotational evolution as well as their radio emission and incorporate selection biases of typical radio surveys. We then generate a dataset of mock pulsar populations to train and validate a mixture-density neural network. In particular, I will present recent results that demonstrate how we successfully train neural networks on simulated data to infer the initial period distributions and magnetic-field properties of neutron stars without assuming a simplified likelihood.

[1] Cranmer, K., Brehmer, J., and Louppe, G. 2020, Proceedings of the National Academy of Science, **117**, 30055 (2020).
[2] Papamakarios, George, and Iain Murray. Advances in neural information processing systems **29** (2016).
[3] Faucher-Giguere, C.-A., and Kaspi, V. M., ApJ, **643**,332 (2006).
[4] Gullón, M., Pons, J. A., Miralles, J. A., et al. , MNRAS, **454**, 615 (2015).
[5] Ciéslar, M., Bulik, T., Os lowski, S. 2020, MNRAS, **492**, 4043 (2020).

P19

Gaussian processes for time-series data analysis: case of study for a large dataset

Isac Pasianotto[1], **Ruggero Lot**[2], **and Niccolò Tosato**[1,2]

[1] *University of Trieste*
[2] *AREA Science Park*

Gaussian processes (GPs)[2] are powerful function approximators employed in regression tasks. Compared to other methods, they benefit from analytical tractability, a non-parametric nature, and the possibility of quantifying uncertainty. Still, all these advantages come at a cost. Given a dataset with N entries, fitting the parameters of the kernel function and performing the prediction requires a computational cost of $O(N^3)$ due to the GP's covariance matrix.
The canonical approaches to tackle this limitation are based on approximations [1], which induce artificial sparsity in the covariance matrix, making it easier to manipulate and cheaper to store in memory.

This preliminary work attempts to contain this cost by reimplementing the approach proposed in [3] using the distributed framework DASK [4]. Assuming that most points in a real dataset are naturally uncorrelated, such an approach leverages well-defined kernel families that can discover (hence not induce) the sparsity in the matrix, making this an exact approach at the price of adding hyperparameters to the model. Our novel implementation improves over the previous one, considering the combination of the proposed "*Sparsity discovering*" kernel with kernels that are more suitable for time-series data (e.g., periodic kernel). In particular, the poster presents the code implementation and the extension of the time-series data analysis performed in [3] on the average temperatures in the US for the past 30 years to a larger dataset of records about the worldwide temperatures over 50 years.

We conclude by presenting our initial results obtained by choosing different hyperparameters and discussing how the trade-off between the total number of hyperparameters introduced by this approach and the accuracy of predictions behave in the case of estimating periodic function.

[1] Matthew J Heaton et al., J Agricultural, Biological, and Environmental Statistics. 398–425, "A Case Study Competition Among Methods for Analyzing Large Spatial Data" (2019).

[2] MacKay, David J. C., Information Theory, Inference and Learning Algorithms. Copy- right Cambridge University Press, 2003. ISBN: 97805216429896.

[3] Marcus M. Noack et al., "Exact Gaussian Processes for Massive Datasets via Non-Stationary Sparsity-Discovering Kernels". arXiv:2205.09070 [stat.ML], (2022)

[4] Matthew Rocklin. "Dask: Parallel Computation with Blocked algorithms and Task Scheduling". 14th "PYTHON IN SCIENCE" conf., (Scipy 2015)

P20

HMM and extesions to classify time series data: an application in animal movement

Sofia Ruiz Suarez [1,3] ,**Vianey Leos-Barajas** [1] , **and Juan Manuel Morales** [2,3]

[1]*Department of Statistical Sciences, University of Toronto*
[2] *School of Biodiversity, One Health and Veterinary Medicine, University of Glasgow*
[3] *Grupo de Ecología Cuantitativa. Instituto de Investigaciones en Biodiversidad y Medioambiente, CONICET*

Hidden Markov models (HMMs) and their extensions have proven to be powerful tools for classification of observations that stem from systems with temporal dependence as they take into account that observations close in time are likely generated from the same state (i.e., class). When information on the classes of the observations is available in advanced, supervised methods can be applied. In this paper, we provide details for the implementation of four models for classification in a supervised learning context: HMMs, hidden semi-Markov models (HSMMs), autoregressive-HMMs, and autoregressive-HSMMs. Using simulations, we study the classification performance under various degrees of model misspecification to characterize when it would be important to extend a basic HMM to an HSMM. As an application of these techniques we use the models to classify accelerometer data from Merino sheep to distinguish between four different behaviors of interest. In particular in the field of movement ecology, collection of fine-scale animal movement data over time to identify behavioral states has become ubiquitous, necessitating models that can account for the dependence structure in the data. We demonstrate that when the aim is to conduct classification, various degrees of model misspecification of the proposed model may not impede good classification performance unless there is high overlap between the state-dependent distributions, that is, unless the observation distributions of the different states are difficult to differentiate.

Artificial Intelligence Review
https://doi.org/10.1007/s10462-023-10513-4

DeepThink IoT: The Strength of Deep Learning in Internet of Things

Divyansh Thakur[1] · Jaspal Kaur Saini[1] · Srikant Srinivasan[2]

Abstract

The integration of Deep Learning (DL) and the Internet of Things (IoT) has revolutionized technology in the twenty-first century, enabling humans and machines to perform tasks more efficiently. The combination of DL and the IoT has resulted in significant advancements in technology by improving the efficiency, security, and user experience of IoT devices and systems. The integration of DL and IoT offers several benefits, including improved data processing and analysis capabilities, the ability for IoT devices to learn from data and adapt to changing conditions, and the early detection of system malfunctions and potential security breaches. This survey paper provides a comprehensive overview of the impact of DL on IoT, including an analysis of sensor data to detect patterns and make predictions, and the implications for various industries such as healthcare, manufacturing, agriculture, and smart cities. The survey paper covers topics such as DL models, frameworks, IoT connectivity terminologies, IoT components, IoT service-oriented architecture, IoT applications, the role of DL in IoT, and challenges faced by DL in IoT. The study also presents quantitative achievements that highlight the potential impact of IoT and DL in environmental contexts such as precision farming and energy consumption. Overall, the survey paper provides an excellent resource for researchers interested in exploring the potential of IoT and DL in their field.

Keywords Deep learning · Deep learning models · Strength of deep learning · Activation functions · IoT · IoT applications · IoT challenges in deep learning

✉ Divyansh Thakur
divyanshthakur8@gmail.com

Jaspal Kaur Saini
jaspalkaursaini@iiitu.ac.in

Srikant Srinivasan
srikant.srinivasan@plaksha.edu.in

1 School of Computing, Indian Institute of Information Technology Una, Una, India

2 Plaksha University, Mohali, India

Published online: 04 June 2023

1 Introduction

For centuries, humans have dreamed of creating machines that could stimulate the human brain. Artificial Intelligence (AI) (Russell 2010) is the key solution that mimics the working of the human brain and incorporates all those functionalities of the human brain into machines to make machines think and respond like humans. AI is a field of computer science that focuses on developing machines or systems capable of carrying out activities that would typically necessitate human intellect, including speech recognition, decision-making, language translation, and visual perception. AI has the capacity to bring about a radical transformation in numerous industries and can alter our way of life and work. However, it also presents moral and social difficulties like privacy concerns and job displacement.

Machine learning (ML) is a subset of AI that involves creating algorithms and statistical models that allow machines to learn from data and make decisions or predictions without being explicitly programmed (Mitchell 2007). These algorithms can be trained on a variety of data sources, such as images, text, or sensor readings, to identify patterns and relationships within the data. The main goal of ML is to create models that can make predictions or decisions that generalize well to new, unseen data. There are three primary types of ML algorithms: supervised learning (Cunningham et al. 2008), unsupervised learning (Celebi and Aydin 2016), and reinforcement learning (Sutton and Barto 2018). In supervised learning, the machine is provided with a labelled dataset and can learn to predict the output for a new input based on the examples in the dataset. For example, a supervised learning algorithm could be trained to classify images of animals as either dogs or cats based on labelled examples of each. In unsupervised learning, the machine is given an unlabelled dataset and must find patterns or structures in the data on its own. This type of learning is often used for tasks such as clustering or anomaly detection, where the goal is to identify groups or anomalies in the data. Lastly, reinforcement learning involves the ML by interacting with an environment and receiving feedback in the form of rewards or penalties. These three types of ML algorithms play a significant role in enabling machines to learn, adapt, and improve their decision-making capabilities. With the increasing availability of data and computing resources, ML has become a critical component in many industries, including healthcare, finance, and transportation, among others. By allowing machines to learn from data, ML has the potential to transform how we interact with technology and solve complex problems.

ML algorithms have a diverse range of applications, including image and speech recognition, natural language processing (NLP), and detection of anomalies. The effectiveness of these algorithms is traditionally determined by the quality of the input data representation. A flawed representation can result in subpar performance compared to a superior representation. Additionally, ML may fail when dealing with vast amounts of data and may necessitate human involvement to improve and train. The growth of big data, the expansion of computational capacity, and the advancements in algorithms have enabled the creation of cutting-edge ML models, such as Deep Learning (DL) (Goodfellow et al. 2016), that can tackle complex tasks.

To manage large quantities of data and minimize human involvement, DL is crucial. The roots of DL can be traced back to Aristotle's associationism in 300 B.C., one of the earliest endeavours to comprehend the human brain. In 1943, the McCulloch–Pitts (MCP) model (Hayman 1999) marked the beginning of the modern era of DL and served as the prototype for artificial neural networks (ANNs). Subsequently, the Hebbian theory (Brown and Milner 2003), originally applied to biological systems in nature, was introduced. In 1958, the perceptron (Minsky and Papert 1988), the first electronic device in the field of cognition systems, was developed.

DL employs deep neural networks, also referred to as multi-layered neural networks. These networks are modelled after the human brain's information processing and are particularly effective for tasks Examples of these include recognizing images and speech, processing natural language, and making decisions. DL models can be trained using extensive data and learn features and representations automatically from the data. The primary benefit of DL models is that they are capable of making predictions or decisions based on data without the need for human input or manual feature engineering. DL models are utilized predominantly in computer vision, NLP, speech recognition, recommendation systems and prediction. They can be trained through methods such as supervised, unsupervised and reinforcement learning, allowing them to execute tasks such as image/speech recognition, NLP, and decision-making.

DL models form the foundation of various cutting-edge methods and are widely applied across various industries, including finance, healthcare, transportation, retail, and more. The field is rapidly evolving, and researchers are constantly exploring new ways to improve the effectiveness of these models through new architectures and techniques. DL algorithms empower computers to gain knowledge through experience and perceive the world through a hierarchy of concepts, each described in terms of its relationship to basic logic (Schmidhuber 2015). These methods have brought about dramatic improvements in the domains of speech recognition (Deng 2016) visual object recognition (Cichy et al. 2016), object detection (Zhao et al. 2019), and many other areas such as genomics and drug discovery.

The Internet of Things (IoT) (Wortmann and Flüchter 2015), is coming across as the next wave of revolution in the era of computing sensors or networks (Thakur et al. 2019). The ability to retrieve data from sensors and actuators embedded in our local environment. IoT technology allows devices to communicate with each other and with a central system, enabling the automation of various processes and the collection of data for analysis. This can lead to increased efficiency, cost savings, and improved decision-making in a wide range of industries such as manufacturing, transportation, healthcare, and home automation. IoT devices can be connected through wired or wireless networks, including a wide range of devices such as smart thermostats, security cameras, industrial machines, and wearable devices.

The IoT relies on several key technologies to function effectively. One of the most important is sensor technology (Gharibzahedi et al. 2022), which allows IoT devices to collect data from the physical world. Sensors are capable of measuring a wide range of physical properties, including temperature, humidity, pressure, and motion. Another critical technology for IoT is wireless communication (Khanh et al. 2022), as devices must be able to communicate with one another and other systems. Bluetooth, Zigbee, and LoRaWAN are common wireless technologies used in IoT. Cloud computing (Phasinam et al. 2022) is also essential, as IoT devices generate vast amounts of data that require storage and analysis. Cloud computing provides a cost-effective solution to manage this data and offers scalability. Furthermore, big data technologies (Zhou 2022) and analytics are used to process and analyze this data to extract insights and make decisions. As the volume of data produced by IoT devices grows, edge computing (Kong et al. 2022) has become crucial technology, enabling data processing to take place closer to the source and reducing the amount of data that needs to be transmitted to the cloud.

The IoT is rapidly evolving and is expected to significantly impact many aspects of our lives, from how we live and work to how we travel and consume goods and services.

1.1 Duo of DL and IoT

The fusion of IoT and DL has led to a transformative shift in our interaction with machines and devices. DL, which is a subdivision of ML, involves instructing algorithms to identify patterns in data. Meanwhile, IoT pertains to the interconnected network of physical objects, structures, vehicles, and other devices fitted with sensors, software, and connectivity for the purpose of exchanging data. The amalgamation of these two technologies has facilitated the development of intelligent machines and devices that can function independently and process vast amounts of information in real-time. The coupling of DL and IoT has the potential to revolutionize various industries, ranging from healthcare to manufacturing, by enabling efficient decision-making, automation, and predictive maintenance.

DL integration into IoT enables devices to make decisions and predictions using data, through neural networks that mimic human brain processing. This integration leads to more intelligent devices capable of tasks such as object recognition, speech recognition and anomaly detection. It also results in more efficient and accurate data processing, improving overall IoT system performance.

DL is a critical technology in the IoT, and it has several key applications. One of the most important is object recognition (Salari et al. 2022), which uses DL algorithms to enable IoT devices to recognize and identify objects in images and videos. This can be used in applications such as security, surveillance, and robotics. DL algorithms can also be used for anomaly detection (Xia et al. 2022a), which can help identify potential problems with equipment or systems by analyzing sensor data. Another application is predictive maintenance (Achouch et al. 2022), where DL algorithms analyze sensor data to predict when equipment or systems are likely to fail, allowing for proactive maintenance to be performed. DL algorithms can also be used for speech recognition (Li 2022), allowing IoT devices to understand and respond to spoken commands in applications such as voice assistants and home automation. Additionally, DL enables IoT devices to process and understand natural language input (Khurana et al. 2023), which can be used in applications such as language-based interaction and personal assistance. Moreover, DL can be used to create recommender systems (Wu et al. 2022) that provide customized suggestions to users based on their preferences, past actions, and habits.

Overall, the integration of DL in IoT can improve the performance and functionality of IoT systems by allowing devices to make decisions and predictions based on data and also make them more intelligent and efficient. Moreover, DL algorithms can be used to analyze sensor data from IoT devices to make predictions or detect patterns, which can be used to improve the efficiency of IoT systems and automate decision-making. Additionally, DL can be used to improve the security of IoT systems by detecting and preventing cyberattacks. Some examples of DL in IoT include image recognition on cameras, voice recognition for smart speakers, and anomaly detection in manufacturing.

This paper is aimed at IoT researchers and developers who aim to create analytics, AI systems, and learning solutions using DL techniques for their IoT infrastructure. The paper makes the following contributions:

1. Identifies and discusses the importance of different DL frameworks required for establishing an IoT network, highlighting the strength of DL in handling large, complex datasets.

2. Provides an in-depth explanation of the functioning of various DL models that can be used in an IoT network, showcasing the strength of DL in extracting patterns and features from raw data.
3. Presents a comprehensive overview of the different applications of IoT where DL can be integrated, including real-world examples, demonstrating the strength of DL in enabling predictive and prescriptive analytics for IoT systems.
4. Offers a detailed explanation of the components, layers, and terminology related to connectivity and communication technologies required for setting up an IoT network, emphasizing the strength of DL in enabling real-time decision-making and automation.
5. Examines the challenges and difficulties in integrating DL and IoT applications, including issues related to data management, security, and privacy, highlighting the strength of DL in enabling effective and efficient data processing and analysis for IoT systems.

2 Paper layout

The paper is organized in the following manner: Section 2 outlines the survey procedure. Section 3 discusses biological neurons, ANNs, perceptrons, and the functioning of neural networks. Section 4 discusses DL architectures, activation functions, and DL frameworks. Section 5 provides a comprehensive overview of IoT characteristics, connectivity terms, components, service-oriented architecture, layers, communication technologies, and applications. Section 6 examines the application of DL in IoT devices and applications. Section 7 highlights the open challenges in IoT for DL. Section 8 concludes the paper with a summary. Figure 1 provides a visual representation of the paper's overall structure, while Table 1 presents the same information in a tabular format.

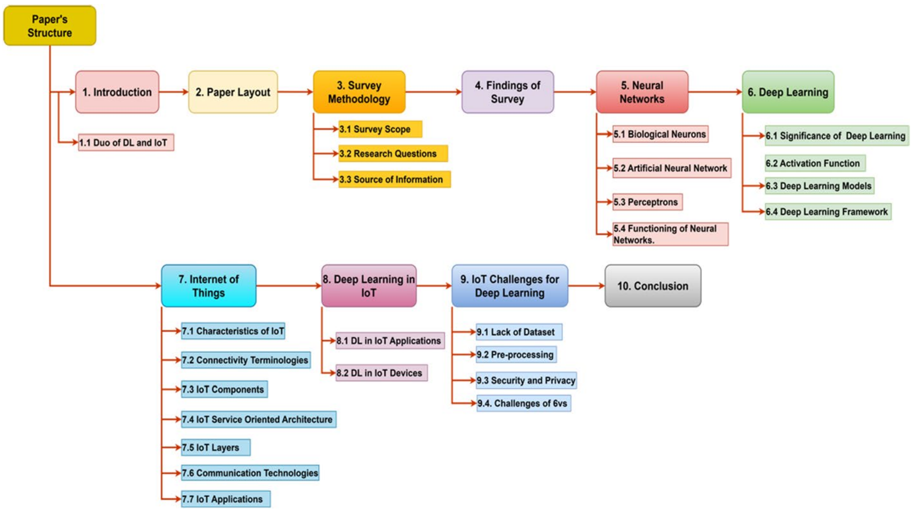


Fig. 1 Visual representation of the layout of a research paper, including its sections, headings, and subheadings

Table 1 Tabular layout of paper

Section no.	Section	Sub section no.	Sub section
1.	Introduction	1.1	Duo of DL and IoT
2.	Paper layout	–	–
3.	Research methodology	3.1	Survey scope
		3.2	Research questions
		3.3	Source of information
4.	Findings of the survey	–	–
5.	Neural network	5.1	Biological neurons
		5.2	ANNs
		5.3	Perceptrons
		5.4	Functioning of neural network
6.	DL	6.1	Significance of DL
		6.2	Activation functions
		6.3	DL models
		6.4	DL framework
7.	IoT	7.1	Characteristics of IoT
		7.2	Connectivity terminologies
		7.3	IoT components
		7.4	IoT service-oriented architecture
		7.5	IoT layers
		7.6	Communication technologies
		7.7	IoT applications
8.	Usage of DL in IoT applications and IoT devices	8.1	DL for IoT applications
		8.2	DL in IoT devices
9.	Challenges of IoT for DL	–	–
10.	Conclusion	–	–

3 Research methodology

3.1 Survey scope

In this survey, various DL models and deep neural network architectures are presented, which are suitable for use in IoT applications and devices have been reviewed. The main projection of this survey is to focus on the combined capabilities of two evolving technologies one in the field of AI that is DL and the second in the field of communication that is IoT. With the help of this survey, various researchers can attain knowledge about the role of DL in various fields of IoT.

3.2 Research questions

The motive of this in-depth survey is to attain the answers to the following research questions (RQ):

- *RQ1* What are the various frameworks for DL?
- *RQ2* What are the various DL models which can be used in IoT data?
- *RQ3* What are the IoT challenges for DL?
- *RQ4* What will be the future direction for the usage of DL in IoT?
- *RQ5* What are the computational limits of DL?
- *RQ6* What are the strengths of DL in IoT?

3.3 Source of information

Due to the extensive scope of research articles, it is recommended that various databases be utilized to survey the existing literature in the field of the role of DL in IoT, due to the broad range of research articles. Six research databases were therefore searched in this study, including:

- Google Scholar (https://scholar.google.com/).
- IEEE (https://www.ieee.org/).
- Springer (Home—Springer).
- Elsevier (https://www.elsevier.com/en-in).
- Science Direct (https://www.sciencedirect.com/).
- ACM digital library (https://dl.acm.org/).

The choice of six research sources in this study was made based on several factors.

Firstly, these six databases are some of the most widely used and respected sources of scientific literature in the field of computer science and engineering, with a vast collection of research articles related to DL and IoT. Secondly, searching these six databases ensures that the study covers a broad range of research articles, which is important given the extensive scope of research in this field. Thirdly, these six databases offer a range of search functionalities and options, which allowed for a comprehensive and focused search on the topic of DL in IoT. Finally, the choice of six research sources was also guided by practical considerations such as time and resource constraints. Conducting a comprehensive search of all available databases would have been time-consuming and costly, and six databases were deemed to be a reasonable compromise between breadth and feasibility. Overall, the choice of these six research sources was based on their reputation, breadth of coverage, search functionality, and practical considerations, and was deemed to be appropriate for this study.

The following criteria's were taken into consideration when choosing the six research sources:

- Relevance: the sources had to be relevant to our research questions and objectives.
- Credibility: we selected sources that were published by reputable academic publishers and peer-reviewed journals.
- Recency: we preferred sources that were recently published (within the last 5 years) to ensure that the information was up-to-date.
- Accessibility: we considered sources that were easily accessible to us, such as open access or available through our institutional library.
- Diversity: we aimed to include sources from a variety of publishers and journals to ensure a diverse range of perspectives.

By using these criteria, we were able to identify six research sources that met our needs and provided a comprehensive and high-quality foundation for our paper.

4 Findings of the survey

The IoT has emerged as one of the most transformative technologies in recent times, enabling smart and interconnected systems that can operate autonomously and make intelligent decisions based on data. With the increasing amount of data generated by IoT devices and the need for intelligent decision-making, DL algorithms provide the necessary tools to process and analyze this data in real-time. As such, there has been a rising trend in adopting DL in IoT devices and domains, with a broad range of practical applications. In this survey paper, we investigate the versatility of DL in the realm of IoT and provide insights into the role of DL frameworks, DL models, challenges in implementing DL in IoT, and future directions for DL in IoT. Our findings highlight the potential of DL to revolutionize IoT devices and applications and provide recommendations for organizations looking to adopt DL in their IoT systems.

One of the key aspects of DL is the use of frameworks, which provide an interface for developers to build, train, and deploy DL models. There are several popular frameworks available, including TensorFlow, PyTorch, Deeplearning4j, Microsoft cognitive toolkit, Keras, MXNet, and Caffe. Each of these frameworks has its own strengths and capabilities, making it important to choose the right one for a given project.

DL models are used in a variety of ways in IoT, including for anomaly detection, predictive maintenance, image and speech recognition, time-series forecasting, data compression and dimensionality reduction, decision-making, and more. Some of the most commonly used DL models include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Autoencoders (AE), Generative Adversarial Networks (GANs), Long Short-Term Memory (LSTM) Networks, Variational Autoencoders (VAE), and others.

Implementing DL models in IoT devices can be challenging due to several factors, including limited computing power and memory, a lack of labelled data for training models, power consumption and energy constraints, limited bandwidth and connectivity, security and privacy concerns, integration with legacy systems and protocols, scalability and management of many devices, and cost and deployment barriers. To overcome these challenges, organizations can adopt techniques such as edge computing, federated learning, transfer learning, and compatibility standards.

DL will play a crucial role in the future of IoT by enabling devices to make decisions based on data and perform tasks with minimal human intervention. Integrating DL into IoT will improve the efficiency of various applications such as predictive maintenance, smart homes, and autonomous vehicles. The future direction for DL in IoT will likely include edge computing, predictive maintenance, anomaly detection, computer vision, and NLP.

However, DL models also have several computational limits, including data requirements, computational power, overfitting, data privacy and bias, interpretability, convergence, and generalization. It is important to consider these limitations when developing DL models for IoT applications.

5 Neural network

Neural networks imitate the functioning of the human brain, enabling computers to identify and predict patterns, as well as tackle complex challenges in the realm of ML and DL. Neural networks consist of interconnected processing units called neurons, which collaborate to analyze and process information. They have proven to be highly effective in solving complex problems and are widely used in various applications, such as image and speech recognition, NLP, and autonomous systems. This section provides a summary of neural networks, ANNs, and the learning process of neural networks.

