Artificial Intelligence Review (2023) 56:3619–3658
https://doi.org/10.1007/s10462-022-10259-5

Deep learning for compressive sensing: a ubiquitous systems perspective

Alina L. Machidon[1] · **Veljko Pejović**[1,2]

Published online: 7 September 2022

Abstract
Compressive sensing (CS) is a mathematically elegant tool for reducing the sensor sampling rate, potentially bringing context-awareness to a wider range of devices. Nevertheless, practical issues with the sampling and reconstruction algorithms prevent further proliferation of CS in real world domains, especially among heterogeneous ubiquitous devices. Deep learning (DL) naturally complements CS for adapting the sampling matrix, reconstructing the signal, and learning from the compressed samples. While the CS–DL integration has received substantial research interest recently, it has not yet been thoroughly surveyed, nor has any light been shed on practical issues towards bringing the CS–DL to real world implementations in the ubiquitous computing domain. In this paper we identify main possible ways in which CS and DL can interplay, extract key ideas for making CS–DL efficient, outline major trends in the CS–DL research space, and derive guidelines for the future evolution of CS–DL within the ubiquitous computing domain.

Keywords Neural networks · Deep learning · Compressive sensing · Ubiquitous computing

1 Introduction

The capacity to sense the world around them represents the key affordance of computing devices nowadays found under popular terms, such as the Internet of Things (IoT), cyber-physical systems, and ubiquitous computing (ubicomp). This integration of computing and sensing was essential for achieving such early milestones as the Moon landing and the first humanoid robot in late 1960s. Yet, the moment when the first iPhone hit the shelves in 2008 marked the start of a new era of sensor-computing integration, the one in which compact mobile computing devices equipped with an array of sensors will soon outnumber people on this planet. The ever-increasing range of sensors available on mobile devices,

✉ Alina L. Machidon
alina.machidon@fri.uni-lj.si

Veljko Pejović
veljko.pejovic@fri.uni-lj.si

[1] Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, Ljubljana, Slovenia

[2] Department of Computer Systems, Institut "Jožef Stefan", Jamova cesta 39, Ljubljana, Slovenia

nowadays including multiple cameras, microphones, accelerometers, gyroscopes, location, light and temperature sensors, and wireless gesture recognition sensors, to name a few, enabled revolutionary new services to be provided by the mobiles. Furthermore, in parallel to the rise of smartphone and wearable sensing, the advances in embedded computing have propelled the use of sensors in systems ranging from unmanned areal vehicles (UAVs, i.e. "drones") over factory robots and IoT devices, to self-driving cars. Consequently, the spectrum of applications relying on integrated sensing and computing has already expanded to cover anything from wildfire monitoring to vacuuming a home, and with the increase in number of deployed devices showing no signs of waning, we can safely assume that the true potential of sensing-computing integration is yet to be observed.

Widening the range and the sophistication of sensing-based applications calls for the increased amount of data to be collected as well as for the complex computation to be supported by sensing-computing systems. For instance, high-level inferences from sensor data are possible, but only if enough data is funneled through complex data processing pipelines which include data filtering, extracting features from raw sensor data, and machine learning modeling. Recent advancements in the area of deep learning pushed the complexity of the requested computation and the amount of data needed even further. Thus, models containing millions of parameters can process high-resolution camera images and detect objects present in these images. Similarly, high-frequency samples taken from on-body accelerometers can, with the help of long short-term memory (LSTM) models infer a wearer's physical activity.

Unfortunately, resource-constrained sensing devices, such as various microcontrollers, wearables, IoT devices and similar devices predominant in today's ubiquitous computing deployments often cannot cope with the sampling and processing burden of modern machine learning on sensor data. Equipped with multicore CPUs and GPUs and relatively large storage space, modern smartphones can run certain deep learning models. However, even these high-end devices support only sporadically used and carefully optimized models processing sensor data streams of relatively modest sampling rates (Wu et al. 2019a). Even disregarding the processing burden, sensor sampling is itself sometimes the most energy-hungry aspect of ubiquitous computing (Pramanik et al. 2019). With battery energy being the most precious resource in mobile computing and the battery advances heavily lagging behind the storage and processing component improvements (Pramanik et al. 2019), the problem is unlikely to resolve itself with the incoming generations of ubiquitous computing systems.

Supporting advanced inference applications, while reducing the sampling and processing burden appears unsolvable at the first glance. According to the Nyquist theorem, (low-pass) signals can be reliably reconstructed only if sensor sampling rates are as twice as high as the highest frequency expressed in such signals. Real-world phenomena, however, tend to be fast changing. It appears that objects can be recognized only in images of sufficient resolution, wireless radars can detect small movements only if the signal is sampled millions of times per second, etc.

In 2000s a series of papers by Candès et al. (2006), Donoho (2006), Candes and Romberg (2006), Candes et al. (2006), investigated the properties of signals that can be successfully reconstructed even if sampled at rates lower than prescribed by the Nyquist theorem. Signal sparsity, i.e. a property that in a certain projection most of the signal's components are zero and incoherence, a condition of low correlation between the acquisition domain and the sparsity domain, are needed in order for the signal to be fully preserved with only about $Klog(N/K)$ samples taken from the original K-sparse N-dimensional signal.

Compressive sensing (CS)[1] involves drastically reduced sampling rate, administered when the above conditions are fulfilled, and subsequent signal reconstruction via signal processing, often reduced to finding solutions to underdetermined linear systems. In the last two decades, CS has been successfully demonstrated in image processing, wireless ranging, and numerous other domains.