5.1 Biological neurons

The whole idea behind DL is to have computers artificially mimic biological natural intelligence. Before proceeding to neural networks, we should probably build a general understanding of how biological neurons works. Neurons are the basic functional units of our nervous system, they generate biological signals which help them to transmit information. All neurons have three main units as shown in Fig. 2 which are:

- *Dendrites* (Li et al. 2020a) are responsible for receiving incoming signals, they make neurons produce and stop output signals.
- *Cell body (Soma)* (Zou et al. 2008) A single neuron can receive thousands of incoming signals through multiple dendrites and whether or not a neuron generates an output signal depends on the sum of all excitatory and inhibiting signals. The processing of this information occurs inside the cell body (soma).
- *Axon* (Zaimi et al. 2018) The axon is responsible for transmitting output signals from the neuron to targeted cells. In the brain, a network of neurons connected through chemical and electrical impulses is referred to as a neural network. Neurons use electrical signals or impulses to perform mental processes like thinking, memory recall,

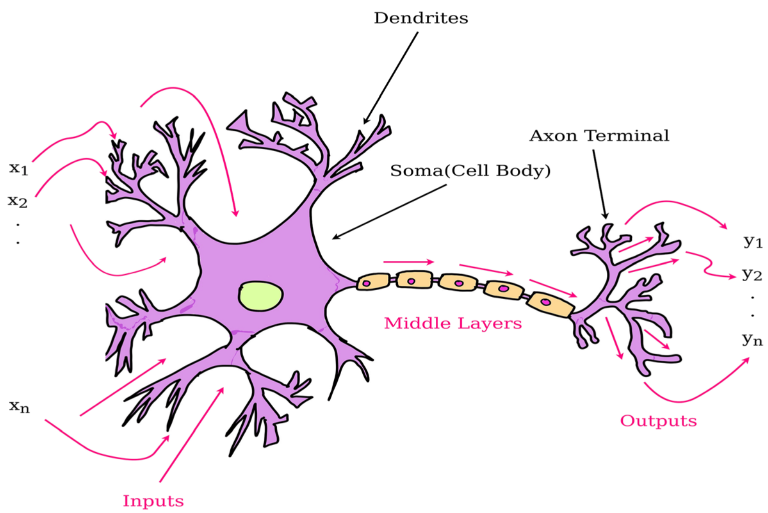


Fig. 2 Anatomy of a biological neuron showing input and output structures

and learning. The neuron sends the electrical impulse along its axon or nerve fibre. The axon terminates in several branches called dendrites. When the impulse reaches the dendrites, neurotransmitters are released into the space between cells. The cells across this space have receptors that attach to the neurotransmitters, leading to changes in the cells.

5.2 ANNs

ANNs are modelled on the architecture and operation of biological neurons in the human brain. In ANNs, the analogue of biological neurons are layers comprised of interconnected nodes that transmit signals to other nodes (Pantic et al. 2022). In an ANN, each node is connected to other nodes with a weight and threshold value. When the output of a node surpasses the threshold, the node is activated and forwards information to the next layer of the network. If not, no data transfer occurs. The structure of an ANN is depicted in Fig. 3 and consists of the following elements:

- Input layers, where the input data is entered from the outside world and then processed and categorized.
- Hidden layers, there are a lot of hidden layers in a neural network, where the processing of the input data takes place.
- Output layers, where the response is generated and delivered, in one output node the output is found to be either 0 or 1.

Every node takes data input and assigns a weight to it. The node with more weight contributes more to the output than the node with less weight. Some of the ANNs models in practice are CNN, Multi-Layer Perceptrons (MLP) and RNN.

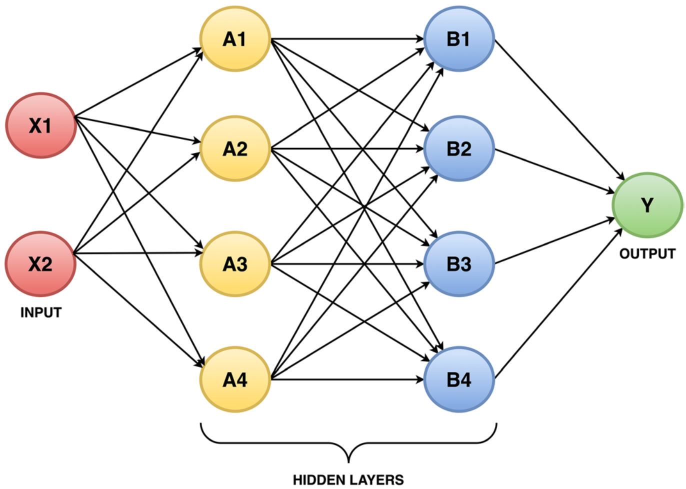


Fig. 3 Schematic diagram of an ANNs, highlighting its layers and connections

5.3 Perceptrons

The concept of perceptron was introduced by American psychologist, Frank Rosenblatt (https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon) in 1957 at Cornell Aeronautical Laboratory. Perceptrons are basic functional units of ANNs. They are mathematical functions which take some set of inputs along with parameters and generate output. Each perceptron has three basic functions:

- Take inputs.
- Associate weights with inputs and sum them up.
- Pass the sum in some function to generate output.

Perceptrons consist of a single node, or neuron, that receives input, processes the input through a set of weights and biases, and produces a single output. The weights and biases are adjustable, allowing the perceptron to learn from the input–output pairs it is presented with during training. Perceptrons are considered to be the building blocks of more complex neural networks and are often used for binary classification problems, where the goal is to separate the input into two distinct classes based on the output.

A single perceptron cannot learn complex systems, but a multi-layered perceptron, also known as a neural network, can. This is achieved by connecting multiple perceptrons and using the output of one as input for the others, enabling the network to learn about the interactions and relationships between features. Figure 4 illustrates the functioning of a multi-layered perceptron model. The functionalities of different layers are:

- The first layer in a neural network is the input layer, which receives the input vector "x" for the network to learn from. The input layer has the same number of neurons as the entries in the input vector, with each neuron corresponding to a specific entry.
- The final layer in a neural network is the output layer, which can contain multiple neurons. The output layer produces the vector "y," representing the result generated by the network.

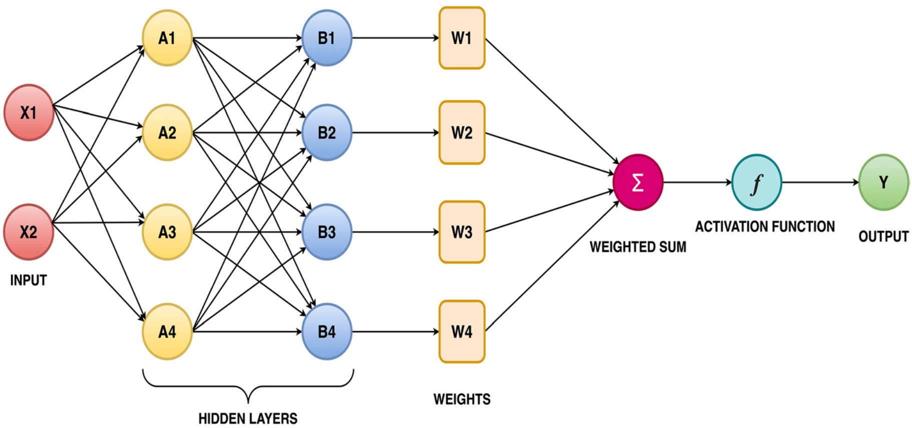


Fig. 4 Perceptron model depicting its key components: inputs, weights, bias, and output

The layers between the input and output layers in a neural network are referred to as the hidden layers. These layers perform mathematical operations to generate a prediction vector "y".

Here is how the perceptron model operates:

- Input values are fed to the model.
- The input values are multiplied by the weights to calculate the weighted sum and a bias term is added.
- The procured result decides whether the perceptron will activate or not. If the result is above the threshold then the perceptron will fire, otherwise, it will not.

The perceptron is considered to be working well if the predicted output is close to the actual output for a given input. To minimize the impact of these errors on future predictions, the weights need to be adjusted if there is any discrepancy between the expected and actual results. The correct weights and biases in a perceptron are determined through a process called training. During training, the model is presented with a set of input–output pairs, and the weights and biases are adjusted to minimize the error between the predicted and actual outputs. This adjustment of weights and biases is done iteratively until the model produces an acceptable level of accuracy. Common training algorithms include gradient descent, stochastic gradient descent, and backpropagation.

5.4 Functioning of neural network

Neural networks learn by adjusting the connections between their artificial neurons in response to input data. This process is called training, and it involves feeding the network large amounts of labelled examples so that it can gradually improve its ability to make predictions or decisions. During training, the network uses a mathematical algorithm called backpropagation to calculate the error or loss between its predictions and the actual labels. It then uses this error signal to adjust the weights of its connections, in a way that reduces the overall error. Over time, with enough training data and iterations, the network learns to recognize patterns and relationships in the data, and can make accurate predictions even on new, unseen examples. This ability to generalize from training data is what makes neural networks so powerful and versatile for tasks such as image recognition, language translation, and decision making.

A neural network learns by modifying its weights and biases through a training process that involves several steps. The first step is forward propagation, where the input data is processed and used to generate a prediction for the output. The loss function is then used to measure the difference between the predicted output and the actual output, and this is minimized through a process called backpropagation. During backpropagation, the network adjusts its weights and biases based on the error calculated by the loss function. Gradient descent is used to update the weights and biases in a direction that minimizes the loss, and this process is repeated until the minimum loss is achieved. This cycle of forward propagation, loss calculation, backpropagation, and gradient descent is repeated multiple times until the network converges to a set of weights and biases that result in accurate predictions. By learning the underlying patterns in the data, the neural network can improve its predictions over time.

The forward propagation process (Tang et al. 2015) in a neural network involves taking input X and using it to calculate a prediction vector Y. This is done by multiplying the input vector X and weight matrix W between two layers to produce a vector Z. The activation function is applied to convert Z into the output of the perceptron and fed to the next layer. In a multi-layer neural network, this process is repeated using different weight matrices for every two consecutive layers. The value of a perceptron in a layer is a linear combination of activations from the previous layer and some weights. During training, these weights are adjusted to improve the accuracy of the predictions. After making predictions, the neural network is evaluated using a loss function to measure the difference between the prediction and actual value (Li et al. 2019). Equation 1 shows the mathematical expression for evaluating the loss value. Where L is the loss, y_{pred} is the predicted value, and y is the actual value

$$L = \frac{1}{2}(y_{pred} - y)^2 \tag{1}$$

Minimizing the loss function leads to more accurate predictions, with two of the most commonly used loss functions being Mean Squared Error and Cross-Entropy. To minimize the loss function and improve the predictions, gradient descent is used, which consists of forward propagation to calculate the prediction vector and backpropagation to fine-tune the weights and bias terms. Backpropagation involves travelling from the output layer to the input layer, adjusting the weights and bias terms using the gradient of the loss function. This process continues until the optimal weights are reached and the neural network is capable of making desired predictions.

6 Deep Learning

Deep Learning (DL) is a branch of AI that emulates the human brain in processing data and generating patterns that are valuable in decision-making. The term ‘Deep’ refers to the multiple layers in the model, while ‘Learning’ refers to the ability to learn from these layers. In this way, DL models analyse input data to extract high-level abstractions across multiple layers. The goal of DL is to educate computers to learn through examples and previous experiences.

DL focuses on creating ANNs with multiple layers, referred to as deep neural networks. These networks are designed to automatically learn from large amounts of data and make predictions or decisions based on input data. DL has proven to be highly effective in solving complex problems in various domains, such as image and speech recognition, NLP, and self-driving cars. This success is due to the ability of DL algorithms to model high-level abstractions and representations of data, allowing them to make highly accurate predictions.

DL models offer high accuracy in recognizing patterns in real-life scenarios by being fed a training set of labelled data. During the training phase, data is provided to the computer to learn from, while during the testing phase, the trained system is given new data and asked to predict the label. Deep neural networks consist of three layers: input, hidden, and output. Each layer contains multiple units called neurons, which receive multiple inputs, calculate a weighted sum of these inputs, and produce an output through an activation function. Each neuron has a weight and bias that are optimized during training.

DL is playing a crucial role in several cutting-edge technologies such as driverless cars (Kuutti et al. 2020), voice assistants (Andics et al. 2010), sentiment analysis (Zhang et al. 2018a), healthcare (Esteva et al. 2019), agriculture (Kamilaris and Prenafeta-Boldú 2018) and many more. According to LeCun et al. (2015a), DL has made significant progress in solving complex problems that have challenged AI for many years. It is particularly adept at uncovering intricate patterns in high-dimensional data and is therefore applicable to numerous fields including engineering, science, business, and agriculture. DL has shown outstanding results for various tasks in NLP, sentiment analysis, question answering and language translation.

A strong advantage of DL is feature learning (Li et al. 2014), i.e. DL extracts the features automatically from raw data. More complex problems can be easily solved by DL in a fast manner with the help of a powerful computer system. DL is a rapidly growing field in AI that focuses on learning high-level representations of data. It uses hierarchical architectures to identify patterns and relationships in complex data and has seen successful application in various domains, including semantic parsing (Quispe and Pedrini 2019), transfer learning (Shao et al. 2014), NLP (Otter et al. 2020), computer vision (Alyamkin et al. 2019), and more. With its increasing popularity and proven effectiveness, DL is expected to continue advancing and expanding into new areas in the future.

In this section, we will provide an overview of several key aspects of DL, including the significance of this rapidly advancing field, the importance of activation functions, and various DL models such as CNN, RNN, AE, VAE, and GAN. Additionally, we will explore DL frameworks and their role in enabling the development and implementation of complex models in a more efficient and streamlined manner. By gaining an understanding of these important components, we can appreciate the significant impact that DL is having across numerous industries and fields, and its potential for continued growth and innovation in the future.

6.1 Significance of DL

Before the surge in the popularity of DL, conventional ML algorithms such as SVM, Linear Regression, and Logistic Regression were utilized. These algorithms faced a challenge in that they were not capable of processing raw data directly, instead requiring a laborious and knowledge-intensive step known as Feature Extraction. However, Neural Networks became sought after due to their capability of automatically extracting features from raw data without the need for pre-processing, a process known as Feature Learning. The abundance of data is another factor contributing to the rise of DL. With vast amounts of information available, there are numerous opportunities for advancements in this field, leveraging the power of big data to drive innovation.

Dr Andrew Ng (https://scholar.google.com/citations?user=mG4imMEAAAAJ&hl=en&oi=ao) is a renowned professor and AI expert who has compared DL to rocket science. He has stated that DL models are like rocket engines and the large amounts of data fed to them serve as fuel, which helps to propel the models towards their intended goal. This comparison highlights the importance of having large amounts of data to train DL algorithms and the complexity involved in designing and building these models.

The key advantage of DL models is that they can learn to make predictions or decisions based on the data without human intervention or feature engineering. The models can be trained using large amounts of data, and they automatically learn features and representations from the data, allowing them to perform feature extraction without the need for preprocessing steps. The rise of DL is also attributed to the availability of vast amounts of data, which provides opportunities for innovation in the field. With advancements in technology, DL is being widely used in industries such as finance, healthcare, transportation, and retail, and is continuously advancing with researchers developing new architectures and techniques to improve performance. The future looks bright for DL, with many challenges and opportunities to improve and bring it to new frontiers.

6.2 Activation functions

The activation functions (Liu and Wang 2008) are essential components in neural networks, serving a critical role in their operations. They generate the output of the network, which ranges from 0 to 1 or − 1 to 1 depending on the activation function used. These functions are divided into two categories: Linear functions and Non-Linear functions. Linear functions do not limit the output of the network to any specific range, while non-linear activation functions are the most commonly used and their outputs are confined to certain ranges. The following listed are the activation functions that are mostly used:

- Linear (Agostinelli et al. 2014): The output of this function is not confined. It simply returns the input as the output, without applying any transformation to it. In other words, it performs a simple linear transformation on the input data. Equation 2 is the mathematical equation of the linear activation function:

$$f(x) = x \tag{2}$$

- Sigmoid (Langer 2021): Function is differentiable, commonly used in binary classification problems, where the output of the network should represent the probability of the input belonging to one of two classes. Equation 3 is the mathematical equation of the sigmoid activation function:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

- Tanh (Karlik and Olgac 2011): It is preferred over the sigmoid activation function for some problems because it has a wider range of output values and is zero-centred, which can help with optimization during training. Equation 4 is the mathematical equation of the Tanh activation function:

$$tanh = \frac{2}{1 + e^{-x}} - 1 \tag{4}$$

- ReLu (Schmidt-Hieber 2020): ReLU has become a popular activation function in DL due to its ease of computation and ability to prevent the vanishing gradient problem associated with functions such as sigmoid and tanh. Despite its advantages, ReLU is prone to the “dying ReLU” problem, in which a neuron’s output becomes stuck at zero and cannot learn during training. Equation 5 is the mathematical equation of the ReLu activation function:

$$f(x) = max(0, x) \tag{5}$$

- Leaky ReLu (Xu et al. 2020a): The Leaky ReLU (rectified linear unit) activation function is a variation of the traditional ReLU function utilized in ANNs. It addresses the issue of “dying ReLU”, where a neuron’s output can become zero for negative inputs, by having a slight negative slope for inputs below zero, determined by the constant alpha (usually set to 0.01). Equation 6 is the mathematical equation of the Leaky ReLu activation function:

$$f(x) = max(\alpha x, x) \tag{6}$$

- ArcTan (Zhang et al. 2018b): Also known as the inverse tangent activation function. It is a smooth, non-linear activation function that can be used as an alternative to popular activation functions like ReLU, sigmoid, and tanh in ANNs. Equation 7 is the mathematical equation of the arctan activation function:

$$f(x) = tan^{-1}(x) \tag{7}$$

- Softmax (Chen et al. 2018): Function gives out probabilities of the states of input classes, it has the properties of converting the input vector into a probability distribution, ensuring that the outputs for all classes sum to 1, and representing the confidence of the model's prediction for each class. Equation 8 is the mathematical equation of the softmax activation function, where "z" is a vector of real numbers representing the input to the activation function, and "k" is the number of elements in the vector.

$$\frac{e^{z(i)}}{\sum_{j=1}^{k} e^{z(j)}} \tag{8}$$

- Swish (Ramachandran et al. 2017): The Swish activation function offers advantages over conventional activation functions, such as a smoother gradient and reduced saturation for large inputs. Some research has also indicated that the Swish function can result in better performance than ReLU and other activation functions in specific neural network configurations. Equation 9 is the mathematical equation of the swish activation function, where "x" represents the input to the activation function. It can be a scalar value, a vector, or a tensor, depending on the input to the neural network.

$$f(x) = x \times sigmoid(x) \tag{9}$$

Table 2 presents the category and output range of various activation functions.

Table 2 Category and output range of various activation functions

Activation function	Category	Output range
Linear (Agostinelli et al. 2014)	Linear	$-\infty$ to ∞
Sigmoid (Langer 2021)	Non-linear	0 to 1
Tanh (Karlik and Olgac 2011)	Non-linear	-1 to 1
ReLU (Schmidt-Hieber 2020)	Piecewise linear (Rebennack and Krasko 2020)	0 to ∞
Leaky ReLU (Xu et al. 2020a)	Piecewise linear	$-\infty$ to ∞
ArcTan (Zhang et al. 2018b)	Non-linear	$\frac{-\Pi}{2} to \frac{\Pi}{2}$
Softmax (Chen et al. 2018)	Non-linear	0 to 1
Swish (Ramachandran et al. 2017)	Non-monotonic (Misra 2019)	1 to ∞

6.3 DL models

DL models find extensive use in diverse industries for resolving intricate issues. For instance, in computer vision, they assist with image classification, object detection, and segmentation. Similarly, in the domain of NLP, DL models come in handy for tasks like language translation, text classification, and sentiment analysis. In healthcare, DL models are utilized for image analysis, diagnosis, and drug discovery. In finance, they are used for fraud detection and algorithmic trading. DL models can also be applied in areas such as speech recognition, recommendation systems, and autonomous vehicles. Their ability to learn from large amounts of data and perform complex tasks has made DL a key player in the advancement of AI and ML. DL models play a crucial role in IoT data processing and analysis by allowing:

- Anomaly detection: DL models can identify patterns in large amounts of IoT data and flag any data points that deviate from these patterns as anomalies.
- Predictive maintenance: DL models can be trained on IoT data to predict when a device is likely to fail, allowing maintenance to be performed proactively.
- Image and speech recognition: CNNs and RNNs can be used to process and analyze image and speech data generated by IoT devices.
- Time-series forecasting: LSTMs and other RNN models can be used to analyze time-series data generated by IoT devices and make predictions about future trends.
- Data compression and dimensionality reduction: AE and other DL models can be used to reduce the size of IoT data and remove any redundant information.
- Decision-making: Reinforcement learning can be used to train models that make decisions based on IoT data, for example, in the case of an autonomous vehicle or industrial control system.

In this section, a comprehensive overview of the most commonly used DL algorithms has been provided.

6.3.1 CNN

A CNN is a DL algorithm that excels in analysing image pixels and recognizing patterns in images, audio, and signals. It is widely used for various purposes, including medical imaging, audio processing, object detection, and generating synthetic data. A CNN operates by accepting an image as input, determining the significance of various objects or elements within the image through adjustable weights and biases, and differentiating between them. It accomplishes this by utilizing filters that can effectively identify spatial and temporal relationships within an image. The CNN structure is optimized for image data, as it reduces the number of parameters and recycles weights. The ultimate goal of a CNN is to extract higher-level concepts from image information through the application of multiple non-linear layers. CNNs have demonstrated exceptional results in various fields related to pattern recognition (Wu et al. 2018), image processing (Han et al. 2020), speech recognition (Noda et al. 2015), and more. CNNs are a specific type of DL model that have made significant contributions to the field of computer vision and image analysis. The architecture of CNN is classified into two main parts:

Table 3 Available CNNs models

Model	Year
LeNet-5 (LeCun et al. 2015b)	1998
AlexNet (Alom et al. 2018)	2012
VGG-16 (Simonyan and Zisserman 2014)	2014
Inception-v1 (Szegedy et al. 2015)	2014
GoogLeNet (Ballester and Araujo 2016)	2014
Inception-v3 (Szegedy et al. 2016)	2015
ResNet (He et al. 2016)	2015
Inception-v4 (Szegedy et al. 2017)	2016
Xception (Chollet 2017)	2016
ResNext-50 (Xie et al. 2017)	2017

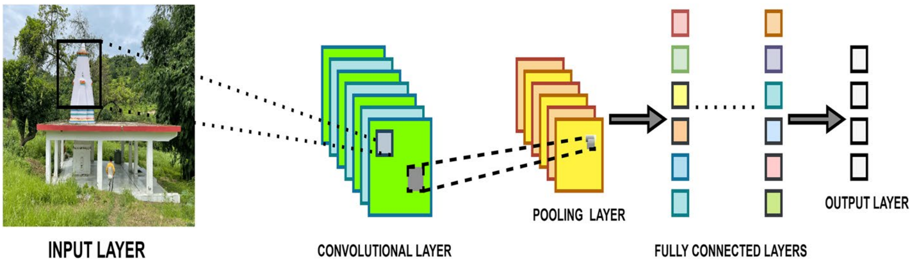


Fig. 5 CNN architecture, featuring specialized convolutional and pooling layers, and FC layers

- Feature extraction: this is a tool that separates and recognizes the different features of the image for processing/analysis.
- Classification: output generated from the convolutional process is utilized by a fully connected (FC) layer and prediction of the class of the image is done based on the features extracted in the previous stage.

CNNs are known for their accuracy, especially in the field of image analysis and target detection. Two challenges in training CNNs are the number of training examples and the duration of training. To obtain an accurate CNN model, it is crucial to have a substantial number of examples in the training dataset. The features learned by a CNN model are highly accurate and superior to manually crafted features. Fortunately, the availability of pre-trained CNN models on popular image datasets simplifies the process of setting up or adjusting CNN models for computer vision tasks. A commonly used CNN architecture was first proposed by LeCun (https://scholar.google.com/citations?user=WLN3QrAAAAAJ&hl=en&oi=ao) and his colleagues for handwriting recognition, which now has become a benchmark for DL models. Table 3 provides the list of various available CNNs models.

The architecture of a CNN is illustrated in Fig. 5, consisting of convolution layers, pooling layers, and FC layers. Arranging these layers in a specific sequence creates the CNN architecture. Along with these three layers, CNNs also include two crucial components: the dropout layer and the activation function.

- Convolutional layer: the initial stage of a CNN structure is the convolutional layer, which extracts different features from the input image by applying mathematical convolution operations, creating a feature map that contains image information.
- Pooling layer: the pooling layer helps to decrease the computational cost by decreasing the size of the feature map obtained from the convolutional layer. This is accomplished by cutting down the connections between layers and processing each feature map separately. Different pooling methods can be utilized to enhance the efficiency of the model, including:
  - Max Pooling (Ma et al. 2019a).
  - Average Pooling (Wang et al. 2017).
  - Sum Pooling (Liu et al. 2016).
- FC layer: the FC layer is composed of neurons, weights, and biases that link neurons across different layers. It is commonly located just prior to the output layer, serving as the final component of a CNN structure. The input image obtained from the preceding layers is converted into a flattened shape and channeled into the FC layer. The resulting flattened vector undergoes mathematical operations as it advances through subsequent FC layers, signifying the start of the input image's categorization or classification procedure.
- Dropout: when assessing a model, it is important to consider the issue of overfitting in the training dataset. Overfitting can occur when a model performs very well on the training data but does not generalize well to new data. To tackle this issue, a dropout layer can be added to the model. The dropout layer excludes or "drops out" specific units or neurons from the model during training in a random manner, thereby decreasing the model's complexity and minimizing the chances of overfitting.
- Activation function: activation functions are an essential hyperparameter in CNN models. They are used to learn and approximate complex relationships between variables in the network and add non-linearity to the network. The most commonly used activation functions were discussed in Sect 6.2.

Table 4 provides information about the work done by various researchers using CNN.

6.3.2 RNN

A RNN is a state-of-the-art neural network algorithm that is specialized in dealing with sequential data. They are commonly used in NLP (Cambria and White 2014) tasks due to their ability to handle text and other successive information. They are utilized in technologies such as Apple's Siri and Google's voice search (Park et al. 2019). RNNs are particularly useful because they have an internal memory which allows them to retain information from previous steps, making them well-suited for AI problems that involve sequential data. They are considered to be among the most powerful algorithms in the history of DL advancements in recent years.

In RNN nodes are connected in a way that forms cycles, meaning the output of a node can influence the output of another node. The output from the previous step is incorporated as part of the input in the current step. This neural network is commonly used in applications where the previous output is necessary to determine the output puts. For example, in

Table 4 Usage of CNNs in various types of research

Author(s)	Work done	Model
Lee et al. (2018)	Extract the characteristics of images, and then utilize the AdaBoost method to construct a classifier for identifying the images	CNN
Ma et al. (2019b)	developed a model called LungBRN for identifying respiratory illness	ResNet
Sharma et al. (2017)	Proposed a model for getting land cover information through remote sensing images	CNN
Khan et al. (2019)	Analyse ResNet and GoogleNet models for the detection of malware	ResNet, GoogleNet
Liu et al. (2019a)	Fruit detection on RGB and NIR images	VGG16
Wen et al. (2020)	Proposed a model for fault diagnosis using transfer learning	ResNet-50
Lu et al. (2019)	Suggested a DL model that automatically detects pathological brain in Magnetic Resonance Images (MRI)	AlexNet
Shanthi and Sabeenian (2019)	Proposed a model for early detection of diabetic retinopathy disease	Alexnet
Zhang et al. (2017a)	Denoising of images using the DL model	DnCNN
Baloglu et al. (2019)	Developed a model for detecting myocardial infarction (heart attack) using ECG lead signals	DeepCNN
Kollias and Zafeiriou (2020)	Put forward a new CNN-RNN approach utilizing various features of CNN for recognizing subtle facial expressions of human emotions	CNN, RNN
Mao et al. (2020)	For the effective classification of Facial Attributes, a multi-label model based on CNN has been proposed	CNN
Tao et al. (2020)	Recognition of emotions based on electroencephalography (EEG) and attention-based Convolutional Recurrent Neural Network	ACRNN
Li et al. (2021a)	Recognition of emotions based on EEG	RNN
Jia et al. (2020)	Developed a system for understanding user behavior that utilizes contactless Radio Frequency (RF) technology and leverages the WiFi Channel State Information (CSI)	CNN
Ma et al. (2021)	Recognizes facial expressions	CNN
Xu et al. (2021)	Predict security performance for IoT-based healthcare networks	CNN
Zhang et al. (2021a)	Classify hyperspectral images (HSI) and light detection and ranging (LiDAR) data	CNN
Jia et al. (2022)	Proposed a resource optimization approach in Edge Computing Environment	CNN
Kim et al. (2020)	Proposed a CNN accelerator that utilizes feature skipping to improve energy efficiency in mobile devices for face recognition tasks	CNN

sentence auto-completion, previous words are taken into consideration before the neural network can provide suggestions to complete the sentence.

RNNs use memorization to store information from previous steps that have already been calculated. They use the same parameters every time to operate on the input or the hidden layers. This is different from traditional neural networks that have independent layers and do not store previous outputs in memory. RNNs use dependent activations, which means they use the same weights and biases for all hidden layers, reducing the complexity of increasing parameters by remembering the outputs that are fed as input to the next layer. Equation 10 is the formula for calculating the current hidden state is:

$$H_t = f[(H_t - 1) + A_T] \tag{10}$$

where H_t is the current state, $H_t - 1$ is the previous state and A_t is the input. Now that the hidden state is calculated, we can find out the output by the following formula:

$$B_t = [W_0 + H_t] \tag{11}$$

where B_t is the output, W_t is the weight at the output layer, and H_t is the previously calculated hidden state. Figure 6 shows the architecture of RNN. In a RNN, the nodes from multiple layers are condensed into a single layer.

The network's parameters are represented by F, G, and H. In this context, "X" represents the input layer, "A" represents the hidden layer, and "Y" represents the output layer. The network parameters F, G, and H are utilized to optimize the model's output. At a given time step "t", the current input is a combination of the input at X(t) and the previous input X(t − 1). The output at any given time is fed back into the network to improve its performance. In RNN, the information flows in a loop back to the intermediate hidden layer. The input layer "X" processes and receives the input for the neural network and transfers it to the middle layer. The middle layer "A" can comprise multiple hidden layers, each with its own set of activation functions, weights, and biases. If the parameters of the various hidden layers are not impacted by previous layers, meaning the network does not have memory, then a RNN can be used. The RNN standardizes the activation functions, weights, and biases across all hidden layers, making them all have the same parameters. Instead of

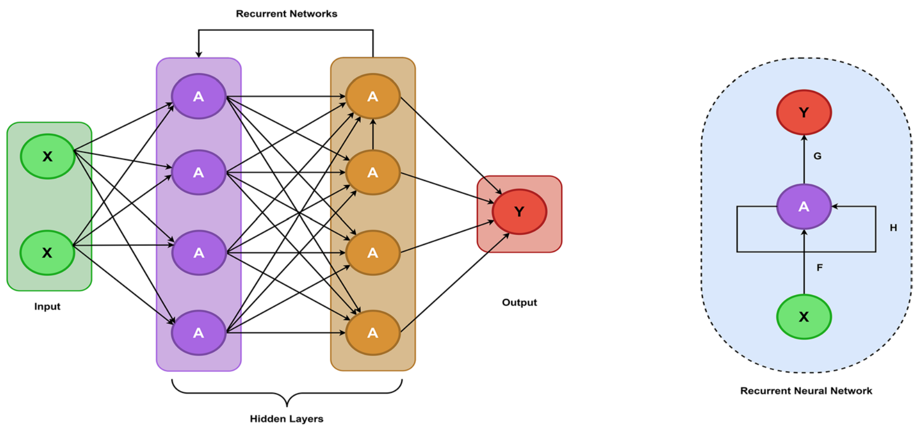


Fig. 6 Schematic representation of a RNN, emphasizing its recurrent connections and hidden state

multiple hidden layers, it creates a single layer and repeats it as necessary. Advantages of RNNs include:

- Memory: RNNs have an internal memory that allows them to retain information from previous steps, making them well-suited for AI problems that involve sequential data.
- Handling sequential data: RNNs are specialized in dealing with sequential data, making them commonly used in tasks such as NLP and speech.
- Predictions: RNNs are useful for providing predictions, such as stock price prediction and auto-complete features in various word processing software.
- Image processing: RNNs can be used in conjunction with CNNs to extend the effective pixel neighbourhood, such as extending images or removing objects from an image.
- Real-time processing: RNNs can process data in real-time, which makes them useful in applications such as speech recognition and machine translation.
- Handling long-term dependencies: RNNs are capable of handling long-term dependencies, which makes them useful in tasks such as language modelling and time series prediction.

Disadvantages of RNNs include:

- Training difficulty: training an RNN can be difficult due to the complex nature of the network and the need to handle sequential data.
- Vanishing and exploding gradients: when the gradient is too large or too small, it can make it difficult for the network to learn.
- Limited ability to process long sequences: if activation functions such as read or tanh are used, RNNs may have difficulty processing very long sequences.
- Large computational cost: RNNs can be computationally expensive, particularly when working with large datasets or when multiple layers are used.
- High memory requirements: RNNs require large amounts of memory to store the information from previous steps, which can be a limitation when working with large datasets.
- Inability to parallelize: RNNs are inherently sequential and it's difficult to parallelize their computations, which can slow down the training process.
- Limited interpretability: the internal workings of RNNs can be difficult to interpret, making it hard to understand how the network is making its predictions.

Table 5 provides information about the work done by various researchers using RNN.

6.3.3 LSTM

LSTM is a type of RNN that is specifically designed to handle long sequences of data. It is particularly useful for training models on time series data, where the order of the data points is important. One of the main challenges with training RNNs on long sequences is the problem of vanishing gradients, where the network struggles to learn patterns in the data. LSTM addresses this problem by using a memory cell, which allows the network to decide when to erase certain information and keep other information. This allows LSTM to effectively learn temporal dependencies in sequences. However, LSTMs still have difficulty learning long-term dependencies in very long sequences. To address the problem of

Table 5 Usage of RNNs in various research

Author(s)	Work done	Model
Xia et al. (2021)	Introduced a GRU-based approach for predicting the demand for electricity and renewable energy in smart grid networks	RNN
Wang et al. (2020a)	A method for forecasting and preventative maintenance of high-speed railway power equipment using LSTM-RNN algorithm has been proposed	LSTM, RNN
Jin et al. (2020)	Proposed an efficient method using RNNs to solve time-varying Generalized Sylvester Equations, with potential applications in areas such as robotic control and identifying the location of acoustic sources	RNN
Li et al. (2020b)	Proposed a rasterization-based architecture for recognizing vector sketches	RNN
Wang et al. (2021a)	DRLIVE, a visual analytics system, has been created to efficiently examine, comprehend, and diagnose deep reinforcement learning models based on RNNs	RNN
Zhang et al. (2022)	Proposed a RNN approach for predicting the length of time a car user will stay in a given location	RNN
Chen et al. (2020a)	Evaluated data-driven rollover risk for articulated steering of vehicles	RNN
Li et al. (2022)	Developed a model for jointly modelling and forecasting time series and event sequence data types	RNN, VAE
Khan et al. (2020a)	An RNN-based metaheuristic optimization method has been proposed to regulate mobile robots tracking control while imposing nonholonomic restrictions	RNN
Habi and Messer (2020)	Determined rain rate using commercial microwave links	RNN
Jin et al. (2020)	A RNN model with noise suppression has been suggested as a solution for the time-variant generalized Sylvester equation	RNN
Feigl et al. (2020)	Determined human velocity estimation by combining a CNN and a bidirectional RNN	RNN, CNN
Zhou et al. (2020)	An intelligent recommendation has been proposed for online medical prediagnosis support	CNN, RNN
Liu et al. (2019b)	Suggested a DL-based approach to analyze payloads for attack detection	RNN, CNN
Xie et al. (2020a)	Suggested an orthogonal scheme for repetitive motion generation of redundant robot manipulators	RNN
Qi et al. (2019)	An RNN-based approach has been used for activity recognition and group activity analysis	RNN
Hadjeres and Nielsen (2020)	Suggested an RNN-based DL model for sequence and music generation	RNN
Tan et al. (2020)	A self-attentive gated RNN has been proposed for binaural speaker separation	RNN
Tasyurek and Celik (2020)	A regression-based RNN has been developed for frequently updated data of geographically weighted areas (ecology, environmental management, public health, tourism)	RNN

learning long-term dependencies in very long sequences, the attention mechanism can be used in conjunction with LSTMs (Karim et al. 2017). The attention mechanism allows the network to focus on specific parts of the input sequence, rather than processing the entire sequence as a whole. This helps the network to learn the dependencies in the data more effectively.

An LSTM network has an input layer, an output layer and some hidden layers, as shown in Fig. 10. The main idea of LSTM is that the memory cells present in the hidden layer can retain their state and the non-linear gating components control the flow of data into and out of the cell (Greff et al. 2016). Each LSTM layer contains a memory cell, input gates, output gates and forget gates. This structure allows LSTM to manage the flow of information, determining which information should be forgotten and which should be remembered.

There are different variations of LSTM such as PC-LSTM (Rao et al. 2018), CIFG-LSTM (Ji et al. 2018), GRU (Gao et al. 2020), and Bi-LSTM (Yu et al. 2020). These variations have been proposed to improve the performance of LSTM on long sequences. The cell state acts as a transport highway that transfers relevant information down the sequence chain. In this way, LSTM can improve the performance of long sequences.

Some of the advantages of LSTM are:

- Handling of long-term dependencies: one of the main advantages of LSTM is its ability to handle long-term dependencies in sequences. It can learn patterns in the data that span over a long period.
- Handling of sequential data: LSTM is specifically designed to handle sequential data, making it suitable for tasks such as speech recognition, language modelling, and time series forecasting.
- Handling of missing data: LSTM can handle missing data, by selectively choosing which information to retain and which to discard.
- Robustness to noise: LSTM is robust to noise in the input data, due to its ability to selectively retain relevant information and discard irrelevant information.
- Variety of architectures: LSTM has a variety of architectures such as PC-LSTM, CIFG-LSTM, GRU, and Bi-LSTM which can be used to improve the performance of the model depending on the task and data.
- Attention Mechanism: LSTM can be combined with an attention mechanism which helps to focus on specific parts of the input sequence, thus allowing the network to learn the dependencies in the data more effectively

Some of the disadvantages of LSTM are:

- Computational complexity: LSTM networks can be computationally expensive, as they have more parameters and require more computation than traditional RNNs.
- Difficulty in parallelization: LSTM networks are difficult to parallelize, which can make training on large datasets time-consuming.
- Overfitting: LSTM networks can be prone to overfitting, especially when trained on small datasets.
- Difficulty in handling very long sequences: LSTM networks can have difficulty handling very long sequences, as the information stored in the memory cells may become irrelevant over time.

Table 6 Usage of LSTMs in various research

Author(s)	Work done	Model
Essien and Giannetti (2020)	Proposed a DL-based model for efficient and smart manufacturing in factories	LSTM, AE
Chen et al. (2021a)	To reveal real-time traffic demand for traffic management, a bidirectional unidirectional-LSTM network has been proposed	LSTM
Azzam et al. (2020)	To reduce semantic pose estimation error a stacked-based LSTM model has been developed	LSTM
Du et al. (2020)	Proposed an LSTM-based network to recognise emotions from n-channel EEG signals	LSTM
Santo et al. (2020)	Developed an application for hard disk drive health assessment using DL	LSTM
Yuan et al. (2020)	Proposed an approach for Industrial Soft Sensor Model Development using SpatIoTemporal LSTM	LSTM
Jahangir et al. (2020)	Proposed a Forecasting approach in smart grids of electricity	LSTM
Ma and Mao (2020)	A DL-based model has been developed to predict the remaining life of an electronic component	LSTM
Wu et al. (2021)	Predicts the remaining useful life of the components	LSTM, AE
Ye et al. (2020)	To tackle the problem of referring to image segmentation a DL-based solution has been given	LSTM
Khalil et al. (2019)	To reduce the computational cost of RNN, an LSTM approach has been given	LSTM, RNN
Li et al. (2020c)	Detects phishing attacks on huge email data	LSTM
Bandara et al. (2020)	Generates forecasts for time series data of industries' production with different season cycles	LSTM
Khairdoost et al. (2020)	Predicts the real-time driver manoeuvre	LSTM
Ke and Vikalo (2021)	Classifies radio technologies and modulation	LSTM
Wang et al. (2021b)	Predicts risk assessment in the local path for autonomous vehicles	LSTM
Li et al. (2020d)	Predicts the volume of Long-Term traffic	LSTM
Ma et al. (2020)	Proposed a method to identify unauthorised broadcasting	LSTM
Lin et al. (2020)	To reduce the high uncertainty effect for the effective working of wind turbine power, a DL-based framework is developed	LSTM

- Difficulty in interpreting the model: LSTM networks can be difficult to interpret, as it is not always clear how the network is using the stored information to make predictions.
- Vanishing Gradients: LSTM networks suffer from the problem of vanishing gradients when learning through long data sequences, which makes it difficult to train the network.

Table 6 provides information about the work done by various researchers using LSTM.

6.3.4 GANs

GANs are a form of neural network that can learn intricate and high-dimensional distributions by instructing two networks, namely a generator and a discriminator, to compete against each other (Xu et al. 2020b). The generator produces fresh examples that aim to imitate actual data, whereas the discriminator endeavors to differentiate between the genuine and generated samples. Over time, the generator enhances its capacity to create realistic samples, and the discriminator improves its ability to identify counterfeit ones. Nevertheless, one of the predicaments encountered with GANs is the phenomenon of mode collapse, wherein the generator produces a restricted set of data variations. To surmount this issue, researchers have developed techniques like conditional GANs (Iqbal and Ali 2018; Jiang et al. 2019), which employ supplementary information to guide the generator's output. GANs find wide-ranging utility in several applications, including image synthesis, text-to-speech, and anomaly detection.

Figure 7 shows the architecture of a GAN; A GAN consists of a generator and a discriminator. The generator's task is to create realistic data, which is then used as fake examples for the discriminator to distinguish from real data. The discriminator's goal is to identify the generator's fake data and penalize it for producing unrealistic results. At the start of training, the generator produces easily recognizable fake data, which the discriminator quickly learns to identify. As training continues, the generator improves and becomes better at creating convincing fake data that the discriminator finds difficult to distinguish from real data. Eventually, if the generator is trained effectively, the discriminator may start mistaking fake data as real and its accuracy will decline. Both the generator and discriminator

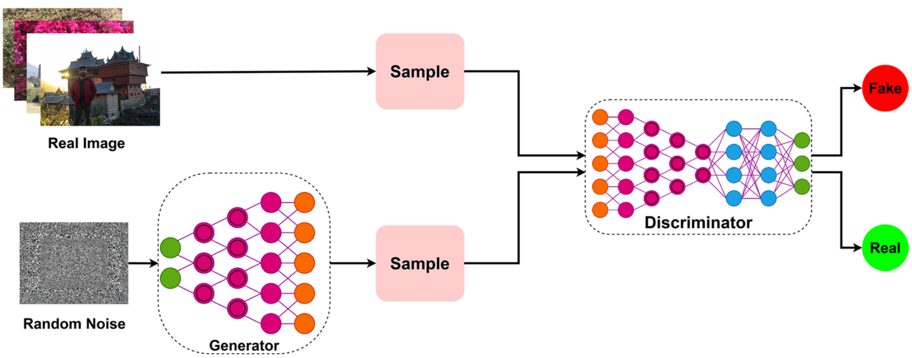


Fig. 7 Illustrating the architecture of a GANs, consisting of a generator network that generates new samples and a discriminator network that evaluates their authenticity

are neural networks, with the generator's output directly connected to the discriminator's input. The discriminator's classifications provide feedback to the generator through back-propagation, which the generator uses to adjust its weights.Advantages of GAN:

- High-dimensional data generation: GANs can generate high-dimensional data such as images, videos, and audio.
- Unsupervised learning: GANs can be trained in an unsupervised manner, which means that the model does not require labelled data to learn.
- Flexibility: GANs can be used for a wide range of applications, such as image generation, super-resolution, and text-to-speech synthesis.
- Handling mode collapse: GANs can handle mode collapse, which is a common problem in generative models where the generator produces only a limited number of variations of the data.
- Conditional GANs: by using conditional GANs, the generator can generate specific types of data based on the given conditions.

Disadvantages of GAN:

- Training instability: GANs can be difficult to train due to the instability of the training process, which can lead to poor performance or complete failure.
- Mode collapse: GANs can suffer from mode collapse, which is a common problem in generative models where the generator produces only a limited number of variations of the data.
- Requires a large amount of data: GANs require a large amount of data to train effectively.
- Requires powerful computing resources: GANs require powerful computing resources to train due to the complexity of the model.
- The complexity of the architecture: GANs have complex architecture which is difficult to understand and interpret the results.

Table 7 provides information about the work done by various researchers using GAN.

6.3.5 AE

AE are a type of unsupervised ANN that aim to learn an efficient way to compress and encode data, and then convert it back to a form that is similar to the original. The key feature of an AE is its encoder–decoder architecture, which is used to train a representation code. This representation code is typically smaller than the input code and can be considered a compressed version of the original data, useful for other data mining tasks. AE architectures typically have three layers: an input layer, a hidden layer, and an output layer. The input layer is where the raw data is fed in, the hidden layer is where the compression and encoding occur, and the output layer is where the data is reconstructed to be similar to the original input. The output and input layers of an AE have the same number of neurons. An example of this can be seen when providing an image of a handwritten number to the AE, it first compresses the image into a lower resolution and then tries to determine the hidden representation to reconstruct the original image. AE use an unsupervised learning algorithm that continues to train itself by setting the target output to match the input, this

Table 7 Usage of GANs in various research

Author(s)	Work done	Model
Yeo et al. (2021)	Improves the performance of GANs	GAN
Xia et al. (2022b)	Done a deep survey on GAN inversion	GAN
Emami et al. (2020)	Introduced the attention mechanism for the image-to-image translation	GAN
Souibgui and Kessentini (2020)	Proposed a DL-based model to enhance the viewing quality of documents	GAN
Tran et al. (2021)	Properly manage the distributions of augmented data	GAN
Niu et al. (2020)	Improved the power of the DL-based model for the detection of defectivity images	GAN
Zheng et al. (2020)	Proposed a key secret-sharing technology to tackle the problem of less security, tough recovery of lost keys, and less communication efficiency in the blockchain	GAN
Chen et al. (2021b)	Introduced a robust face detection method	GAN, Xception
Qu et al. (2020)	Proposed a novel methodology for obtaining differential privacy during data sharing in cyber-physical social systems	GAN
Albahrani et al. (2020)	Proposed a model for high-temperature modelling in industries	GAN
Li et al. (2021b)	Suggested a GAN model referred to as Hausdorff GAN, designed to resolve intrinsic dimension problems and produce top-notch data	GAN
Hou et al. (2020)	Proposed characterization and modeling of power loss in GaN-based HardSwitching half-bridges with dynamic ON-state resistance in mind	GAN
Prakash and Karam (2021)	Proposed object detection using GANs on images of varying quality	GAN
Shu et al. (2021)	A new MS-CartoonGAN was suggested, which transforms photos into various cartoon styles using a hierarchical semantic loss, an edge-promoting adversarial loss, and a style loss to keep semantic content and create high-quality cartoon images	GAN
Zheng et al. (2021)	Presented a novel GAN-based approach, SymReg-GAN, for symmetric image registration. Unlike existing methods that face limitations in speed, inter-modality relations, or data labelling, SymReg-GAN uses a semi-supervised strategy to exploit both labelled and unlabeled data for efficient and precise registration	GAN
Zhang et al. (2021b)	The newly suggested GAN-FM approach for merging infrared and visible images maintains all important information and surpasses current leading methods in keeping clear contrast and vivid textures	GAN

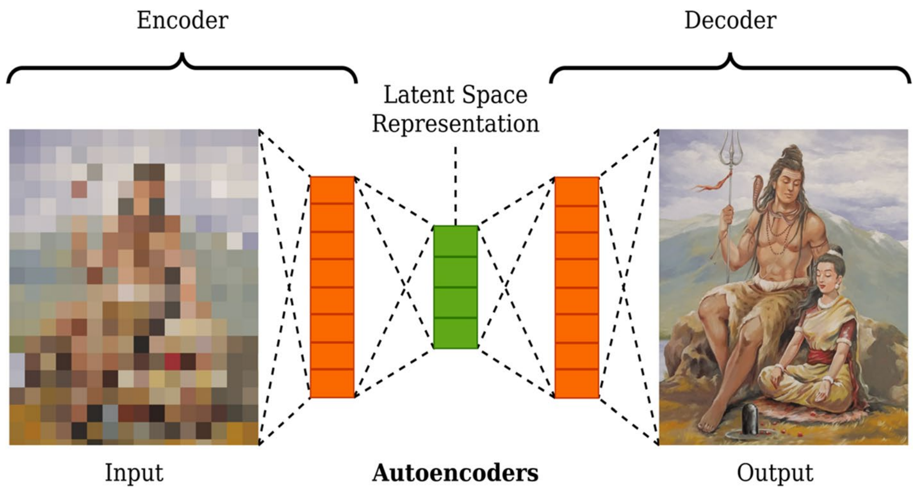


Fig. 8 Autoencoder architecture, consisting of an encoder network that compresses the input data and a decoder network that reconstructs it back to its original form

forces the small coding layer to reduce the dimensionality and reconstruct the input. The design of the AE is depicted in Fig. 8. Here are the terms utilized in the architecture of AE:

- Encoder (Wang et al. 2016): the encoder component in a neural network uses both convolutional and pooling layers to condense and convert the input information into a simplified form. The resulting encoded information is then sent to the bottleneck layer for additional handling.
- Latent Space Representation (Yu and Principe 2019): Latent Space is the space where the data lies after it has been compressed and encoded by the encoder in the bottleneck. The bottleneck is an important module in the model that compresses the input data to a lower dimension while retaining the most important features. The decoder then uses this compressed, encoded data to reconstruct the original input. The main purpose of the bottleneck is to extract useful features from the input data and discard less important information, making the data more compact and easier to process.
- Decoder (Irsoy and Alpaydın 2017): The decoder module is responsible for converting the compressed, encoded data from the bottleneck back to its original representation. It typically consists of multiple layers of transposed convolutional layers, also known as "deconvolutional" layers, which help to effectively "up sample" the data and restore the lost information. The goal of the decoder is to generate an output that is as similar as possible to the original input. In the case of image reconstruction, the decoder aims to rebuild the lost original image from the compressed data.
- The Reconstruction Loss (Wang et al. 2019): Reconstruction loss in an AE is a measure of how well the AE can reconstruct its input. It is typically calculated as the difference between the original input and the output of the decoder portion of the AE, and I used it as a training objective to minimize during training. Common choices of reconstruction loss include mean squared error and cross-entropy.

There are mainly seven types of AE, and their usage is mentioned in Table 8.

The advantages of Autoencoder are:

Table 8 Types of AE

Type of Autoencoder	Features
Denoising Autoencoder (Vincent 2011)	It generally reduces the loss function between the output and the aged node
Sparse Autoencoder (Xu et al. 2015)	It takes very high-performance values in the hidden layer and removes zero from all other hidden nodes
Deep Autoencoder (Hong et al. 2015)	It is majorly used for real-valued datasets
Contractive Autoencoder (Lv et al. 2018)	It is used to learn how to enter input in a small area for results
Under complete Autoencoder (Thies and Alimohammad 2019)	This does not need to be redone as it increases the chances of data rather than copying the output input
Convolutional Autoencoder (Chen et al. 2017)	It can remove audio from an image or re-create missing parts
Variational Autoencoder (Xie et al. 2020b)	It gives us a significant control, i.e. how we want to model our hidden distribution which other AE can't do

- Dimensionality reduction: AE can be used to reduce the dimensionality of high-dimensional data while preserving the essential features and relationships.
- Data compression: AE can be used as a lossy data compression method as they can learn to reconstruct data with a lower number of features, thus making the data more compact.
- Data denoising: AE can be trained to remove noise from data by learning to reconstruct the original input from a noisy version of it.
- Data generation: AE can be used for generating new samples of data by sampling in the latent space and decoding it back to the original space.

AEs have several disadvantages:

- They may suffer from overfitting, particularly if the dataset is small or the encoder and decoder architectures are too complex.
- AEs may struggle to learn useful features from the data if the encoder and decoder architectures are not appropriately designed.
- AEs are computationally intensive, especially for large datasets or high-dimensional data.
- AEs may fail to capture the underlying structure of the data, particularly if the data is highly structured or non-linear.
- AEs are not generally considered as good as GANs and VAEs for generating new, previously unseen samples from the data distribution.
- AEs are not designed for adversarial scenarios, where the goal is to generate samples that are difficult to distinguish from real data.
- AEs are not suitable for tasks where the data is highly irregular or unstructured.

Table 9 provides information about the work done by various researchers using AE.

Table 9 Usage of Autoencoder in various types of research

Author(s)	Work done	Model
Elkholy et al. (2020)	Developed Deep Convolutional Autoencoder-powered hyperspectral unmixing network	Deep Convolutional Autoencoder network
Yan et al. (2021)	Representing and understanding building shapes on maps using Graph Convolutional Autoencoder (GCAE)	GCAE
Zeng et al. (2018)	Suggested DL-based facial expression recognition with a sparse autoencoder	Sparse AE
Li et al. (2016)	Suggested DL-based remote sensing image classification with stacked AE	Stacked AE
Liu et al. (2018)	Diagnosing rolling bearing faults using Autoencoder-based RNN	Autoencoder
Lopez-Alvis et al. (2021)	Examines the application of Deep Generative Models (DGMs) in resolving inverse problems in geophysical imaging. DGMs enforce structured spatial patterns in subsurface solutions, backed by additional data like geological context	DGM, VAE
Lore et al. (2017)	Suggested a Deep Autoencoder method for enhancing natural low-light images	Deep AE
Lu et al. (2017)	A technique is proposed for identifying the health state of rotary machinery components by utilizing Stacked Denoising Autoencoder (SDA)-based methods to diagnose faults	SDA
Meng et al. (2018)	Enhancement of rolling bearing fault diagnosis using a Denoising Autoencoder (DAE)	DAE
Othman et al. (2016)	Classifying land-use scenes using Convolutional Features and a Sparse Autoencoder	Sparse Autoencoder
Soui et al. (2020)	Predicting bankruptcy using Stacked AE	Stacked AE
Sun et al. (2021)	Constructed multiscale DAE for enhancing target detection	DAE
Xu et al. (2017)	Assessing the quality of building footprint data	Deep autoencoder
Zabalza et al. (2016)	Proposed a new Segmented Stacked Autoencoder (SSAE) for efficient dimensionality reduction and feature extraction in hyperspectral imaging	SSAE
Zhao et al. (2020)	Classifying HSI using a combination of SSAE and Random Forest	SSAE

6.3.6 VAE

A VAE is a DL model consisting of two main components: an encoder and a decoder. The encoder of a VAE compresses the input data into a compact representation referred to as the latent space, and the decoder recreates the original data from this compressed representation. VAEs are commonly utilized in unsupervised learning and have the capability to generate new images by sampling from the latent space (Kingma and Welling 2019). The optimization of the VAE model is achieved by balancing the reconstruction error and the divergence between the encoded representation and a target distribution, which is measured using the Kullback–Leibler divergence (Joyce 2011). The continuous nature of the latent space in VAEs allows for seamless random sampling and interpolation. Unlike traditional input representations as fixed vectors, VAEs represent inputs as probability distributions utilizing two carriers: one carrying the meaning of the distribution and the other indicating the general deviation of the distribution. The VAE model is trained to minimize both the reconstruction error and the Kullback–Leibler divergence. VAEs have a continuous latent space, which enables easy random sampling and interpolation, making them effective in generating new images.

An Autoencoder transforms an input into a compact vector representation by minimizing the reconstruction loss between the input and the reconstructed image. A VAE, in contrast, generates its output by minimizing both the reconstruction loss and the KL Divergence loss, which measures the difference between the actual and observed probability distributions. This KL Divergence is a symmetrical score and distance measure between two probability distributions, and it ensures that the distribution learned by the VAE is not far from a normal distribution.

$$D_{KL}(P \parallel Q) = \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)} \tag{12}$$

The architecture of VAE is depicted in Fig. 9, When input data X is fed through the encoder, it outputs the latent state distributions (mean μ and variance σ), from which a vector Z is sampled. The assumption is made that the latent distribution is always Gaussian. The encoder compresses the input X into a smaller dimension, known as the bottleneck or latent space. Random data is sampled from this space, which is then decoded by the decoder by backpropagating the reconstruction loss to generate a new output.

VAE has several advantages, including:

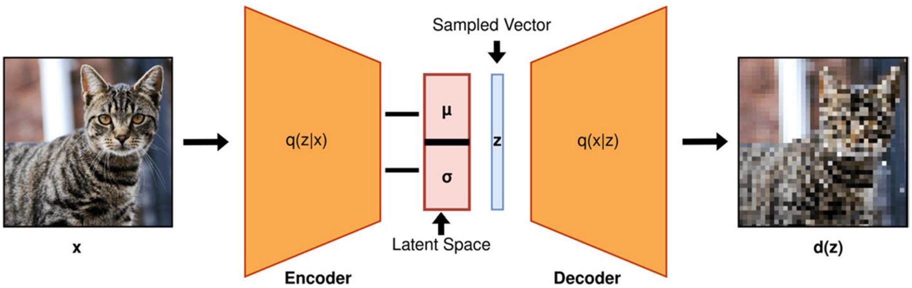


Fig. 9 VAE architecture, consisting of an encoder network that maps the input to a probabilistic latent space, a decoder network that reconstructs the output, and a sampling layer that generates latent variables

- Generating new data: VAEs can generate new data samples that are similar to the training data, which can be useful in a variety of applications, such as image synthesis, text generation, and drug discovery.
- Compact representation: VAEs can learn a compact representation of the data, called the latent code, which can be used for tasks such as data compression and dimensionality reduction.
- Handling missing data: VAEs can handle missing data by learning a probabilistic distribution over the data, which allows them to impute missing values.
- Outlier detection: VAEs can be used for outlier detection by identifying data samples that are unlikely under the learned probabilistic model.
- Anomaly detection: VAEs can be used for anomaly detection by identifying data samples that have a low likelihood under the learned probabilistic model.
- Flexible modelling: VAEs can be used to model a wide variety of data types, including continuous, discrete, and mixed data.

The disadvantage of VAEs is that they can be difficult to train, as the optimization objective is a lower bound on the likelihood of the data. Additionally, VAEs can produce blurry reconstructions and samples, since the model is trained to approximate a continuous latent space, rather than producing discrete outputs. Another disadvantage of VAEs is that they are not able to handle discrete data, such as text or categorical data, as well as other generative models such as GANs or autoregressive models like PixelCNN. Table 10 provides information about the work done by various researchers using VAEs.

6.4 DL framework

The enhancement in the use of DL architecture in various areas has been supported by the introduction of several DL frameworks in recent years. Mostly used frameworks are:

- Tensor Flow (https://www.tensorflow.org): TensorFlow is an open-source platform for building and deploying ML models, particularly DL models. It was developed by the Google Brain team in 2015 and supports multiple programming languages including Python, C++, and R. TensorFlow's flexible architecture allows for easy deployment on a variety of devices, including CPUs, GPUs, and TPUs
- PyTorch (https://pytorch.org): PyTorch is an open-source ML library developed by Facebook's AI Research lab in 2016. It is written in Python, C++, and CUDA and provides strong GPU acceleration for tensor computation. PyTorch is primarily used for creating deep neural networks and supports various DL models. It is known for its simplicity and ease of use, making it popular among researchers and practitioners.
- Deeplearning4j (https://deeplearning4j.konduit.ai): is an open-source DL framework developed in 2019 that is written in Java for the Java Virtual Machine (JVM) also known as DeepJava Library (DJL). It is the only DL framework that relies on the widely used programming language Java. DJL is useful for Java programmers as it allows them to train DL models in a familiar programming environment. DJL aims to provide a simple, high-level API and support for a wide range of DL models and platforms.

Table 10 Usage of VAEs in various types of research

Author(s)	Work done	Model
Li et al. (2020e)	A Sequential Variational Autoencoder (SISVAE) model was suggested for accurately estimating and detecting anomalies in multi-dimensional time series data. The model incorporates VAE and uses a RNN to uncover hidden temporal patterns	VAE
Yi et al. (2021)	A cross-modal variational autoencoder (CMVAE) has been suggested for recommending background music in micro-videos based on their content	VAE
Wang et al. (2020b)	A novel generative model called CVA2E is developed for HSI classification	VAE, GAN
Chadebec et al. (2022)	Proposed a geometry-based VAE for data augmentation in high-dimensional, low-sample size settings	VAE
Qiang et al. (2020)	Presented the Deep Variational Autoencoder (DVAE), a novel generative model for mapping functional brain networks in neuroimaging and brain mapping research. Addresses challenges of limited data and incomplete supervision in DL techniques applied to fMRI data	DVAE
Su et al. (2021)	Presented DOMI, a novel unsupervised method for detecting outliers in large datacenters using machine data	Gaussian mixture VAE
Chen et al. (2020b)	Presented a semisupervised classification method using an alternate discriminative mixture VAE	VAE
Ruan and Ling (2021)	An emotion-controlled conditional VAE was proposed to generate conversational responses with emotional content	VAE
Dewangan and Maurya (2021)	Presented an intelligent fault diagnosis method using a deep convolutional variable-beta VAE	CNN, VAE
Fan et al. (2021)	The Heterogeneous Hypergraph Variational Autoencoder (HeteHG-VAE) was presented, a novel link prediction technique for heterogeneous information net-works (HINs). The method transformed HIN into a heterogeneous hypergraph, uncovering high-level meaning and intricate connections while maintaining simple two-node relationships	HeteHG-VAE
Ye and Bors (2021)	Proposed a lifelong learning mixture of expert models using VAEs as experts. Trained by maximizing the sum of individual component evidence lower bounds on training sample log-likelihood	VAE
Khazeiynasab et al. (2021)	A new DL algorithm for calibrating generator parameters in power plants utilizing synchrophasor measurements has been proposed	Conditional variational autoencoders (CVAE)
Jiao et al. (2020)	Predicts the RUL of lithium-ion batteries using a particle filter and a CVAE with a reweighting strategy to enhance precision	CVAE
Wang et al. (2022)	Presented a loop closure detection technique for Autonomous Underwater Vehicles (AUVs) utilizing a VAE network. This approach addresses the difficulties posed by dynamic underwater surroundings, such as fluctuations in viewpoint, textureless imagery, and rapid objects	VAE

- The Microsoft cognitive toolkit (https://learn.microsoft.com/en-us/cognitive-toolkit/): Microsoft's Cognitive Toolkit (CNTK), which is a DL framework that allows for the training and execution of neural networks. It is open-source and written in C++, and it uses directed graphs to define neural networks as a series of computational steps. It was first released in 2019.
- Keras (https://keras.io): Keras is a high-level neural networks API, written in Python that is capable of running on top of TensorFlow 2.0. Keras is a popular choice for creating DL models because it offers a simple and user-friendly interface, making it easier for developers to quickly prototype and build models. It follows best practices for reducing cognitive load by providing consistent and simple APIs and minimizing the number of actions required for common use cases. Keras also offers a wide range of pre-built, commonly-used layers and models, allowing developers to focus on building their models rather than implementing low-level functionality.
- Open Neural Network Exchange (ONNX) (https://onnx.ai): ONNX is an open-source format for representing DL models. It was developed to enable AI developers to easily transfer models between different frameworks, by providing a common set of operators and data types. ONNX uses a computation graph model to describe the structure of a DL model, with nodes in the graph representing the various mathematical operations, and edges representing the data flowing between them. ONNX is designed to be extensible, allowing for the addition of new operators and data types as needed. This allows for greater flexibility and interoperability across different frameworks and tools, making it easier for developers to use their preferred framework while still being able to share and use models created in other frameworks.
- Mxnet (https://mxnet.apache.org/versions/1.9.1/): MXNet is an open-source DL framework that was developed by the Apache Software Foundation. It was first released in 2015. MXNet is designed to be fast, flexible and efficient, making it well-suited for both research prototyping and production use. It supports multiple programming languages, including C++, Python, R, Java, Julia, JavaScript, Scala, Go, and Perl. MXNet is known for its ability to scale to multiple GPUs and distributed systems, making it well-suited for large-scale, computationally intensive DL tasks. Additionally, it provides a high-level, easy-to-use API for specifying neural network models, allowing developers to quickly prototype and experiment with new models.
- Caffe (https://caffe.berkeleyvision.org): Caffe was developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors in 2013. It is a DL framework that is written in C++ with a Python interface. It is known for its speed and its ability to process images and videos quickly, making it a popular choice for computer vision tasks. Caffe offers an expressive architecture and extensible code, which allows developers to easily customize and experiment with new models. It has been widely used in industry and academia for image classification, feature extraction and many other DL tasks.

7 IoT

The IoT is a network of connected devices, such as sensors and cameras, that gather and share data through the Internet. The term "Internet" in IoT refers to the global network that facilitates connection and data sharing between users, while "Things" refers to the devices that collect and transmit information in various forms. IoT allows these devices to communicate with each other, and as of 2021, there are an estimated 9 billion connected devices worldwide. The use of sensors and connected devices allows for data collection and sharing in real time. IoT relies not only on hardware but also on software, which plays a crucial role in connecting and controlling these devices. As a result, users can remotely manipulate the behaviour of connected devices without the need for direct contact.

The IoT is important because it has the potential to revolutionize the way we live, work, and interact with the world around us. The IoT is a rapidly growing field that has the potential to greatly impact our daily lives. By connecting everyday objects and devices to the internet, IoT can improve efficiency, increase safety, and provide new opportunities for businesses. Through automation and real-time data analysis, IoT has the potential to increase productivity and make processes more efficient. Additionally, IoT can enhance customer experiences through personalized services and real-time information. By providing data-driven insights, IoT can improve decision-making and lead to better outcomes. In terms of safety, IoT-connected devices can provide real-time monitoring and help prevent accidents or other harmful events (Li et al. 2015). On a larger scale, IoT can promote sustainability by reducing waste and promoting the efficient use of resources. Finally, IoT is also opening up new business opportunities by creating new products and services. Overall, IoT represents a significant shift in how we interact with technology and the world around us.

7.1 Characteristics of IoT

IoT has been rapidly expanding and is expected to continue growing in the coming years. IoT devices such as smart speakers, smart home systems, and wearable devices have become more common and have made it easier for people to control and automate their homes and devices. Some of the key characteristics of IoT include:

- Intelligence: the IoT network is made up of a combination of hardware devices, algorithms, and software. These components work together to collect and transmit data, allowing for the automation and remote control of devices. Ambient intelligence is a concept that refers to the ability of IoT networks to respond intelligently to different situations.
- Connectivity: having a reliable connection between devices on a network is important for ensuring a high rate of data transfer. This can be achieved through a variety of methods, such as using wired connections, implementing network redundancy, and using network protocols that are designed for efficient data transfer. Additionally, maintaining strong signal strength and minimizing interference can also help to improve the reliability of a network connection.
- Efficient: efficiency is a critical aspect of IoT networks. In terms of power consumption, it is important to use low-power devices and protocols to minimize the amount of energy required for communication. Additionally, using sleep modes and other power-saving techniques can help to further reduce energy consumption. In terms of data generation, it is important to use sensors and devices that are designed to generate minimal amounts of data while still providing accurate information. Additionally, implementing data compression and aggregation techniques can help to minimise the quantity of data that needs to be transmitted. Ensuring that the sensor works efficiently is also important. This can be achieved by using high-quality sensors that are designed for specific applications, and by regularly maintaining and calibrating the sensors to ensure their

optimal performance. Overall, designing IoT networks with efficiency in mind can help to reduce power consumption, minimize data generation, and ensure that the sensors are working properly.

- Scalable: IoT networks are designed to be scalable, allowing users to easily add or remove devices as their needs change. This scalability allows for flexibility in terms of the number of devices that can be connected to the network and makes it easy to adapt the network as new devices or technologies become available. Additionally, it allows users to easily expand their network as their needs change, whether that means adding more devices to increase capacity or removing devices to reduce costs.
- The abundance of sleeping nodes: using sleep modes on sensors can be an effective way to reduce power consumption in an IoT network. When a sensor is not actively being used, it can be placed in sleep mode to conserve energy. This can be especially useful in situations where certain sensors may only be needed occasionally, such as environmental sensors that are used to monitor air quality or temperature. Implementing sleep modes also helps to extend the battery life of the sensors, which can be beneficial for sensors that are placed in remote or hard-to-reach locations. Additionally, it can also help to reduce the amount of data that needs to be transmitted, by only sending data when a sensor wakes up from sleep mode. Overall, using sleep modes on sensors can be an effective way to conserve energy and extend battery life in an IoT network, while still providing the necessary data when required.
- Smart sensing: sensors play a crucial role in providing smart sensing functionalities in IoT networks. For example, infrared (IR) sensors can be used to identify the existence of objects or materials in a designated region and provide information about the location and size of the object (Gorostiza et al. 2011). This can be useful in a wide range of applications, such as security systems, industrial automation, and robotics. Motion sensors, on the other hand, can be used to detect the presence of people or objects in a specific area, and can be used to trigger different actions based on the detected motion. Additionally, other types of sensors can provide different smart sensing functionalities, like temperature sensors, humidity sensors, pressure sensors, and many more. These can be used in different industries like agriculture, healthcare, transportation and many more. Overall, sensors are an essential part of IoT networks, providing smart sensing functionalities that can be used to automate different processes, improve security, and make systems more efficient.
- Dynamic nature: one of the main functions of IoT is to gather and transmit data from its environment through the use of various devices. These devices are crafted to detect and gather information regarding the condition and circumstances of their environment, such as temperature, location, speed, and more. The state of these devices can change dynamically, depending on various factors such as the device's power usage, connection status, and surrounding conditions. For example, a device may be in sleep mode to conserve power but will wake up and start collecting data when it detects a change in its environment. Similarly, a device may connect or disconnect from a network depending on its location and proximity to other devices. The number of devices in a given environment can also change dynamically, depending on the person, place, and time. For example, a person may have a different set of devices in their home than at work, or the number of devices in a given location may change depending on the time of day or the presence of certain individuals. Overall, the ability to gather and transmit data from the environment through the use of dynamic and adaptive devices is a key feature of IoT and enables a wide range of applications and use cases.

7.2 Connectivity terminologies

IoT connectivity technologies are the foundation for connecting IoT devices and facilitating communication within an IoT network. IoT connectivity terminologies are classified into five categories that are IoT LAN (Hashida et al. 2019), IoT WAN (Sanchez-Iborra and Cano 2016), IoT Node (Cerchecci et al. 2018), IoT Gateway (Kang and Choo 2018), and IoT Proxy (Jin and Kim 2018) as shown in Fig. 10. These building blocks include IoT LAN for short-range communication, IoT WAN for wide-area communication, IoT Node for connecting different nodes in a LAN, IoT Gateway for transferring data between IoT LAN and IoT WAN, and IoT Proxy for active application layer functions between IoT nodes.

7.3 IoT components

IoT components are the major key elements of any IoT network. IoT consists of both hardware and software elements. The hardware elements consist of sensors, actuators, and devices that gather, handle, and transmit information. The Software components include the IoT platforms, middleware, and applications that provide the ability to manage, analyse, and act on the data collected by the Hardware components. Together, these components make up the infrastructure of an IoT network, enabling communication and data exchange between devices and systems. They are accountable for complete IoT system operation, from data acquisition to evaluation and response. The following are the IoT components:

- Devices: devices are a key hardware component of IoT networks and play a crucial role in data gathering and transferring. Devices can be a wide range of hardware such as routers, sensors, computers, towers, and many other types of equipment. They gather and quantify information from the physical environment and transmit it to the IoT system for further analysis and interpretation (Meneghello et al. 2019). The type of device used depends on the specific application and the type of data being collected. For example, a temperature sensor would be used to collect temperature data, while a camera would be used to collect image data. These devices are connected to the IoT network and communicate with the other components of the system to provide real-time data and insights.

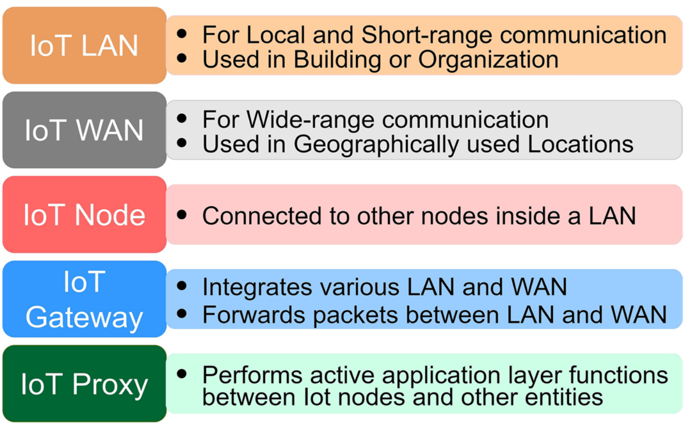


Fig. 10 IoT connectivity terminologies

- Local network: the network is a key component of IoT systems, connecting all the different nodes or hardware components. The network enables communication and data exchange between devices and systems. It can be either wired or wireless and can range from a simple local area network (LAN) to a wide-area network (WAN) spanning multiple locations (Stiller et al. 2020). The network infrastructure is responsible for connecting all the devices and allowing them to communicate with each other and with the IoT platform. It also provides the necessary security, reliability, and scalability to support a large number of devices and the large amounts of data that are generated in an IoT system. The network infrastructure includes various technologies such as Zigbee, Z-wave, LoRa, Sigfox, and many more, which help the data communication between the devices and the Gateway, which further sends the data to the cloud or the data centre for further processing and analysis.
- Internet: The Internet plays a crucial role in connecting and sharing information between various nodes in an IoT network. The internet allows the devices and systems in an IoT network to communicate with each other and with other systems and services beyond the local network. This enables the data collected by the devices to be transmitted to the IoT platform, where it can be analyzed and acted upon. The internet also allows users to access the data and insights generated by an IoT system, regardless of their location. Internet connectivity also helps in remote monitoring and controlling of the devices and systems, which makes the IoT system more efficient and cost-effective. The internet also enables the integration of various IoT systems and services, allowing them to share data and resources and creating new opportunities for innovation. It is important to note that, for secure and reliable communication, the devices and systems need to be connected to the internet via secure protocols and have robust security measures in place to protect against cyber threats.
- Backend services: backend services act as a bridge between the user and the hardware devices in an IoT system. These services are responsible for processing and analysing the data collected by the devices, and for providing the necessary functionality to manage and control the devices (Saidi et al. 2022). The backend services are typically hosted on cloud servers or data centres and can include various components such as:

  - IoT platform: this is the core component of the backend services and provides the ability to manage and control the devices, collect and process the data, and provide APIs and SDKs for developers to build applications on top of the platform.
  - Middleware: this component sits between the IoT platform and the devices, and provides the necessary functionality to connect, manage, and control the devices.
  - Analytics and Business Intelligence (BI) tools: these tools are used to analyse the data collected by the devices and provide insights and intelligence to the users.
  - Database management systems: this component is responsible for storing and managing the data collected by the devices.
  - Security and Identity Management: this component is responsible for ensuring the security and protection of the data and devices, and for managing the identities of the users and devices.

  The backend services work together to provide a seamless experience for the user, by responding to the outcomes generated from the hardware components, making it easy for the user to interact with the system, monitor it, and make decisions based on the data generated by the system.

- Applications: applications play a vital role in IoT networks, providing users with a user friendly interface to interact with the ongoing process of the network. Applications can be built for a wide range of devices and platforms, such as smartphones, tablets, laptops, and web browsers, and can be used to access and control devices, view and analyse data, and receive alerts and notifications. The applications can be categorized into two types:
  - Web-based applications: these are web-based applications that can be accessed through a web browser. They provide users with the ability to access and control the devices, view and analyse the data, and receive alerts and notifications, from anywhere with an internet connection.
  - Native applications: these are applications that are developed specifically for a particular device or platform, such as iOS or Android. They can be downloaded and installed on the device and provide users with a more seamless and responsive experience.

The applications can be developed using a wide range of programming languages and frameworks and can be integrated with various IoT platforms and services. They can be also integrated with third-party services such as social media, messaging, and location services to provide more functionality and value to the users. In summary, Applications are an important part of IoT systems, providing users with an easy way to interact with the network and access the information generated by the system, making it more user-friendly and efficient.

7.4 IoT service-oriented architecture

Service-Oriented Architecture (SOA) refers to a design methodology for software architecture that uses service calls within the layers of architecture to develop applications. SOA has two major roles: providing services and consuming services. As a service provider, SOA acts as a maintainer of the service and the organization that makes one or more services available for others to use. As a service consumer, SOA locates the metadata of a service in a registry and creates the necessary user components to bind and use the service (Mishra and Sarkar 2022). In IoT systems, SOA plays a vital role in providing a flexible and scalable infrastructure for developing IoT applications. It allows different layers of the system to communicate and interact with each other through well-defined interfaces and protocols. The layers involved in IoT-based service-oriented architecture include:

- Sensing Layer: this layer is responsible for sensing the data and sending it to the network layer for further processing.
- Network Layer: this layer is responsible for processing the data and transmitting it to the service layer.
- Service Layer: this layer deals with service delivery in the form of a repository, service division, and integration to support the functionality of the IoT system.
- Interface Layer: this layer provides the interface for interacting with the system.
- Security Layer: this layer is spread across all the layers to provide secure data communication and protect the system from cyber threat

In summary, SOA plays a crucial role in IoT systems by providing a flexible and scalable architecture for developing IoT applications, allowing different layers to communicate and interact with each other through well-defined interfaces and protocols, and also providing security to the system. Figure 11 shows the schematic diagram of IoT service-oriented architecture.

7.5 IoT layers

The different layers of the IoT play a critical role in enabling the seamless flow of data between physical devices and end-users. Each layer provides unique functions that are necessary for the overall functioning of the IoT system. In this section, Fig. 12 depicts the layered design of the IoT and a brief about each layer is described as follows:

- Perception Layer (Zhang et al. 2017b): the perception layer in IoT refers to the physical devices and components that collect and transmit data within the network. Examples of hardware in this layer include sensors, routers, and other devices that gather and transmit information. This layer serves as the foundation for the rest of the IoT system by providing the raw data that is used to make decisions and perform actions.
- Connectivity Layer (Ullah et al. 2019): the connectivity layer in IoT refers to the technology and protocols used to establish and maintain connections between devices in the network. This layer is responsible for allowing devices to communicate with each other and with the outside world. Examples of connectivity technologies in this layer include WiFi, Bluetooth, Zigbee, and cellular networks. This layer is critical for the proper functioning of the IoT system, as it enables the flow of data between devices, and allows for remote monitoring and control of devices.

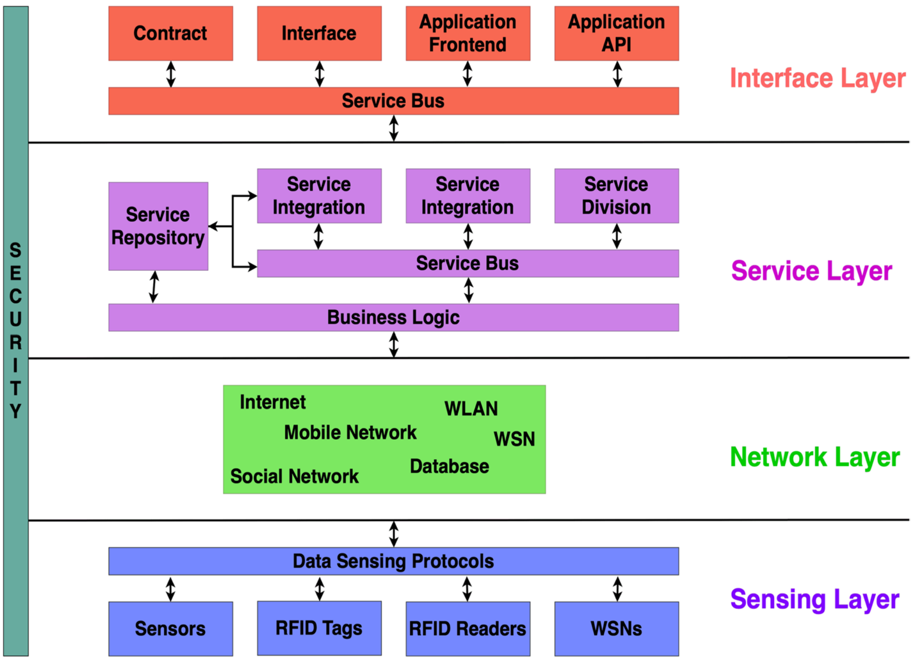


Fig. 11 IoT service-oriented architecture

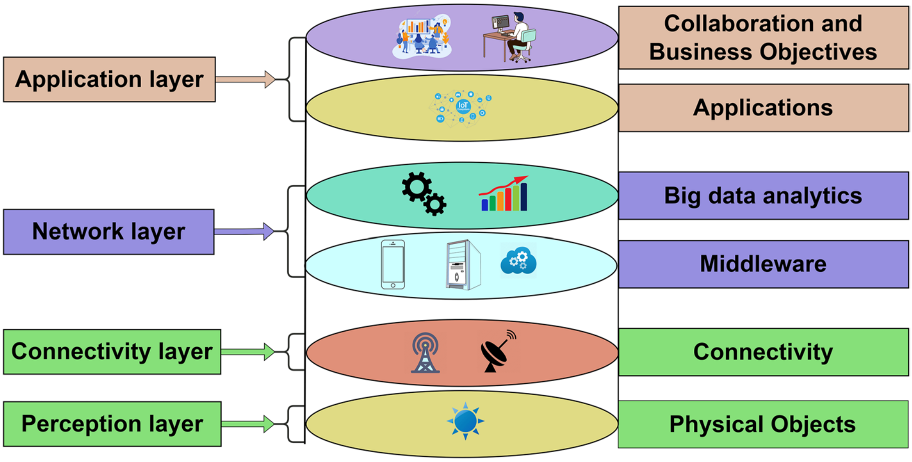


Fig. 12 Layers of IoT

- Network Layer (Bello et al. 2017): the network layer in IoT manages the delivery and transmission of information between devices. It facilitates inter-device communication and guarantees smooth data flow. This layer is responsible for the management of the network and its resources, such as IP addressing, routing tables, and data encryption. Examples of technologies in this layer include IPv6, 6LoWPAN, and MQTT. This layer plays a crucial role in the smooth operation of the IoT system as it facilitates inter-device communication, and network administration, and safeguards data security and consistency.
- Application Layer (Karagiannis et al. 2015): the application layer in IoT refers to the software and user interfaces that enable interaction with the network and data collection. This layer provides the framework for users to access and control the devices connected to the network, as well as to view and analyse the data. Examples of applications in this layer include smartphone apps, web interfaces, and dashboards that allow users to monitor and control devices remotely. This layer is important as it provides an easy-to-use interface for users to interact with the IoT system, and enables them to make use of the data collected by the network to make decisions and perform actions.

7.6 Communication technologies

Communication technologies in IoT play a crucial role in transmitting and receiving data between devices in the network. These technologies enable the flow of information between devices, allowing them to communicate with each other and with the outside world. Different communication technologies have different characteristics, such as range, power consumption, and data transfer rates, which make them suitable for different types of IoT applications. The choice of specific communication technology is influenced by factors such as the device and sensor types employed, the necessary reach and speed of data transfer, and the complete network desirous communication technologies are useful in IoT networks some of them are:

- Wi-Fi (Ma et al. 2019c): Wi-Fi, based on the IEEE 802.11 standard (https://www.ieee802.org/11/), uses radio frequency to transmit and receive data and operates on the 2.4 GHz frequency band. It is a crucial communication technology in IoT, with a transmitting range of up to 50 m, making it popular in IoT devices due to its high speeds and long range.
- Bluetooth (Bisdikian 2001): Bluetooth, based on the IEEE 802.15 standard (https://www.ieee802.org/15/), is low-power-consuming device that operates on the 2.4 GHz frequency band. It has a data transmission range of up to 10 m, with a transmitting power ranging from − 20 to + 20 dBm. It's widely used in IoT to connect devices such as smart speakers and wearables.
- ZigBee (Muthu Ramya et al. 2011): ZigBee is an open global standard wireless technology created to address the low-cost and low-power requirements of IoT networks. It is founded on the IEEE 802.15.4 standard (https://standards.ieee.org/ieee/802.15.4/7029/) and operates on the 2.4 GHz frequency band. Compared to other communication technologies, it is a low-cost and low-power device that ensures secure data transmission. The data transmission range of ZigBee ranges between 10 and 100 m.
- WiMAX (Andrews et al. 2007): Worldwide Interoperability for Microwave Access (WiMAX) is a long-distance communication technology that uses the IEEE 802.16e standard (https://www.ieee802.org/16/tge/), providing a network capacity of up to 80 Mbps.
- 6LoWPAN (Mulligan 2007): 6LoWPAN stands for "IPv6 over Low power Wireless Personal Area Networks," and it is a protocol that enables low-power and low-cost devices to communicate using Internet Protocol (IP), facilitating the establishment of IoT networks. The protocol is founded on the IEEE 802.15.4 standard that works on the 2.4 GHz frequency band.

7.7 IoT applications

The IoT is a term used to describe the network of connected physical objects such as home appliances, vehicles, and other devices. Equipped with sensors, electronics, and software, these objects can gather and exchange data, facilitating real-time interaction among themselves and with humans. IoT applications range from smart homes, industrial automation, and wearable technology, to healthcare and transportation. By enabling real-time communication and data exchange, IoT applications are transforming traditional industries and creating new opportunities for innovation and growth. The benefits of IoT applications include increased efficiency, improved decision-making, cost savings, enhanced safety and security, improved customer experience, predictive maintenance, and remote monitoring and control. IoT has a wide range of applications across various industries, including:

- Smart homes: IoT devices such as smart thermostats, lighting, and security systems can be used to automate and control various functions in a home, making it more energy-efficient and secure (Samuel 2016).
- Smart cities: IoT-enabled sensors and devices can be used to monitor and control various functions in a city, such as traffic flow, air and water quality, and waste management (Arasteh et al. 2016).
- Healthcare: IoT-enabled devices such as wearables and remote monitoring systems can be used to track and manage patients' health, enabling doctors to provide more personalized care (Selvaraj and Sundaravaradhan 2020).

- Agriculture: IoT-enabled sensors and devices can be used to monitor and control various aspects of farming, such as soil moisture, temperature, and crop growth (Thakur et al. 2020).
- Industrial automation: IoT-enabled sensors and devices can be used to monitor and control various aspects of industrial processes, such as equipment performance and energy consumption (Park and Park 2016).
- Retail: IoT-enabled devices such as sensors and RFID tags can be used to track inventory, monitor customer behaviour, and improve the shopping experience (Caro and Sadr 2019).
- Transport and Logistics: IoT-enabled devices such as GPS tracking and telematics can be used to track and monitor vehicles, and cargo and improve the efficiency of the supply chain (Sicari et al. 2019).

In this section, we will delve into some of the most significant and promising applications of IoT that are rapidly emerging and transforming various industries. These applications span multiple domains, including Healthcare, Agriculture, Smart Cities, and Smart Homes. Each of these applications leverages the unique capabilities of IoT devices, such as real-time data collection, connectivity, and automation, to deliver significant benefits such as increased efficiency, improved decision-making, cost savings, enhanced safety and security, improved customer experience, predictive maintenance, and remote monitoring and control. By exploring these emerging applications of IoT in-depth, we will gain a better understanding of the potential of this technology and how it is shaping our world.

7.7.1 Healthcare

IoT technology has revolutionized the healthcare industry by enabling real-time monitoring and data management of patient's health status. IoT devices such as sports watches, medical alert bracelets, and wearable sensors can collect a range of biometric data, including heart rate, blood pressure, oxygen levels, and more. This data is then analyzed and managed to provide doctors and healthcare professionals with real-time insights into patients' health status, enabling them to make informed decisions and provide prompt and effective treatment. Additionally, IoT-enabled remote monitoring and telemedicine solutions allow patients to receive medical care from the comfort of their homes, reducing the need for hospital visits and reducing healthcare costs. The impact of IoT on healthcare is only set to grow, and researchers are continuously exploring new and innovative ways to harness its potential. The study by Kim and Kim (2018) explores the user perspective of IoT healthcare applications and their usefulness in managing lifestyle diseases. The study by Baker et al. (2017) provides an in-depth analysis of the various technologies, challenges, and opportunities involved in integrating IoT into smart healthcare. Farahani et al. (2018) propose a fog-driven model for IoT eHealth, while Parthasarathy and Vivekanandan (2020) develop a framework for regular monitoring of arthritis using a time-wrapping algorithm based on an IoT-based architecture. These studies demonstrate the various ways in which IoT is transforming the healthcare industry and highlight the importance of continued research and innovation in this field. The IoT has the potential to revolutionize healthcare by improving patient outcomes, increasing efficiency, and reducing costs. Here are a few examples of how IoT is being used in healthcare:

- Remote monitoring: IoT-enabled devices such as wearables, sensors, and telemedicine equipment can be used to monitor patients' vital signs and health status remotely. This can enable doctors to provide more personalized care and reduce the need for hospital visits
- Medication management: IoT-enabled devices such as smart pill bottles and medication dispensers can be used to remind patients to take their medication, track usage, and alert caregivers if there are any issues.
- Clinical decision-making: IoT-enabled devices can be used to collect and transmit large amounts of data, which can then be analyzed to identify patterns and trends. This can help doctors make more informed decisions about patient care.
- Clinical trials and research: IoT-enabled devices can be used to collect and transmit data from patients participating in clinical trials and research studies, which can lead to new treatments and cures.
- Medical Equipment Management: IoT-enabled devices can be used to monitor the usage, performance and maintenance of medical equipment, which can help with preventive maintenance and reduce downtime.

These are just a few examples of the many ways in which IoT is being used in healthcare to improve patient outcomes, increase efficiency, and reduce costs. As the technology continues to evolve, new applications and use cases for IoT in healthcare will likely emerge.

7.7.2 Agriculture

IoT has always helped in solving various problems which require constant monitoring and controlling. Therefore, its applications in the field of agriculture are numerous. IoT-based sensors and devices help in reducing human effort and save time for many farmers around the world. IoT technology is being increasingly used in the agriculture industry to improve crop yields, reduce waste, and optimize resource utilization. IoT devices such as sensors, drones, and smart irrigation systems can collect real-time data on soil moisture, temperature, and other environmental factors, and use this information to optimize crop growth. IoT-enabled precision agriculture techniques, such as variable rate fertilization and planting, can improve crop yields and reduce waste by using data to determine the optimal amount of resources required for each area of a field. In addition, IoT devices can be used to monitor livestock, track and manage farm machinery, and improve food safety through traceability systems. These technologies are helping farmers to increase productivity, reduce costs, and improve the sustainability of their operations, making IoT a key driver of innovation and growth in the agriculture industry. Popović et al. (2017) conducted a study on an IoT-based platform for precision agriculture, examining various platforms involved. Tzounis et al. (2017) gave an extensive rundown of the function of IoT in agriculture and current advancements in IoT for agriculture. Lakhwani et al. (2019) reviewed multiple IoT applications that support precision agriculture. Muangprathub et al. (2019) put forward an automatic watering system using IoT and sensor networks to irrigate agricultural lands and reduce water waste, utilizing sensors and mobile applications. The benefits of using IoT in agriculture include:

- Increased crop yield and quality: IoT devices can collect data on soil moisture, temperature, light, and other important factors, which can then be used to optimize growing conditions and increase crop yield and quality.

- Efficient resource management: IoT devices can help farmers monitor and control water usage, fertilization, and other resources in real-time, reducing waste and increasing efficiency.
- Improved decision-making: with access to real-time data, farmers can make informed decisions about planting, harvesting, and other critical activities, leading to improved yields and lower costs.
- Predictive maintenance: IoT devices can detect and diagnose issues with machinery, alerting farmers to potential problems before they become critical.
- Enhanced food safety: IoT devices can be used to monitor food production, storage, and distribution, helping to ensure food safety and reduce waste.
- Improved animal monitoring: IoT devices can be used to monitor animal health, feed intake, and behaviour, helping to improve animal welfare and productivity.

7.7.3 Smart cities

The IoT technology constitutes of sensors, devices and applications which monitor, control and communicate with each other and use the data beneficially collected by them. To develop smart cities using IoT, various sensors, monitors, and smart devices are to be installed throughout the city and connect to achieve tracking, monitoring, controlling, and intelligent recognition. The recent growth in smart cities and IoT applications has opened doors for a lot of opportunities for researchers. Smart cities along with smart homes are the most prominent applications of IoT in today's time. The declined cost of sensors and various other equipment has also been a reason for the growth of smart cities. The growing population of the world results in many cities being overpopulated, many of them being in India. This calls for resource utilisation as power consumption in these cities is enormous. The benefits of using IoT in smart cities include:

- Improved traffic management: IoT devices can be used to monitor and control traffic flow, reducing congestion and improving safety.
- Increased energy efficiency: IoT devices can be used to monitor and control energy usage in buildings and infrastructure, reducing waste and saving resources.
- Enhanced public safety: IoT devices can be used to monitor public safety, including air and water quality, crime, and natural disasters, allowing for rapid response and improved outcomes.
- Improved waste management: IoT devices can be used to optimize waste collection and disposal, reducing waste and improving sustainability.
- Better public services: IoT devices can be used to improve public services, such as public transportation, healthcare, and education, by allowing for real-time monitoring and improved decision-making.
- Increased citizen engagement: IoT devices can be used to engage citizens in the management and development of their city, improving civic participation and community spirit.

7.7.4 Smart homes

The IoT is transforming the way we live in our homes. Smart homes use IoT devices to automate and control various systems, such as lighting, heating, cooling, security, and entertainment. With the ability to connect and communicate with these devices through a central hub, such as a smartphone or tablet, homeowners can easily control their home's environment and security from anywhere, at any time. IoT devices can also gather data on energy usage and consumption patterns, allowing homeowners to optimize their energy usage and reduce their carbon footprint. Additionally, IoT devices can help to enhance the overall comfort and convenience of the home, by allowing for customized settings and automating routine tasks, such as adjusting the temperature or turning off the lights when no one is home. The integration of IoT in smart homes is revolutionizing the way we live, work, and play, by providing greater comfort, security, and control over our daily lives. The benefits of IoT in smart homes include:

- Energy efficiency: automated control of devices, such as HVAC systems, lights, and appliances, leads to optimized energy consumption and reduced bills.
- Convenience: remote control of devices and monitoring through a smartphone or tablet offers greater convenience for homeowners.
- Improved safety: the use of sensors and monitoring systems can alert homeowners to potential safety hazards, such as gas leaks or fires.
- Better health: smart homes can monitor and control indoor air quality, ensuring a healthier living environment.
- Increased home value: the integration of IoT technology can increase the value of a home.
- Peace of mind: monitoring systems and remote access provide peace of mind when homeowners are away from their property.

8 Usage of DL in IoT applications and IoT devices

The IoT has rapidly emerged as a transformative technology that connects various physical objects, appliances, and devices to the internet, enabling them to exchange data and communicate with each other. This has resulted in an explosive growth of IoT devices and applications, ranging from smart homes and cities to industrial automation and healthcare. DL, with its ability to learn from large-scale data and make accurate predictions, has become a critical enabler for a wide range of IoT applications. In this section, we will discuss the various applications of DL in IoT devices and systems, including IoT-enabled sensors, edge computing, and network optimization.

8.1 DL for IoT applications

DL is a powerful tool for IoT applications, as it allows for the analysis of large amounts of data and can be used to make predictions and decisions based on that data. Some examples of DL for IoT applications include:

- Predictive maintenance: using DL algorithms to analyze sensor data from industrial equipment, such as machines in a factory, to predict when maintenance is needed.
- Anomaly detection: DL models can identify patterns in large amounts of IoT data and flag any data points that deviate from these patterns as anomalies.
- Time-series forecasting: LSTMs and other RNN models can be used to analyze time-series data generated by IoT devices and make predictions about future trends.

- Decision-making: reinforcement learning can be used to train models that make decisions based on IoT data, for example, in the case of an autonomous vehicle or industrial control system.
- Smart homes: using DL to analyze sensor data from smart home devices, such as thermostats and security cameras, to make decisions about heating, lighting, and security.
- Autonomous vehicles: using DL to process sensor data from cameras, LIDAR, and radar to make decisions about steering, braking, and accelerating.
- Surveillance: using DL algorithms to analyze video footage from surveillance cameras to detect suspicious behaviour or objects.
- Healthcare: utilizing IoT devices with DL algorithms to predict and prevent illnesses and monitor patients remotely.

Overall, the use of DL for IoT applications allows for the analysis of large amounts of data, making predictions and decisions based on that data, and the ability to automate many tasks. In this section, a comprehensive examination of the utilization of DL in various developing applications of IoT has been provided.

8.1.1 DL in agriculture

DL and IoT can be used in agriculture to improve crop yields, reduce costs, and increase efficiency. For example, using IoT sensors and cameras to collect data on soil moisture, temperature, and crop growth, and then using DL algorithms to analyze the data and predict future crop yields. Additionally, DL can be used to classify images of crops, detect pests and diseases, and even control irrigation systems. By automating these tasks, farmers can make more informed decisions, improve crop yields and reduce costs. Agriculture is a crucial sector for the economies of both developing and developed countries and the main source of income for over 50% of a country's population. Farmers aim to maximize their agricultural output each harvest season. However, factors such as environmental changes, intrusion attacks, plant diseases, and others can impact crop productivity. Integrating DL models is crucial in addressing these factors, just as biological substances and chemicals are. Detection and classification of plant diseases are key factors in reducing crop production and quality. To address this issue, researchers have proposed various DL-based solutions. The use of CNN based DL models in image detection and classification offers effective and efficient solutions due to its capability to condense the image and extract crucial information. There are various CNN architectures, such as AlexNet, GoogleNet Inception V3, VGG Net, and Microsoft ResNet, that can accurately detect and classify plant diseases. When deploying DL in agriculture, it is necessary to incorporate IoT components such as sensors, cameras, and UAVs to gather data, as manually collecting data in large agricultural fields in challenging terrain is not feasible. The collected data is then processed using DL. Table 11 presents a summary of the efforts made by various scientists in the field of agriculture utilizing DL.

8.1.2 DL in healthcare

DL and IoT can also be used together in healthcare to enhance the capabilities of medical diagnosis and treatment. IoT refers to the interconnections of physical devices, such as sensors and actuators, that collect and transmit data over a network. In healthcare, IoT devices can be used to gather data on a wide range of systems, such as vital signs, medical

Table 11 DL in agriculture

Author(s)	Work done	DL model
Mustafa et al. (2020)	Designed a computerized mixed method for identifying and diagnosing plant illnesses in herb plants	SVM
Jiang et al. (2020)	Identified four distinct ailments affecting rice plant leaves	CNN
Mishra et al. (2020)	Detects diseases in corn plant	CNN
Sujatha et al. (2021)	Compared the use of DL and ML techniques in identifying plant leaves	VGG16, InceptionV3, VGG19
Subetha et al. (2021)	Conducted comparative analysis of plant pathology classification using ResNet and VGG19 by performing classification and prediction	Resnet and VGG19
Geetharamani and Pandian (2019)	Developed deep CNN model for plant leaf disease identification	CNN
Karadağ et al. (2020)	Detects pepper fusarium disease	KNN, ANN
Jasim and Al-Tuwaijari (2020)	Conducted plant leaf disease detection and classification	CNN
Militante et al. (2019)	Detection and recognition of plant leaf diseases were carried out	CNN
Ashok et al. (2020)	Detection of tomato leaf diseases was performed	CNN
Gayathri et al. (2020)	Analysis and detection of tea leaf diseases were conducted	LeNet
Tiwari et al. (2020)	A DL approach for the detection of potato leaf diseases was employed	VGG19
Francis and Deisy (2019)	Detect and classifies diseases in plants	CNN
Iqbal and Talukder (2020)	Potato disease detection involves using image segmentation	Random Forest Classifier
Adedoja et al. (2019)	The NASNet model has been implemented for the recognition of plant diseases	NASNet

imaging, and patient-generated data. DL, on the other hand, can analyse large and complex datasets, such as those generated by IoT devices. By using DL algorithms, it is possible to extract insights and make predictions from the data collected by IoT devices. Together, DL and IoT can enable a wide range of applications in healthcare, such as:

- Medical imaging: analysing medical images such as X-rays, CT scans and MRIs to detect diseases and aid in diagnosis.
- Predictive analytics: analysing patient data to predict the likelihood of certain health outcomes and provide early warning for potential health issues.
- Telemedicine: monitoring patients remotely using IoT devices to gather data on vital signs and other health metrics.
- Personalized medicine: using DL algorithms to analyse patient data and create personalized treatment plans.

Overall, the combination of DL and IoT in healthcare can help to improve the efficiency, accuracy and effectiveness of medical diagnosis and treatment by providing insights from large and complex data sets. Health emergencies such as heart attack, cancer, mental illness, and various illnesses are significant worries in people's lives. Without adequate healthcare, society cannot progress. Just as medicine, physical activity, and yoga are crucial for good health, integrating DL into healthcare is crucial for better healthcare. DL has numerous applications in healthcare, including disease identification and diagnosis, medical imaging, disease forecasting, intelligent medical records, customized medical treatment, and more.

DL functions similarly to the human brain, creating new neurons and learning from fed information through algorithms and neural networks to produce more precise outcomes. In medical image analysis, DL is divided into classification, segmentation, detection, and others. In classification, images are categorized into the presence or absence of disease. Object classification can be used to identify specific parts of an image for more accurate classification. Detection is the following step in classification and a crucial step in segmentation, where important features such as organs can be extracted. Segmentation is then used for image processing of organs, such as brain scans. DL algorithms have been successful in classifying and predicting disease patterns and treatment options for patients. Table 12 summarizes the contributions of numerous researchers in the field of healthcare through the use of DL.

8.1.3 DL in smart vehicle

DL and IoT can join forces in smart vehicles to improve autonomous driving and ensure a safer and more comfortable driving experience. IoT is a network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, and connectivity which enables these objects to collect and exchange data. In a smart vehicle, IoT devices can be used to gather data on a wide range of systems, such as navigation, engine performance, and vehicle safety. By using DL algorithms, it is possible to extract insights and make predictions from the data collected by IoT devices. Combined, DL and IoT can facilitate numerous uses in intelligent automobiles, including:

Table 12 DL in healthcare

Author(s)	Work Done	Model
Mukherjee et al. (2020)	Proposed DL method for Human activity recognition using smartphone sensors	EnsemConvNet
Shah et al. (2020)	Assessed healthcare service quality through analyzed patient opinions found online	CNN-LSTM
Ahmed et al. (2021)	Detection of patient discomfort through the use of IoT devices and DL	Mask-R CNN
Alhussein and Muhammad (2018)	Mobile Healthcare Framework for Voice Pathology Detection was developed	VGG16, CaffeNet
Gao et al. (2018)	A DL-based human monitoring system for healthcare was developed	LSTM
Veeramakali et al. (2021)	Developed a secure diagnostic model using DL, Blockchain, and IoT	Optimal Deep Neural Networks (ODNN)

Table 13 DL in smart vehicles

Author(s)	Work done	Model
Zhu et al. (2020)	The crowd sensed the application of DL	DLMV
Miglani and Kumar (2019)	Mobility prediction was reviewed, challenges were identified, and solutions using DL were proposed	CNN
Wirges et al. (2018)	DL-based grid map augmentation was performed	UNet, Resnet
Zhao et al. (2017)	The prediction of the steering angle and speed of automobiles was carried out	DBN
Khan et al. (2020b)	The prediction of a driver's behaviour using DL and blockchain technology was conducted	CNN
Lina López et al. (2018)	Smart charging of electric vehicles was facilitated using DL	KNN, SNN, DNN
Canli and Toklu (2021)	Smart parking was implemented using DL	LSTM
Ahmad et al. (2020)	The security of smart vehicles from relay attacks was addressed using DL	CART
Kim et al. (2021)	Dynamic object classification for intelligent vehicles was performed using DL	CNN
Xing et al. (2021)	The analysis of driving behaviour and the prediction of connected vehicles were conducted using DL	RNN, LSTM

- Autonomous driving: analysing sensor data from cameras, lidar, radar, GPS, and other sources to navigate and make decisions on the road.
- Predictive maintenance: analysing sensor data to predict when maintenance is needed and schedule it at the most convenient time.
- Vehicle safety: analysing sensor data to detect potential hazards and act to avoid accidents.
- Advanced driver-assistance systems (ADAS) (Tigadi et al. 2016): Using sensors and DL algorithms to assist drivers in navigating, braking, and steering.

The combination of DL and IoT in smart vehicles can help to create a more convenient, safe and efficient driving experience by automating many of the tasks that human drivers currently perform.

Table 13 condenses the input of various experts in the smart vehicle sector utilizing DL.

8.1.4 DL in smart homes

DL and IoT can also be used together in smart homes to enhance the capabilities of home automation and create a more convenient and efficient living experience for residents. In a smart home, IoT devices can be used to gather data on a wide range of systems, such as lighting, heating, and appliances. By using DL algorithms, it is possible to extract insights and make predictions from the data collected by IoT devices. Together, DL and IoT can enable a wide range of applications in smart homes, such as:

- Smart lighting: adjust lighting levels and colour temperature based on the preferences of the residents and the current time of day.
- Smart heating and cooling: automatically adjust the temperature based on the weather forecast and the preferences of the residents.
- Smart appliances: automatically controlling appliances such as washing machines, dryers, and ovens based on the preferences and routines of the residents.
- Home surveillance: monitoring the home using IoT cameras and DL algorithms to detect suspicious behaviour and improve security.

Overall, the combination of DL and IoT in smart homes can help to create a more convenient, efficient and secure living environment for residents. Table 14 summarizes the input from various experts in the smart home sector using DL.

8.2 DL in IoT devices

DL in IoT devices refers to the integration of deep neural networks in the IoT environment. IoT devices generate vast amounts of data that can be used to train DL models, enabling the development of smart and autonomous systems. DL algorithms are capable of learning patterns and relationships in large amounts of data and are widely used for various applications such as image recognition, NLP, and predictive analytics. By incorporating DL into IoT devices, it becomes possible to enable real-time decision-making and automate complex tasks, leading to improved efficiency and reduced human intervention. Some examples

Table 14 DL in smart homes

Author(s)	Work done	Model
Dey et al. (2018)	Predicting energy consumption in homes through DL	GMDH
Mehmood et al. (2019)	Identifying objects with IoT and DL	MQTT
Bianchi et al. (2019)	Human activity recognition using DL and IoT	CNN
Danaei Mehr and Polat (2019)	Recognise human healthcare using DL and IoT	CNN
Fang and Hu (2014)	Recognise human activity using DL and IoT	DBN
Liciotti et al. (2020)	Proposed A sequential DL application for activity recognition in smart homes	LSTM
Natani et al. (2019)	DL was used for multi-resident activity recognition in ambient sensing smart homes	RNN
Popa et al. (2019)	A DL model was developed for home automation and energy reduction	LSTM
Choi et al. (2013)	Human behaviour prediction for smart homes using DL is being done	DBN-R

of IoT devices that use DL include smart homes, wearable devices, and autonomous vehicles. However, there are still some challenges in implementing DL in IoT devices such as limited computational resources, power constraints, and data privacy and security issues. Despite these challenges, the integration of DL in IoT devices holds great promise for the future and is poised to revolutionize the way we interact with technology. Here are a few examples of IoT devices that use DL:

- Smart home devices: these devices use DL to analyse data from sensors and cameras to automatically control lighting, temperature, and security systems.
- Wearable devices: fitness trackers and smartwatches use DL algorithms to analyse data such as heart rate, sleep patterns, and physical activity to provide personalized health insights.
- Autonomous vehicles: DL algorithms are used in self-driving cars to detect and respond to road conditions, traffic signals, and other vehicles in real-time.
- Industrial IoT devices: DL algorithms are used in manufacturing and production environments to optimize production processes and improve efficiency.
- Healthcare IoT devices: DL algorithms are used in medical devices such as smart insulin pumps and continuous glucose monitoring systems to provide personalized health insights and recommendations.

These are just a few examples of how DL is being integrated into IoT devices. With the continuous advancement of technology, we can expect to see more innovative applications of DL in IoT devices in the future

9 Challenges of IoT for DL

This section delves into the difficulties encountered by DL when processing IoT data or using IoT devices. The challenges are outlined below:

- Lack of dataset: in DL, a large dataset is needed to train the model and improve its performance. The lack of datasets for IoT applications can be a hurdle for integrating DL models in IoT. Researchers often have to create their datasets and work with limited data. This can be a time-consuming and challenging process, particularly for datasets related to human health or other sensitive areas. There are a few platforms that provide IoT datasets, but they are often limited in scope. This is one of the reasons why there is a need for more robust and diverse IoT datasets to be made available to researchers.
- Pre-processing: preparing and pre-processing data for training DL models is a crucial step in the process. Raw data from IoT devices often need to be cleaned and transformed to be in the appropriate format for the model. This includes tasks such as removing outliers, handling missing values, and scaling the data. In addition, the data needs to be split into training, validation, and test sets for the model to be properly trained and evaluated. Without this step, the model may not perform well or may be prone to overfitting.
- Security and privacy: data security and privacy are important considerations when working with DL models, particularly when it comes to IoT data. As you mentioned, IoT datasets are often created by researchers for specific research projects, which can make them valuable targets for data breaches or hacking attempts. Additionally, the

data collected by IoT devices may contain sensitive information about individuals, such as personal health data or location information, which must be protected to maintain privacy. Several steps can be taken to secure and protect IoT data, such as:

  - Encrypting the data both in transit and at rest
  - Using secure protocols for communication between devices and servers
  - Regularly updating software and firmware to address any known vulnerabilities
  - Implementing access controls and authentication mechanisms to limit who can access the data.
  - Conducting regular security audits and vulnerability assessments.
  - Anonymizing or de-identifying data before using it for training models

- Challenges of 6V's: the massive volume of data can pose significant challenges for DL models, particularly in terms of computational complexity and the need for large amounts of labelled data. The high number of parameters and attributes in voluminous data can result in complex DL models with long run time complexity, and noisy or unlabelled data can negatively impact model accuracy. Managing the variety of data received from different IoT devices can also be a challenge, as conflicts in data management can lead to errors in the DL model. Additionally, the high velocity of data can be a concern as it can overwhelm the DL model and cause it to crash if it is not able to process the input data quickly enough. The veracity of IoT-generated data is also important, as data from untrustworthy sources can compromise the security of the system. In addition to this, proper data filtering is important in online streaming data, as it can vary, which might cause different inputs to the model and lead to poor performance.
- Model interpretability: DL models can be difficult to interpret due to their complexity and the use of non-linear functions. This can make it challenging to understand the reasoning behind the model's decisions and to identify potential biases in the data. This can be a significant issue for IoT applications, as decisions made by these models may have real-world consequences, and end users need to understand the reasoning behind these decisions. Researchers have developed methods such as saliency maps, feature visualization, and LIME to help interpret the decision of DL models

10 Conclusion

In this research, the versatility of DL is investigated in the IoT realm. It's noted that there is a rising trend in adopting DL in IoT devices and domains, with a broad range of practicality. DL has shown great potential in revolutionizing IoT devices and applications. With the increasing amount of data generated by IoT devices and the need for intelligent decision-making, DL algorithms provide the necessary tools to process and analyse this data in real-time. As technology advances, DL will likely play a critical role in shaping the future of IoT and unlocking new possibilities in various areas. The significant impact of this literature, as outlined in the designed research questions in Sect. 2.2, on the scientific community is outlined below.

RQ1 The role of DL frameworks is to provide an interface for developers to build, train, and deploy DL models. As discussed in Sect. 6.4, different frameworks for DL are available, and each has its working capabilities. The Trending and most usable frameworks are

TensorFlow, which offers a comprehensive ecosystem for building and deploying ML models. PyTorch provides an easy-to-use interface for building and training dynamic neural networks. Deeplearning4j, A Java-based framework for building and deploying DL models for production environments. The Microsoft cognitive toolkit, A Microsoft-developed framework for building and deploying DL models on a variety of platforms. Keras, A high-level API that enables the quick and easy building of neural networks. MXNet, A flexible and efficient framework for building and deploying DL models, particularly for large-scale distributed systems. Caffe, A fast, lightweight framework for building and deploying DL models.

RQ2 DL models play a crucial role in IoT data processing and analysis by allowing Anomaly detection, Predictive maintenance, Image and speech recognition, Time-series forecasting, Data compression and dimensionality reduction, Decision-making, and more. CNNs, RNNs, AE, GANs, LSTM Networks, VAE, and more DL models frequently used to handle IoT data and efficiently manage IoT networks.

RQ3 IoT devices pose several challenges for implementing DL models, including Limited computing power and memory on IoT devices, Lack of labelled data for training models Power consumption and energy constraints, Limited bandwidth and connectivity, Security and privacy concerns, Integration with legacy systems and protocols, Scalability and management of many devices, Cost and deployment barriers. To overcome IoT challenges for DL, organizations can adopt techniques such as edge computing to reduce data transfer, federated learning to improve privacy and security, transfer learning to reduce data labelling requirements, and ensure device interoperability through compatibility standards. Organizations can also invest in powerful computing resources and implement data protection measures to ensure data privacy and security.

RQ4 DL will play a crucial role in the future of the IoT by enabling devices to make decisions based on data and perform tasks with minimal human intervention. It will allow IoT devices to process and analyse large amounts of data in real-time, making accurate predictions and enhancing their decision-making abilities. Integrating DL into IoT will improve the efficiency of various applications such as predictive maintenance, smart homes, and autonomous vehicles. The future direction for DL in IoT will be focused on making devices more intelligent, secure, and efficient in data processing, communication, and decision-making. The future direction for using DL in IoT will likely include Edge computing, where DL models will be deployed directly on IoT devices for real-time decision-making. DL algorithms will be used to predict equipment failures in predictive maintenance. Anomaly detection in IoT data to improve system performance and security. Computer vision for object detection and recognition in IoT applications. NLP for voice-based interaction with IoT devices.

RQ5 The computational limits of DL include, Data Requirements: DL models necessitate ample amounts of information to attain substantial precision. Computational Power: Training large models requires significant computational resources, including memory and processing power. Overfitting: DL models are prone to overfitting, meaning they perform well on the training data but not on new, unseen data, Data Privacy and Bias: Collecting and using large amounts of personal data for training can raise privacy concerns, and models may also perpetuate existing biases. Interpretability: DL models can be difficult to interpret and understand, making it challenging to debug or analyse errors. Convergence: DL models can be difficult to train, and convergence is not guaranteed. Generalization: DL models can have difficulty generalizing to new, unseen data.

RQ6 DL has shown great potential in the field of IoT due to its ability to learn from large and complex datasets, and to make accurate predictions based on this learning. DL algorithms can help in extracting meaningful insights from massive amounts of data generated by IoT devices, thereby enabling smarter decision-making and predictive maintenance. DL algorithms are also highly adaptable and can learn from data in real-time, which is essential for managing the dynamic and ever-changing nature of IoT environments. Additionally, DL can help in reducing power consumption and optimizing resource utilization in IoT systems, thus improving their efficiency and reliability. Overall, DL provides a powerful tool for realizing the full potential of IoT and can help in solving many of the challenges faced in IoT, making it a highly desirable technology for future IoT applications.

Author contributions DT wrote the manuscript.JKS and SS provided the relevant information and ideas. All the authors reviewed the manuscript.

Funding No funding was received.

Declarations

Conflict of interest There is no conflict of interest to declare. All authors have seen and agree with the contents of the manuscript.

Ethical approval The reported work did not involve any human participation and/or did not harm the welfare of animals.

References

Achouch M, Dimitrova M, Ziane K, Sattarpanah Karganroudi S, Dhouib R, Ibrahim H, Adda M (2022) On predictive maintenance in industry 4.0: overview, models, and challenges. Appl Sci 12(16):8081

Adedoja A, Owolawi PA, Mapayi T (2019) Deep learning based on NASNet for plant disease recognition using leave images. In: 2019 international conference on advances in big data, computing and data communication systems (icABCD). pp 1–5

Agostinelli F, Hoffman M, Sadowski P, Baldi P (2014) Learning activation functions to improve deep neural networks. arXiv preprint. https://arxiv.org/abs/1412.6830

Ahmad U, Song H, Bilal A, Alazab M, Jolfaei A (2020) Securing smart vehicles from relay attacks using machine learning. J Supercomput 76(4):2665–2682

Ahmed I, Jeon G, Piccialli F (2021) A deep-learning-based smart healthcare system for patient's discomfort detection at the edge of internet of things. IEEE Internet Things J 8(13):10318–10326

Albahrani SA, Mahajan D, Kargarrazi S, Schwantuschke D, Gneiting T, Senesky DG, Khandelwal S (2020) Extreme temperature modeling of ALGAN/GAN HEMTS. IEEE Trans Electron Devices 67(2):430–437

Alhussein M, Muhammad G (2018) Voice pathology detection using deep learning on mobile healthcare framework. IEEE Access 6:41034–41041

Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Van Esesn BC, Awwal AAS, Asari VK (2018) The history began from Alexnet: a comprehensive survey on deep learning approaches. arXiv preprint. https://arxiv.org/abs/1803.01164

Alyamkin S, Ardi M, Berg AC, Brighton A, Chen B, Chen Y, Cheng H-P, Fan Z, Feng C, Fu B et al (2019) Low-power computer vision: status, challenges, and opportunities. IEEE J Emerg Sel Top Circuits Syst 9(2):411–421

Andics A, McQueen JM, Petersson KM, Gál V, Rudas G, Vidnyánszky Z (2010) Neural mechanisms for voice recognition. NeuroImage 52(4):1528–1540

Andrews JG, Ghosh A, Muhamed R (2007) Fundamentals of WiMAX: understanding broadband wireless networking. Pearson Education, London

Arasteh H, Hosseinnezhad V, Loia V, Tommasetti A, Troisi O, Shafie-khah M, Siano P (2016) IoT-based smart cities: a survey. In: 2016 IEEE 16th international conference on environment and electrical engineering (EEEIC). IEEE, pp 1–6

Ashok S, Kishore G, Rajesh V, Suchitra S, Sophia SGG, Pavithra B (2020) Tomato leaf disease detection using deep learning techniques. In: 2020 5th international conference on communication and electronics systems (ICCES). IEEE, pp 979–983
Azzam R, Alkendi Y, Taha T, Huang S, Zweiri Y (2020) A stacked LSTM-based approach for reducing semantic pose estimation error. IEEE Trans Instrum Meas 70:1–14
Baker SB, Xiang W, Atkinson I (2017) Internet of things for smart healthcare: technologies, challenges, and opportunities. IEEE Access 5:26521–26544
Ballester P, Araujo RM (2016) On the performance of GoogleNet and Alexnet applied to sketches. In: Thirtieth AAAI conference on artificial intelligence
Baloglu UB, Talo M, Yildirim O, Tan RS, Acharya UR (2019) Classification of myocardial infarction with multi-lead ECG signals and deep CNN. Pattern Recogn Lett 122:23–30
Bandara K, Bergmeir C, Hewamalage H (2020) LSTM-MSNet: leveraging forecasts on sets of related time series with multiple seasonal patterns. IEEE Trans Neural Netw Learn Syst 32(4):1586–1599
Bello O, Zeadally S, Badra M (2017) Network layer inter-operation of device-to-device communication technologies in internet of things (IoT). Ad Hoc Netw 57:52–62
Bianchi V, Bassoli M, Lombardo G, Fornacciari P, Mordonini M, De Munari I (2019) IoT wearable sensor and deep learning: an integrated approach for personalized human activity recognition in a smart home environment. IEEE Internet Things J 6(5):8553–8562
Bisdikian C (2001) An overview of the bluetooth wireless technology. IEEE Commun Mag 39(12):86–94
Brown RE, Milner PM (2003) The legacy of Donald O. Hebb: more than the Hebb synapse. Nat Rev Neurosci 4(12):1013–1019
Caffe. https://caffe.berkeleyvision.org
Cambria E, White B (2014) Jumping NLP curves: a review of natural language processing research. IEEE Comput Intell Mag 9(2):48–57
Canli H, Toklu S (2021) Deep learning-based mobile application design for smart parking. IEEE Access 9:61171–61183
Caro F, Sadr R (2019) The Internet of Things (IoT) in retail: bridging supply and demand. Bus Horiz 62(1):47–54
Celebi ME, Aydin K (eds) (2016) Unsupervised learning algorithms, vol 9. Springer, Cham, p 103
Cerchecci M, Luti F, Mecocci A, Parrino S, Peruzzi G, Pozzebon A (2018) A low power IoT sensor node architecture for waste management within smart cities context. Sensors 18(4):1282
Chadebec C, Thibeau-Sutre E, Burgos N, Allassonnière S (2022) Data augmentation in high dimensional low sample size setting using a geometry-based variational autoencoder. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2022.3185773
Chen M, Shi X, Zhang Y, Wu D, Guizani M (2017) Deep feature learning for medical image analysis with convolutional autoencoder neural network. IEEE Trans Big Data 7(4):750–758
Chen L, Zhou M, Su W, Wu M, She J, Hirota K (2018) Softmax regression based deep sparse autoencoder network for facial emotion recognition in human–robot interaction. Inf Sci 428:49–61
Chen X, Chen W, Hou L, Hu H, Bu X, Zhu Q (2020a) A novel data-driven rollover risk assessment for articulated steering vehicles using RNN. J Mech Sci Technol 34(5):2161–2170
Chen J, Du L, Liao L (2020b) Discriminative mixture variational autoencoder for semisupervised classification. IEEE Trans Cybern 52(5):3032–3046
Chen P, Fu X, Wang X (2021a) A graph convolutional stacked bidirectional unidirectional-LSTM neural network for metro ridership prediction. IEEE Trans Intell Transport Syst. https://doi.org/10.1109/TITS.2021.3065404
Chen B, Liu X, Zheng Y, Zhao G, Shi Y-Q (2021b) A robust GAN-generated face detection method based on dual-color spaces and an improved Xception. IEEE Trans Circuits Syst Video Technol. https://doi.org/10.1109/TCSVT.2021.3116679
Choi S, Kim E, Oh S (2013) Human behavior prediction for smart homes using deep learning. In: 2013 IEEE RO-MAN. IEEE, pp 173–179
Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1251–1258
Cichy RM, Khosla A, Pantazis D, Torralba A, Oliva A (2016) Comparison of deep neural networks to spatiotemporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. Sci Rep 6(1):1–13
Cunningham P, Cord M, Delany SJ (2008) Supervised learning. In: Machine learning techniques for multimedia: case studies on organization and retrieval. pp 21–49
Danaei Mehr H, Polat H (2019) Human activity recognition in smart home with deep learning approach. In: 2019 7th international Istanbul smart grids and cities congress and fair (ICSG). IEEE, pp 149–153

De Santo A, Galli A, Gravina M, Moscato V, Sperlì G (2020) Deep learning for HDD health assessment: an application based on LSTM. IEEE Trans Comput 71(1):69–80

Deeplearning4j. https://deeplearning4j.konduit.ai

Deng L (2016) Deep learning: from speech recognition to language and multimodal processing. APSIPA Trans Signal Inf Process. https://doi.org/10.1017/ATSIP.2015.22

Dewangan G, Maurya S (2021) Fault diagnosis of machines using deep convolutional beta-variational autoencoder. IEEE Trans Artif Intell 3(2):287–296

Dey N, Fong S, Song W, Cho K (2018) Forecasting energy consumption from smart home sensor network by deep learning. In: Smart trends in information technology and computer communications: second international conference, SmartCom 2017, Pune, India, August 18–19, 2017, revised selected papers 2. Springer, pp 255–265

Du X, Ma C, Zhang G, Li J, Lai Y-K, Zhao G, Deng X, Liu Y-J, Wang H (2020) An efficient LSTM network for emotion recognition from multichannel EEG signals. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2020.3013711

Elkholy MM, Mostafa M, Ebied HM, Tolba MF (2020) Hyperspectral unmixing using deep convolutional autoencoder. Int J Remote Sens 41(12):4799–4819

Emami H, Aliabadi MM, Dong M, Chinnam RB (2020) SPA-GAN: spatial attention GAN for image-to-image translation. IEEE Trans Multimed 23:391–401

Essien A, Giannetti C (2020) A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders. IEEE Trans Ind Inform 16(9):6069–6078

Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, Cui C, Corrado G, Thrun S, Dean J (2019) A guide to deep learning in healthcare. Nat Med 25(1):24–29

Fan H, Zhang F, Wei Y, Li Z, Zou C, Gao Y, Dai Q (2021) Heterogeneous hypergraph variational autoencoder for link prediction. IEEE Trans Pattern Anal Mach Intell 44(8):4125–4138

Fang H, Hu C (2014) Recognizing human activity in smart home using deep learning algorithm. In: Proceedings of the 33rd Chinese control conference. pp 4716–4720

Farahani B, Firouzi F, Chang V, Badaroglu M, Constant N, Mankodiya K (2018) Towards fog-driven IoT eHealth: promises and challenges of IoT in medicine and healthcare. Future Gener Comput Syst 78:659–676

Feigl T, Kram S, Woller P, Siddiqui RH, Philippsen M, Mutschler C (2020) RNN-aided human velocity estimation from a single IMU. Sensors 20(13):3656

Francis M, Deisy C (2019) Disease detection and classification in agricultural plants using convolutional neural networks—a visual understanding. In: 2019 6th international conference on signal processing and integrated networks (SPIN). IEEE, pp 1063–1068

Gao Y, Xiang X, Xiong N, Huang B, Lee HJ, Alrifai R, Jiang X, Fang Z (2018) Human action monitoring for healthcare based on deep learning. IEEE Access 6:52277–52285

Gao S, Huang Y, Zhang S, Han J, Wang G, Zhang M, Lin Q (2020) Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. J Hydrol 589:125188

Gayathri S, Wise DCJW, Shamini PB, Muthukumaran N (2020) Image analysis and detection of tea leaf disease using deep learning. In: 2020 international conference on electronics and sustainable communication systems (ICESC). IEEE, pp 398–403

Geetharamani G, Pandian A (2019) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Comput Electr Eng 76:323–338

Gharibzahedi SMT, Barba FJ, Zhou J, Wang M, Altintas Z (2022) Electronic sensor technologies in monitoring quality of tea: a review. Biosensors 12(5):356

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge

Gorostiza EM, Galilea JLL, Meca FJM, Monzú DS, Zapata FE, Puerto LP (2011) Infrared sensor system for mobile-robot positioning in intelligent spaces. Sensors 11:5416–5438

Greff K, Srivastava RK, Koutník J, Steunebrink BR, Schmidhuber J (2016) LSTM: a search space odyssey. IEEE Trans Neural Netw Learn Syst 28(10):2222–2232

Habi HV, Messer H (2020) Recurrent neural network for rain estimation using commercial microwave links. IEEE Trans Geosci Remote Sens 59(5):3672–3681

Hadjeres G, Nielsen F (2020) Anticipation-RNN: enforcing unary constraints in sequence generation, with application to interactive music generation. Neural Comput Appl 32(4):995–1005

Han F, Yao J, Zhu H, Wang C (2020) Underwater image processing and object detection based on deep cnn method. J Sens. https://doi.org/10.1155/2020/6707328

Hashida H, Kawamoto Y, Kato N (2019) Efficient delay-based internet-wide scanning method for IoT devices in wireless LAN. IEEE Internet Things J 7(2):1364–1374

Hayman S (1999) The McCulloch–Pitts model. In: IJCNN'99. International joint conference on neural networks. proceedings (Cat. No. 99CH36339), vol 6. IEEE, pp 4438–4439
He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
Hong C, Yu J, Wan J, Tao D, Wang M (2015) Multimodal deep autoencoder for human pose recovery. IEEE Trans Image Process 24(12):5659–5670
Hou R, Shen Y, Zhao H, Hu H, Lu J, Long T (2020) Power loss characterization and modeling for GAN-based hardswitching half-bridges considering dynamic on-state resistance. IEEE Trans Transport Electrif 6(2):540–553
IEEE 802.11. https://www.ieee802.org/11/
IEEE 802.15. https://standards.ieee.org/ieee/802.15.4/7029/
IEEE 802.15. https://www.ieee802.org/16/tge/
IEEE 802.15. https://www.ieee802.org/15/
Ilsvrc-2015. https://scholar.google.com/citations?user=mG4imMEAAAAJ&hl=en&oi=ao
Ilsvrc-2015. https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon
Ilsvrc-2015. https://scholar.google.com/citations?user=WLN3QrAAAAAJ&hl=en&oi=ao
Iqbal T, Ali H (2018) Generative adversarial network for medical images (MI-GAN). J Med Syst 42(11):1–11
Iqbal MA, Talukder KH (2020) Detection of potato disease using image segmentation and machine learning. In: 2020 international conference on wireless communications signal processing and networking (WiSPNET). pp 43–47
Irsoy O, Alpaydın E (2017) Unsupervised feature extraction with autoencoder trees. Neurocomputing 258:63–73
Jahangir H, Tayarani H, Gougheri SS, Golkar MA, Ahmadian A, Elkamel A (2020) Deep learning-based forecasting approach in smart grids with microclustering and bidirectional LSTM network. IEEE Trans Ind Electron 68(9):8298–8309
Jasim MA, Al-Tuwaijari JM (2020) Plant leaf diseases detection and classification using image processing and deep learning techniques. In: 2020 international conference on computer science and software engineering (CSASE). IEEE, pp 259–265
Ji Z, Li S, Pang Y (2018) Fusion-attention network for person search with free-form natural language. Pattern Recogn Lett 116:205–211
Jia L, Gu Y, Cheng K, Yan H, Ren F (2020) BeAware: convolutional neural network (CNN) based user behavior understanding through WiFi channel state information. Neurocomputing 397:457–463
Jia Y, Liu B, Dou W, Xu X, Zhou X, Qi L, Yan Z (2022) CroAPP: a CNN-based resource optimization approach in edge computing environment. IEEE Trans Ind Inform. https://doi.org/10.1109/TII.2022.3154473
Jiang K, Wang Z, Yi P, Wang G, Lu T, Jiang J (2019) Edge-enhanced GAN for remote sensing image super-resolution. IEEE Trans Geosci Remote Sens 57(8):5799–5812
Jiang F, Lu Y, Chen Y, Cai D, Li G (2020) Image recognition of four rice leaf diseases based on deep learning and support vector machine. Comput Electron Agric 179:105824
Jiao R, Peng K, Dong J (2020) Remaining useful life prediction of lithium-ion batteries based on conditional variational autoencodersparticle filter. IEEE Trans Instrum Meas 69(11):8831–8843
Jin W, Kim D (2018) Development of virtual resource based IoT proxy for bridging heterogeneous web services in IoT networks. Sensors 18(6):1721
Jin L, Yan J, Du X, Xiao X, Fu D (2020) RNN for solving time-variant generalized Sylvester equation with applications to robots and acoustic source localization. IEEE Trans Ind Inform 16(10):6359–6369
Joyce JM (2011) Kullback–Leibler divergence. In: International encyclopedia of statistical science. Springer, Berlin, Heidelberg, pp 720–722
Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: a survey. Comput Electron Agric 147:70–90
Kang B, Choo H (2018) An experimental study of a reliable IoT gateway. ICT Express 4(3):130–133
Karadağ K, Tenekeci ME, Taşaltın R, Bilgili A (2020) Detection of pepper fusarium disease using machine learning algorithms based on spectral reflectance. Sustain Comput Inform Syst 28:100299
Karagiannis V, Chatzimisios P, Vazquez-Gallego F, Alonso-Zarate J (2015) A survey on application layer protocols for the internet of things. Trans IoT Cloud Comput 3(1):11–17
Karim F, Majumdar S, Darabi H, Chen S (2017) LSTM fully convolutional networks for time series classification. IEEE Access 6:1662–1669
Karlik B, Olgac AV (2011) Performance analysis of various activation functions in generalized MLP architectures of neural networks. Int J Artif Intell Expert Syst 1(4):111–122

Ke Z, Vikalo H (2021) Real-time radio technology and modulation classification via an LSTM auto-encoder. IEEE Trans Wirel Commun 21(1):370–382
Keras. https://keras.io
Keras. https://onnx.ai
Khairdoost N, Shirpour M, Bauer MA, Beauchemin SS (2020) Real-time driver maneuver prediction using LSTM. IEEE Trans Intell Veh 5(4):714–724
Khalil K, Eldash O, Kumar A, Bayoumi M (2019) Economic LSTM approach for recurrent neural networks. IEEE Trans Circuits Syst II Express Briefs 66(11):1885–1889
Khan RU, Zhang X, Kumar R (2019) Analysis of ResNet and GoogleNet models for malware detection. J Comput Virol Hacking Tech 15(1):29–37
Khan AH, Li S, Chen D, Liao L (2020a) Tracking control of redundant mobile manipulator: an RNN based metaheuristic approach. Neurocomputing 400:272–284
Khan MZ, Khan MUG, Irshad O, Iqbal R (2020b) Deep learning and blockchain fusion for detecting driver's behavior in smart vehicles. Internet Technol Lett 3(6):e119
Khanh QV, Hoai NV, Manh LD, Le AN, Jeon G (2022) Wireless communication technologies for IoT in 5G: vision, applications, and challenges. Wirel Commun Mob Comput 2022:1–12
Khazeiynasab SR, Zhao J, Batarseh I, Tan B (2021) Power plant model parameter calibration using conditional variational autoencoder. IEEE Trans Power Syst 37(2):1642–1652
Khurana D, Koli A, Khatter K, Singh S (2023) Natural language processing: state of the art, current trends and challenges. Multimed Tools Appl 82(3):3713–3744
Kim S, Kim S (2018) User preference for an IoT healthcare application for lifestyle disease management. Telecommun Policy 42(4):304–314
Kim S, Lee J, Kang S, Lee J, Yoo H-J (2020) A power-efficient CNN accelerator with similar feature skipping for face recognition in mobile devices. IEEE Trans Circuits Syst I 67(4):1181–1193
Kim K, Kim C, Jang C, Sunwoo M, Jo K (2021) Deep learning-based dynamic object classification using LiDAR point cloud augmented by layer-based accumulation for intelligent vehicles. Expert Syst Appl 167:113861
Kingma DP, Welling M (2019) An introduction to variational autoencoders. Found Trends® Mach Learn 12(4):307–392
Kollias D, Zafeiriou S (2020) Exploiting multi-CNN features in CNN-RNN based dimensional emotion recognition on the OMG in-the-wild dataset. IEEE Trans Affect Comput 12(3):595–606
Kong L, Tan J, Huang J, Chen G, Wang S, Jin X, Zeng P, Khan M, Das SK (2022) Edge-computing-driven internet of things: a survey. ACM Comput Surv 55(8):1–41
Kuutti S, Bowden R, Jin Y, Barber P, Fallah S (2020) A survey of deep learning applications to autonomous vehicle control. IEEE Trans Intell Transport Syst 22(2):712–733
Lakhwani K, Gianey H, Agarwal N, Gupta S (2019) Development of IoT for smart agriculture a review. In: Emerging trends in expert applications and security: proceedings of ICETEAS 2018. Springer, pp 425–432
Langer S (2021) Approximating smooth functions by deep neural networks with sigmoid activation function. J Multivar Anal 182:104696
LeCun Y, Bengio Y, Hinton G (2015a) Deep learning. Nature 521(7553):436–444
LeCun Y et al (2015b) LeNet-5, convolutional neural networks, vol 20, no 5, p 14. http://yann.lecun.com/exdb/lenet
Lee S-J, Chen T, Yu L, Lai C-H (2018) Image classification based on the boost convolutional neural network. IEEE Access 6:12755–12768
Li J (2022) Recent advances in end-to-end automatic speech recognition. APSIPA Trans Signal Inf Process 11(1):1–64
Li J, Huang X, Gamba P, Bioucas-Dias JM, Zhang L, Benediktsson JA, Plaza A (2014) Multiple feature learning for hyperspectral image classification. IEEE Trans Geosci Remote Sens 53(3):1592–1606
Li S, Xu LD, Zhao S (2015) The internet of things: a survey. Inf Syst Front 17:243–259
Li W, Fu H, Yu L, Gong P, Feng D, Li C, Clinton N (2016) Stacked autoencoder-based deep learning for remote-sensing image classification: a case study of African land-cover mapping. Int J Remote Sens 37(23):5632–5646
Li S, Liu X, Wang Y, Wang X (2019) A cubic quality loss function and its applications. Qual Reliab Eng Int 35(4):1161–1179
Li X, Tang J, Zhang Q, Gao B, Yang JJ, Song S, Wu W, Zhang W, Yao P, Deng N et al (2020a) Power-efficient neural network with artificial dendrites. Nat Nanotechnol 15(9):776–782
Li L, Zou C, Zheng Y, Su Q, Fu H, Tai C-L (2020b) Sketch-R2CNN: an RNN-rasterization-CNN architecture for vector sketch recognition. IEEE Trans Vis Comput Graph 27(9):3745–3754

Li Q, Cheng M, Wang J, Sun B (2020c) LSTM based phishing detection for big email data. IEEE Trans Big Data. https://doi.org/10.1109/TBDATA.2020.2978915
Li R, Hu Y, Liang Q (2020d) T2F-LSTM method for long-term traffic volume prediction. IEEE Trans Fuzzy Syst 28(12):3256–3264
Li L, Yan J, Wang H, Jin Y (2020e) Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. IEEE Trans Neural Netw Learn Syst 32(3):1177–1191
Li C, Zhang Z, Song R, Cheng J, Liu Y, Chen X (2021a) EEG-based emotion recognition via neural architecture search. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2021.3130387
Li W, Liang Z, Ma P, Wang R, Cui X, Chen P (2021b) Hausdorff GAN: improving GAN generation quality with Hausdorff metric. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2021.3062396
Li L, Yan J, Zhang Y, Zhang J, Bao J, Jin Y, Yang X (2022) Learning generative RNN-ODE for collaborative time-series and event sequence forecasting. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2022.3185115
Liciotti D, Bernardini M, Romeo L, Frontoni E (2020) A sequential deep learning application for recognising human activities in smart homes. Neurocomputing 396:501–513
Lin L, Li M, Ma L, Nazari M, Mahdavi S, Yunianta A (2020) Using fuzzy uncertainty quantization and hybrid RNN-LSTM deep learning model for wind turbine power. IEEE Trans Ind Appl. https://doi.org/10.1109/TIA.2020.2999436
Lina López K, Gagné C, Gardner M-A (2018) Demand-side management using deep learning for smart charging of electric vehicles. IEEE Trans Smart Grid 10(3):2683–2691
Liu Q, Wang J (2008) A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming. IEEE Trans Neural Netw 19(4):558–570
Liu L, Shen C, van den Hengel A (2016) Cross-convolutional-layer pooling for image recognition. IEEE Trans Pattern Anal Mach Intell 39(11):2305–2313
Liu H, Zhou J, Zheng Y, Jiang W, Zhang Y (2018) Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. ISA Trans 77:167–178
Liu Z, Wu J, Fu L, Majeed Y, Feng Y, Li R, Cui Y (2019a) Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. IEEE Access 8:2327–2336
Liu H, Lang B, Liu M, Yan H (2019b) CNN and RNN based payload classification methods for attack detection. Knowl Based Syst 163:332–341
Lopez-Alvis J, Laloy E, Nguyen F, Hermans T (2021) Deep generative models in inversion: the impact of the generator's nonlinearity and development of a new approach based on a variational autoencoder. Comput Geosci 152:104762
Lore KG, Akintayo A, Sarkar S (2017) LLNet: a deep autoencoder approach to natural low-light image enhancement. Pattern Recogn 61:650–662
Lu C, Wang Z-Y, Qin W-L, Ma J (2017) Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. Signal Process 130:377–388
Lu S, Lu Z, Zhang Y-D (2019) Pathological brain detection based on Alexnet and transfer learning. J Comput Sci 30:41–47
Lv N, Chen C, Qiu T, Sangaiah AK (2018) Deep learning and superpixel feature extraction based on contractive autoencoder for change detection in SAR images. IEEE Trans Ind Inform 14(12):5530–5538
Ma M, Mao Z (2020) Deep-convolution-based LSTM network for remaining useful life prediction. IEEE Trans Ind Inform 17(3):1658–1667
Ma Z, Chang D, Xie J, Ding Y, Wen S, Li X, Si Z, Guo J (2019a) Fine-grained vehicle classification with channel max pooling modified CNNs. IEEE Trans Veh Technol 68(4):3224–3233
Ma Y, Xu X, Yu Q, Zhang Y, Li Y, Zhao J, Wang G (2019b) LungBRN: a smart digital stethoscope for detecting respiratory disease using Bi-ResNet deep learning algorithm. In: 2019b IEEE biomedical circuits and systems conference (BioCAS). IEEE, pp 1–4
Ma Y, Zhou G, Wang S (2019c) WiFi sensing with channel state information: a survey. ACM Comput Surv (CSUR) 52(3):1–36
Ma J, Liu H, Peng C, Qiu T (2020) Unauthorized broadcasting identification: a deep LSTM recurrent learning approach. IEEE Trans Instrum Meas 69(9):5981–5983
Ma F, Sun B, Li S (2021) Facial expression recognition with visual transformers and attentional selective fusion. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2021.3122146
Mao L, Yan Y, Xue J-H, Wang H (2020) Deep multi-task multi-label CNN for effective facial attribute classification. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2020.2969189
Mct. https://learn.microsoft.com/en-us/cognitive-toolkit/
Mehmood F, Ullah I, Ahmad S, Kim D (2019) Object detection mechanism based on deep learning algorithm using embedded IoT devices for smart home appliances control in CoT. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-019-01272-8

Meneghello F, Calore M, Zucchetto D, Polese M, Zanella A (2019) IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices. IEEE Internet Things J 6(5):8182–8201
Meng Z, Zhan X, Li J, Pan Z (2018) An enhancement denoising autoencoder for rolling bearing fault diagnosis. Measurement 130:448–454
Miglani A, Kumar N (2019) Deep learning models for traffic flow prediction in autonomous vehicles: a review, solutions, and challenges. Veh Commun 20:100184
Militante SV, Gerardo BD, Dionisio NV (2019) Plant leaf detection and disease recognition using deep learning. In: 2019 IEEE Eurasia conference on IoT, communication and engineering (ECICE). IEEE, pp 579–582
Minsky M, Papert SA (2017) Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry. MIT Press, Cambridge
Mishra SK, Sarkar A (2022) Service-oriented architecture for internet of things: a semantic approach. J King Saud Univ Comput Inf Sci 34(10):8765–8776
Mishra S, Sachan R, Rajpal D (2020) Deep convolutional neural network based detection system for real-time corn plant disease recognition. Procedia Comput Sci 167:2003–2010
Misra D (2019) Mish: a self regularized non-monotonic activation function. arXiv preprint. https://arxiv.org/abs/1908.08681
Mitchell TM (2007) Machine learning, vol 1. McGraw-Hill, New York
Muangprathub J, Boonnam N, Kajornkasirat S, Lekbangpong N, Wanichsombat A, Nillaor P (2019) IoT and agriculture data analysis for smart farm. Comput Electron Agric 156:467–474
Mukherjee D, Mondal R, Singh PK, Sarkar R, Bhattacharjee D (2020) EnsemConvNet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications. Multimed Tools Appl 79(41):31663–31690
Mulligan G (2007) The 6LoWPAN architecture. In: Proceedings of the 4th workshop on embedded networked sensors. pp 78–82
Mustafa MS, Husin Z, Tan WK, Mavi MF, Farook RSM (2020) Development of automated hybrid intelligent system for herbs plant classification and early herbs plant disease detection. Neural Comput Appl 32(15):11419–11441
Muthu Ramya C, Shanmugaraj M, Prabakaran R (2011) Study on ZigBee technology. In: 2011 3rd international conference on electronics computer technology, vol 6. IEEE, pp 297–301
MXNet. https://mxnet.apache.org/versions/1.9.1/
Natani A, Sharma A, Peruma T, Sukhavasi S (2019) Deep learning for multi-resident activity recognition in ambient sensing smart homes. In: 2019 IEEE 8th global conference on consumer electronics (GCCE). IEEE, pp 340–341
Niu S, Li B, Wang X, Lin H (2020) Defect image sample generation with GAN for improving defect recognition. IEEE Trans Autom Sci Eng 17(3):1611–1622
Noda K, Yamaguchi Y, Nakadai K, Okuno HG, Ogata T (2015) Audio-visual speech recognition using deep learning. Appl Intell 42(4):722–737
Othman E, Bazi Y, Alajlan N, Alhichri H, Melgani F (2016) Using convolutional features and a sparse autoencoder for land-use scene classification. Int J Remote Sens 37(10):2149–2167
Otter DW, Medina JR, Kalita JK (2020) A survey of the usages of deep learning for natural language processing. IEEE Trans Neural Netw Learn Syst 32(2):604–624
Pantic I, Paunovic J, Cumic J, Valjarevic S, Petroianu GA, Corridon PR (2022) Artificial neural networks in contemporary toxicology research. Chemico-Biol Interact 369:110269
Park SH, Park JK (2016) IoT industry & security technology trends. Int J Adv Smart Converg 5(3):27–31
Park K, Kim J, Lee J (2019) Visual field prediction using recurrent neural network. Sci Rep 9(1):1–12
Parthasarathy P, Vivekanandan S (2020) A typical IoT architecture-based regular monitoring of arthritis disease using time wrapping algorithm. Int J Comput Appl 42(3):222–232
Phasinam K, Kassanuk T, Shinde PP, Thakar CM, Sharma DK, Mohiddin MK, Rahmani AW (2022) Application of IoT and cloud computing in automation of agriculture irrigation. J Food Qual 2022:1–8
Popa D, Pop F, Serbanescu C, Castiglione A (2019) Deep learning model for home automation and energy reduction in a smart home environment platform. Neural Comput Appl 31(5):1317–1337
Popović T, Latinović N, Pešić A, Zečević Ž, Krstajić B, Djukanović S (2017) Architecting an IoT-enabled platform for precision agriculture and ecological monitoring: a case study. Comput Electron Agric 140:255–265
Prakash CD, Karam LJ (2021) It GAN do better: GAN-based detection of objects on images with varying quality. IEEE Trans Image Process 30:9220–9230
pytorch. https://pytorch.org
Qi M, Wang Y, Qin J, Li A, Luo J, Van Gool L (2019) StagNet: an attentive semantic RNN for group activity and individual action recognition. IEEE Trans Circuits Syst Video Technol 30(2):549–565

Qiang N, Dong Q, Ge F, Liang H, Ge B, Zhang S, Sun Y, Gao J, Liu T (2020) Deep variational autoencoder for mapping functional brain networks. IEEE Trans Cogn Dev Syst 13(4):841–852
Qu Y, Yu S, Zhou W, Tian Y (2020) GAN-driven personalized spatial-temporal private data sharing in cyber-physical social systems. IEEE Trans Netw Sci Eng 7(4):2576–2586
Quispe R, Pedrini H (2019) Improved person re-identification based on saliency and semantic parsing with deep neural network models. Image Vis Comput 92:103809
Ramachandran P, Zoph B, Le QV (2017) Searching for activation functions. arXiv preprint. https://arxiv.org/abs/1710.05941
Rao G, Huang W, Feng Z, Cong Q (2018) LSTM with sentence representations for document-level sentiment classification. Neurocomputing 308:49–57
Rebennack S, Krasko V (2020) Piecewise linear function fitting via mixed-integer linear programming. INFORMS J Comput 32(2):507–530
Ruan Y-P, Ling Z (2021) Emotion-regularized conditional variational autoencoder for emotional response generation. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2021.3073809
Russell SJ (2010) Artificial intelligence a modern approach. Pearson Education Inc., London
Saidi SJ, Matic S, Gasser O, Smaragdakis G, Feldmann A (2022) Deep dive into the IoT backend ecosystem. In: Proceedings of the 22nd ACM internet measurement conference. pp 488–503
Salari A, Djavadifar A, Liu XR, Najjaran H (2022) Object recognition datasets and challenges: a review. Neurocomputing
Samuel SSI (2016) A review of connectivity challenges in IoT-smart home. In: 2016 3rd MEC international conference on big data and smart city (ICBDSC). IEEE, pp 1–4
Sanchez-Iborra R, Cano M-D (2016) State of the art in LP-WAN solutions for industrial IoT services. Sensors 16(5):708
Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
Schmidt-Hieber J (2020) Nonparametric regression using deep neural networks with ReLU activation function. Ann Stat 48(4):1875–1897
Selvaraj S, Sundaravaradhan S (2020) Challenges and opportunities in IoT healthcare systems: a systematic review. SN Appl Sci 2(1):139
Shah AM, Yan X, Shah SAA, Mamirkulova G (2020) Mining patient opinion to evaluate the service quality in healthcare: a deep-learning approach. J Ambient Intell Human Comput 11(7):2925–2942
Shanthi T, Sabeenian RS (2019) Modified Alexnet architecture for classification of diabetic retinopathy images. Comput Electr Eng 76:56–64
Shao L, Zhu F, Li X (2014) Transfer learning for visual categorization: a survey. IEEE Trans Neural Netw Learn Syst 26(5):1019–1034
Sharma A, Liu X, Yang X, Shi D (2017) A patch-based convolutional neural network for remote sensing image classification. Neural Netw 95:19–28
Shu Y, Yi R, Xia M, Ye Z, Zhao W, Chen Y, Lai Y-K, Liu Y-J (2021) GAN-based multi-style photo cartoonization. IEEE Trans Vis Comput Graph. https://doi.org/10.1109/TVCG.2021.3067201
Sicari S, Rizzardi A, Coen-Porisini A (2019) Smart transport and logistics: a node-RED implementation. Internet Technol Lett 2(2):e88
Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint. https://arxiv.org/abs/1409.1556
Soui M, Smiti S, Mkaouer MW, Ejbali R (2020) Bankruptcy prediction using stacked auto-encoders. Appl Artif Intell 34(1):80–100
Souibgui MA, Kessentini Y (2020) DE-GAN: a conditional generative adversarial network for document enhancement. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2020.3022406
Stiller B, Schiller E, Schmitt C, Ziegler S, James M (2020) An overview of network communication technologies for IoT. Handbook of Internet-of-Things, 12.
Su Y, Zhao Y, Sun M, Zhang S, Wen X, Zhang Y, Liu X, Liu X, Tang J, Wu W et al (2021) Detecting outlier machine instances through Gaussian mixture variational autoencoder with one dimensional CNN. IEEE Trans Comput 71(4):892–905
Subetha T, Khilar R, Christo MS (2021) A comparative analysis on plant pathology classification using deep learning architecture—ResNet and VGG19. Mater Today. https://doi.org/10.1016/j.matpr.2020.11.993
Sujatha R, Chatterjee JM, Jhanjhi NZ, Brohi SN (2021) Performance of deep learning vs machine learning in plant leaf disease detection. Microprocess Microsyst 80:103615
Sun Q, Liu X, Bourennane S, Liu B (2021) Multiscale denoising autoencoder for improvement of target detection. Int J Remote Sens 42(8):3002–3016
Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press, Cambridge

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1–9

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 2818–2826

Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence

Tan K, Xu B, Kumar A, Nachmani E, Adi Y (2020) SAGRNN: self-attentive gated RNN for binaural speaker separation with interaural cue preservation. IEEE Signal Process Lett 28:26–30

Tang J, Deng C, Huang G-B (2015) Extreme learning machine for multilayer perceptron. IEEE Trans Neural Netw Learn Syst 27(4):809–821

Tao W, Li C, Song R, Cheng J, Liu Y, Wan F, Chen X (2020) EEG-based emotion recognition via channel-wise attention and self attention. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2020.3025777

Tasyurek M, Celik M (2020) RNN-GWR: a geographically weighted regression approach for frequently updated data. Neurocomputing 399:258–270

tesnsorflow. https://www.tensorflow.org

Thakur D, Kumar Y, Kumar A, Singh PK (2019) Applicability of wireless sensor networks in precision agriculture: a review. Wirel Pers Commun 107(1):471–512

Thakur D, Kumar Y, Vijendra S (2020) Smart irrigation and intrusions detection in agricultural fields using IoT. Procedia Comput Sci 167:154–162

Thies J, Alimohammad A (2019) Compact and low-power neural spike compression using undercomplete autoencoders. IEEE Trans Neural Syst Rehabil Eng 27(8):1529–1538

Tigadi A, Gujanatti R, Gonchi A, Klemsscet B (2016) Advanced driver assistance systems. Int J Eng Res Gen Sci 4(3):151–158

Tiwari D, Ashish M, Gangwar N, Sharma A, Patel S, Bhardwaj S (2020) Potato leaf diseases detection using deep learning. In: 2020 4th international conference on intelligent computing and control systems (ICICCS). IEEE, pp 461–466

Tran N-T, Tran V-H, Nguyen N-B, Nguyen T-K, Cheung N-M (2021) On data augmentation for GAN training. IEEE Trans Image Process 30:1882–1897

Tzounis A, Katsoulas N, Bartzanas T, Kittas C (2017) Internet of things in agriculture, recent advances and future challenges. Biosyst Eng 164:31–48

Ullah I, Ahmad S, Mehmood F, Kim D (2019) Cloud based IoT network virtualization for supporting dynamic connectivity among connected devices. Electronics 8(7):742

Veeramakali T, Siva R, Sivakumar B, Senthil Mahesh PC, Krishnaraj N (2021) An intelligent internet of things-based secure healthcare framework using blockchain technology with an optimal deep learning model. J Supercomput 77(9):9576–9596

Vincent P (2011) A connection between score matching and denoising autoencoders. Neural Comput 23(7):1661–1674

Wang Y, Yao H, Zhao S (2016) Auto-encoder based dimensionality reduction. Neurocomputing 184:232–242

Wang S, Jiang Y, Hou X, Cheng H, Du S (2017) Cerebral micro-bleed detection based on the convolution neural network with rank based average pooling. IEEE Access 5:16576–16583

Wang W, Yang D, Chen F, Pang Y, Huang S, Ge Y (2019) Clustering with orthogonal autoencoder. IEEE Access 7:62421–62432

Wang Q, Bu S, He Z (2020a) Achieving predictive and proactive maintenance for high-speed railway power equipment with LSTM-RNN. IEEE Trans Ind Inform 16(10):6509–6517

Wang X, Tan K, Du Q, Chen Y, Du P (2020b) CVA^{2}E: a conditional variational autoencoder with an adversarial training process for hyperspectral imagery classification. IEEE Trans Geosci Remote Sens 58(8):5676–5692

Wang J, Zhang W, Yang H, Michael Yeh C-C, Wang L (2021a) Visual analytics for RNN-based deep reinforcement learning. IEEE Trans Vis Comput Graph. https://doi.org/10.1109/TVCG.2021.3076749

Wang H, Lu B, Li J, Liu T, Xing Y, Lv C, Cao D, Li J, Zhang J, Hashemi E (2021b) Risk assessment and mitigation in local path planning for autonomous vehicles with LSTM based predictive model. IEEE Trans Autom Sci Eng. https://doi.org/10.1109/TASE.2021.3075773

Wang Y, Ma X, Wang J, Hou S, Dai J, Gu D, Wang H (2022) Robust AUV visual loop-closure detection based on variational autoencoder network. IEEE Trans Ind Inform 18(12):8829–8838

Wen L, Li X, Gao L (2020) A transfer convolutional neural network for fault diagnosis based on ResNet-50. Neural Comput Appl 32(10):6111–6124

Wirges S, Stiller C, Hartenbach F (2018) Evidential occupancy grid map augmentation using deep learning. In: 2018 IEEE intelligent vehicles symposium (IV). IEEE, pp 668–673
Wortmann F, Flüchter K (2015) Internet of things. Bus Inf Syst Eng 57(3):221–224
Wu H, Huang Q, Wang D, Gao L (2018) A CNN-SVM combined model for pattern recognition of knee motion using mechanomyography signals. J Electromyogr Kinesiol 42:136–142
Wu J-Y, Wu M, Chen Z, Li X-L, Yan R (2021) Degradation-aware remaining useful life prediction with LSTM autoencoder. IEEE Trans Instrum Meas 70:1–10
Wu S, Sun F, Zhang W, Xie X, Cui B (2022) Graph neural networks in recommender systems: a survey. ACM Comput Surv 55(5):1–37
Xia M, Shao H, Ma X, de Silva CW (2021) A stacked GRU-RNN-based approach for predicting renewable energy and electricity load for smart grid operation. IEEE Trans Ind Inform 17(10):7050–7059
Xia X, Pan X, Li N, He X, Ma L, Zhang X, Ding N (2022a) GAN-based anomaly detection: a review. Neurocomputing
Xia W, Zhang Y, Yang Y, Xue J-H, Zhou B, Yang M-H (2022b) GAN inversion: a survey. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2022.3181070
Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1492–1500
Xie Z, Jin L, Luo X, Sun Z, Liu M (2020a) RNN for repetitive motion generation of redundant robot manipulators: an orthogonal projection-based scheme. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2020.3028304
Xie M, Li C, Liu X, Wong T-T (2020b) Manga filling style conversion with screentone variational autoencoder. ACM Trans Graph 39(6):1–15
Xing Y, Lv C, Mo X, Hu Z, Huang C, Hang P (2021) Toward safe and smart mobility: energy-aware deep learning for driving behavior analysis and prediction of connected vehicles. IEEE Trans Intell Transport Syst 22(7):4267–4280
Xu J, Xiang L, Liu Q, Gilmore H, Wu J, Tang J, Madabhushi A (2015) Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. IEEE Trans Med Imaging 35(1):119–130
Xu Y, Chen Z, Xie Z, Wu L (2017) Quality assessment of building footprint data using a deep autoencoder network. Int J Geogr Inf Sci 31(10):1929–1951
Xu J, Li Z, Du B, Zhang M, Liu J (2020a) Reluplex made more practical: leaky ReLU. In: 2020a IEEE symposium on computers and communications (ISCC). IEEE, pp 1–7
Xu D, Wei C, Peng P, Xuan Q, Guo H (2020b) GE-GAN: a novel deep learning framework for road traffic state estimation. Transport Res C 117:102635
Xu L, Zhou X, Tao Y, Liu L, Yu X, Kumar N (2021) Intelligent security performance prediction for IoT-enabled healthcare networks using an improved cnn. IEEE Trans Ind Inform 18(3):2063–2074
Yan X, Ai T, Yang M, Tong X (2021) Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. Int J Geogr Inf Sci 35(3):490–512
Ye F, Bors AG (2021) Lifelong mixture of variational autoencoders. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2021.3096457
Ye L, Liu Z, Wang Y (2020) Dual convolutional LSTM network for referring image segmentation. IEEE Trans Multimed 22(12):3224–3235
Yeo Y-J, Shin Y-G, Park S, Ko S-J (2021) Simple yet effective way for improving the performance of GAN. IEEE Trans Neural Netw Learn Syst 33(4):1811–1818
Yi J, Zhu Y, Xie J, Chen Z (2021) Cross-modal variational auto-encoder for content-based micro-video background music recommendation. IEEE Trans Multimed. https://doi.org/10.1109/TMM.2021.3128254
Yu S, Principe JC (2019) Understanding autoencoders with information theoretic concepts. Neural Netw 117:104–123
Yu X-M, Feng W-Z, Wang H, Chu Q, Chen Q (2020) An attention mechanism and multi-granularity-based Bi-LSTM model for chinese Q&A system. Soft Comput 24(8):5831–5845
Yuan X, Li L, Shardt YAW, Wang Y, Yang C (2020) Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development. IEEE Trans Ind Electron 68(5):4404–4414
Zabalza J, Ren J, Zheng J, Zhao H, Qing C, Yang Z, Du P, Marshall S (2016) Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. Neurocomputing 185:1–10
Zaimi A, Wabartha M, Herman V, Antonsanti P-L, Perone CS, Cohen-Adad J (2018) AxonDeepSeg: automatic axon and myelin segmentation from microscopy data using convolutional neural networks. Sci Rep 8(1):1–11

Zeng N, Zhang H, Song B, Liu W, Li Y, Dobaie AM (2018) Facial expression recognition via learning deep sparse autoencoders. Neurocomputing 273:643–649
Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017a) Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans Image Process 26(7):3142–3155
Zhang G, Kou L, Zhang L, Liu C, Da Q, Sun J (2017b) A new digital watermarking method for data integrity protection in the perception layer of IoT. Secur Commun Netw. https://doi.org/10.1155/2017/3126010
Zhang L, Wang S, Liu B (2018a) Deep learning for sentiment analysis: a survey. Wiley Interdiscip Rev Data Min Knowl Discov 8(4):e1253
Zhang H, Weng T-W, Chen P-Y, Hsieh C-J, Daniel L (2018b) Efficient neural network robustness certification with general activation functions. Advances in neural information processing systems, vol 31
Zhang M, Li W, Tao R, Li H, Du Q (2021a) Information fusion for classification of hyperspectral and LiDAR data using IP-CNN. IEEE Trans Geosci Remote Sens 60:1–12
Zhang H, Yuan J, Tian X, Ma J (2021b) GAN-FM: infrared and visible image fusion using GAN with full-scale skip connection and dual Markovian discriminators. IEEE Trans Comput Imaging 7:1134–1147
Zhang Q, Zeng F, Xiao Z, Jiang H, Regan AC, Yang K, Zhu Y (2022) Toward predicting stay time for private car users: a RNN-NALU approach. IEEE Trans Veh Technol 71(6):6007–6018
Zhao C, Gong J, Lu C, Xiong G, Mei W (2017) Speed and steering angle prediction for intelligent vehicles based on deep belief network. In: 2017 IEEE 20th international conference on intelligent transportation systems (ITSC). IEEE, pp 301–306
Zhao Z-Q, Zheng P, Xu S-T, Wu X (2019) Object detection with deep learning: a review. IEEE Trans Neural Netw Learn Syst 30(11):3212–3232
Zhao T, Li F, Tian P (2020) A deep-learning method for device activity detection in MMTC under imperfect CSI based on variationalautoencoder. IEEE Trans Veh Technol 69(7):7981–7986
Zheng W, Wang K, Wang F-Y (2020) GAN-based key secret-sharing scheme in blockchain. IEEE Trans Cybern 51(1):393–404
Zheng Y, Sui X, Jiang Y, Che T, Zhang S, Yang J, Li H (2021) SymReg-GAN: symmetric image registration with generative adversarial networks. IEEE Trans Pattern Anal Mach Intell 44(9):5631–5646
Zhou M (2022) Evolution from AI, IoT and Big Data analytics to metaverse. IEEE/CAA J Autom Sin 9(12):2041–2042
Zhou X, Li Y, Liang W (2020) CNN-RNN based intelligent recommendation for online medical pre-diagnosis support. IEEE/ACM Trans Comput Biol Bioinform 18(3):912–921
Zhu X, Luo Y, Liu A, Tang W, Bhuiyan MZA (2020) A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility. IEEE Trans Intell Transport Syst 22(7):4648–4659
Zou J, Han Y, So S-S (2008) Overview of artificial neural networks. In: Artificial neural networks. pp 14–22

P22

A Kappa Model of Thermotaxis in Juvenile Honeybees

Albin Salazar [1,2] , **Tatjana Petrov** [1,2,3]

[1] *Centre for the Advanced Study of Collective Behaviour, University of Konstanz, Konstanz, Germany*

[2] *Department of Computer and Information Science, University of Konstanz, Konstanz, Germany*

[3] *Department of Mathematics and Geosciences, University of Trieste, Trieste, Italy*

Nature provides a variety of examples of how collective behavior can be observed in diverse animal groups. Yet it remains a difficult challenge to extract from these large groups individual interactions and to assess their roles in such collectives, and vice versa. This challenge is compounded further when performing *in-silico* studies of such complex systems, as one must consider a combinatorial number of interactions as the size of a system increases. To tackle this problem, here we use a rule-based, formal language, KappaLanguage [1], to explore a preliminary case study of a thermotaxis collective behavior observed in juvenile honeybees [2]. Further, we discuss notions of heterogeneity in our case study and argue that KappaLanguage can capture such phenomenon in an intuitive manner, which can potentially facilitate the model construction process and analysis of collective systems.

[1] P. Boutillier, M. Maasha, X. Li, H. F. Medina-Abarca, J. Krivine, J. Feret, I. Cristescu, A. G. Forbes, W. Fontana, Bioinformatics **34**, 13, (2018).
[2] M. Szopek, T. Schmickl, R. Thenius, G. Radspieler, K. Crailsheim, PLoS ONE **8**, 10 (2013).

P23

INFERENCE OF COSMOLOGY FROM THE MORPHOLOGICAL STRUCTURE OF DARK MATTER HALO AND GALAXY DISTRIBUTIONS USING NEURAL NETWORKS

Anirudh Shankar, **Jean-Michel Alimi**

(Presenting author underlined)
Observatory of Paris

Several cosmological models indicate that the fine structure of the universe that we observe today comes from the gravitational amplification of non-linearities of the cosmic matter field. Many of the recent developments in its understanding have risen from the study of N-body cosmological simulations, of which a fair share involve the use of machine learning techniques for a variety of tasks. These techniques allow the identification of multi-dimensional correlations that isn't possible through conventional methods. Our work probes the cosmological trace contained in the non-gaussian nature of the morphological structure of dark matter halo and galaxy distributions deduced from the Quijote[3] and CAMELS[2] simulation suites. We study the high order correlation property from several graph representations of these distributions and isolate through the construction of the graph's minimum-spanning (MST), the cosmological information thus contained. We use Bayesian optimised multi-layer perceptron networks to predict the parameters of the cosmological framework in which the set of simulations are realised, including the establishment of error limits using the Monte Carlo dropout method. We show that an increase in the non-linearity and thus the non-gaussianity of the halo and galaxy distributions allows for significantly better prediction of the cosmology. Through a comparative study of halo and galaxy distributions of the Quijote and CAMELS simulation suites respectively, we discuss the accessibility of cosmology by the different structural parameters of the MST at different length scales [1]. The extensions of our work include other descriptions of graphs, other methods of isolation of their non-gaussian character and also the effect of the presence of modified gravity beyond General Relativity.

[1] Anirudh Shankar and Jean-Michel Alimi. "Cosmology Inference from the morphological structure of the minimal spanning tree of dark matter halos and galaxies distributions". In: *to be submitted to the Astronomy and Astrophysics journal* (2024).

[2] Francisco Villaescusa-Navarro et al. "The camels project: Cosmology and astrophysics with machine-learning simulations". In: *The Astrophysical Journal* 915.1 (2021), p. 71.

[3] Francisco Villaescusa-Navarro et al. "The quijote simulations". In: *The Astrophysical Journal Supplement Series* 250.1 (2020), p. 2.

AI for Spatial Rural-Urban Transformation

Mohamed Rabii Simou[1] , **Safia Loulad**[1] , **Mohamed Benayad**[1] **,and Hassan Rhinane**[1]

[1] *GEOSCIENCES LABORATORY, EARTH SCIENCES DEPARTMENT, FACULTY OF SCIENCES-AIN CHOCK, UNIVERSITY HASSAN II, CASABLANCA, MOROCCO*

In the rapidly evolving field of spatial analysis, the application of Artificial Intelligence technologies has emerged as a critical tool for understanding and managing the complex dynamics of rural-urban transformation. This study introduces a novel approach to spatial analysis through the integration of AI SegFormer, a state-of-the-art transformer model, into the investigation of rural and urban area transformations. The findings will underscore the effectiveness of AI SegFormer in enhancing the precision and depth of spatial analyses, this approach facilitates informed decision-making and policy formulation aimed at sustainable urban planning and rural development.

P25

Refining the Giants: A Comprehensive Review of Fine-Tuning Strategies for Large Language Models

Kunal Singh [1], **Dr. Santosh Deshpande** [2], **and Dr. Swapnaja Patwardhan** [3]

[1,2,3] *MES' Institute of Management and Career Courses, Pune, India*

The number of training parameters and the computational costs have grown exponentially with the increased efficiency of large language models [2]. These large language models (LLMs) have remarkable linguistic and generative abilities that can be leveraged using fine-tuning to optimize their performance across specific tasks. Fine-tuning involves training with fewer parameters, making it more effective over certain domains, and it also utilizes various approaches to cater for varied outputs as required by the particular task. Fine-tuning is a significant technique in the context of current advancements in NLP, with contributions like multilingual chatbots, personalized healthcare assistants, combating misinformation, and shaping the future with few-shot learning and multimodal fine-tuning, thus improving their performance and applicability in real-world scenarios. Reviews of fine-tuning strategies for large language models offer insights into challenge mitigation, prompt engineering, regularization methods, computational efficiency, and techniques like retrieval-augmented generation (RAG) [3], enhancing the performance and applicability of these models. This poster comprehensively dives deep into eight prominent fine-tuning strategies, including instruction tuning [4], transfer learning, basic hyperparameter tuning, retrieval-augmented generation (RAG), parameter-efficient tuning (PEFT) [1], reward modelling, preference learning, and proximal policy optimization (PPO), which significantly enhances the efficiency, scalability, and interpretability of LLMs, followed by a conclusion and future prospectus. Each approach has been reviewed for its unique advantages, challenges, and working methodology for fine-tuning the LLMs.

[1] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter efficient transfer learning for nlp. In International Conference on Machine Learning, pages 2790–2799. PMLR, 2019.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

[3] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In International conference on machine learning, pages 3929–3938. PMLR, 2020.

[4] Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Victoria Lin, Wen-tau Yih, and Srinivasan Iyer. Instruction-tuned language models are better knowledge learners. arXiv preprint arXiv:2402.12847, 2024.

P26

Fast Online Few-Shot Object Detection via Prototype Fusion

Weikai Li[1] , Yanlai Wu[1] , and Yuan Li[2]

[1] *School of Mathematics and Statistics, Chongqing Jiaotong University, Chongqing, 400074, China*

[2] *China Science IntelliCloud Technology, Co., Ltd., Anhui, 230000, China*

Few-shot object detection (FSOD) remains a significant challenge in computer vision, where the majority of existing methods focus on fine-tuning the backbone network to accommodate novel categories. However, these approaches often struggle due to the challenges posed by domain/category shift and limited sample sizes in real-world scenarios, which can limit their performance. To address such issues, three core components are conducted: dual-level multi-scale information integration, support image mask processing, and online prototype refinement. Our unique approach fuses information across network layers and image scales, enhancing discriminative capabilities. We introduce a mask processing mechanism, leveraging segmentation from other models, to effectively utilize support context during training and inference. Moreover, our prototype refinement technique dynamically updates prototypes based on query images, alleviating distribution shift. Our method outperforms existing SOTA approaches on multiple benchmarks, demonstrating significant performance improvements in few-shot object detection.