The benefits of reduced sampling rates do not, however, come for free, as CS remains challenging to implement. First, not all reduced-rate sampling is equal. Compression rates interplay with the properties of the input to impact the ability to successfully reconstruct the signal from limited samples. Furthermore, while the early implementations focused on random sampling, recent research advances demonstrate the utility of carefully crafted reduced sampling strategies (Wang et al. 2019). Second, reconstructing the signal from sparse measurements, in theory, requires solving an NP hard problem of finding non-zero signal components. In practice, the problem is solved through iterative solutions that are nevertheless computationally demanding. Finally, high-level inference, not signal reconstruction, is often the key aim of sensing. Thus, it is essential to construct a full machine learning pipeline that natively supports CS.

1.1 Towards deep learning-supported compressive sensing

Interestingly, the above challenges have a joint characteristic—for a specific use case, a suitable sampling strategy, a reliable reconstruction algorithm, and a highly accurate inference pipeline can be *learned* from the collected sensor data and data labels. Machine learning methods, therefore, naturally augment CS. The inclusion of GPUs and TPUs together with programming support for deep learning (e.g. TensorFlow Lite Li 2020) made DL pervasively possible, even in embedded and mobile computers.

Over the past decade, compressive sensing evolved from theoretical studies and its initial practicality was predominantly limited by the time complexity of the reconstruction algorithms. Deep learning brought tremendous improvements on that front, enabling real-time reconstruction in certain applications. ReconNet (Kulkarni et al. 2016), for example is up to 2700 times faster than the conventional iterative CS algorithm D-AMP (Metzler et al. 2016) and can reconstruct a 256×256 image in only about 0.02 seconds at any given measurement rate. But the true benefits of using deep learning for compressive sensing can be observed in the quality of the reconstruction, where the DL-based approaches surpass conventional algorithms, due to the potential of deep learning to sidestep the sparsity assumptions, and to capture and exploit relevant features in the data.

Especially promising is the revolutionizing potential of CS–DL integration in the area of ubiquitous computing. Here, devices are characterized by wide heterogeneity, and limited computational and battery resources. Besides the general benefits of accelerating signal reconstruction, fine-tuning the sampling matrix, and improving the high-level inference, in the ubiquitous computing domain CS–DL integration can reduce the energy, storage, and processing requirements. As a result, there is a potential for previously prohibitively demanding continuous sensing and inference to finally be realized in certain domains. Furthermore, a graceful degradation in end-result quality can be supported with the CS–DL pipeline (Machidon and Pejović 2022). Through reduced CS sampling and reduced accuracy DL inference we can, in a controlled manner, trade result quality for resource usage.

[1] also known as *compressed sensing* and *sparse sampling*

This allows seamless adaptation of the sensing-inference pipeline, so that complex applications can run on low-resource devices, albeit with limited accuracy. Finally, mobile devices operate in dynamic environments. With the environment so can vary the signal properties (i.e. its sparsity) as well as a user's requirements with respect to the calculated result quality.

Figure 1 depicts possible ways deep learning and compressive sensing can interplay. A common CS pipeline (a) consists of the reduced-frequency sampling, followed by signal reconstruction, from which high-level inferences are made, if needed. Iterative signal reconstruction algorithms, in particular, tend to represent a weak point in the pipeline due to their temporal requirements. Yet, with sufficient CS-sampled and original signal data available, a rather fast-to-query DL reconstruction model can be built. Using DL for signal reconstruction (b) by either mimicking the iterative CS algorithm (Sect. 3.1) or not (Sect. 3.2), has been successfully demonstrated in numerous domains (Kulkarni et al. 2016; Iliadis et al. 2016; Wang et al. 2016; Schlemper et al. 2017; Han et al. 2018b; Kim et al. 2020b). The sampling matrix, too can be adapted to a problem at hand thanks to DL (c). Often an encoder-like structure is trained to guide the sampling[2] in the most efficient manner (Sect. 3.2). Finally, as the reconstructed signal is usually used as a basis for high-level inference, DL allows us to short-circuit the expensive reconstruction step and train a network that provides high-level inferences directly from the CS-sampled data (d) (Sect. 4). The performance of such solutions not only matched, but also significantly exceeded the performance of the standard reconstruction approaches as additional signal structure can be captured by the DL models (Polania and Barner 2017; Ma 2017; Grover and Ermon 2019).

1.2 Survey rationale, research methodology, and survey organization

The above-identified natural links between efficient sampling embodied in CS and powerful learning enabled by DL have recently been recognized by the research community. Tremendous research interest that has spurted is evident in a steady increase in the number of scientific papers published on the topic yearly from 2015 to 2020 (see Fig. 2). The exploration is far from theoretical with a range of application fields, including magnetic resonance imaging (MRI), ultra wideband (UWB) radar ranging, human activity recognition, and numerous other domains benefiting from the CS–DL integration.

The building blocks enabling CS–DL integration, i.e. both compressive sensing and deep learning, have been thoroughly addressed already. Compressive sensing remains the main subject of a few monographs (e.g. Eldar and Kutyniok 2012) that introduce the topic from the historical perspective, present key theoretical postulates, and discuss open challenges in the area of signal sampling and processing. Yet, these almost exclusively focus on the mathematical issues and remain implementation platform-oblivious. The volume by Khosravy et al. (2020) investigates the use of CS in healthcare and considers applications, such as electrocardiogram (ECG) and electroencephalogram (EEG) sensing, that are, with the expansion of wearable computing capabilities, highly relevant for the ubicomp domain. Still, the book focuses on the sensing part and does not discuss potential integration with deep learning.

[2] In this survey we use the term "compressive sensing" in a broader sense, as it is commonly used in the related literature (Shi et al. 2020; Zhao et al. 2020b), yet we acknowledge that by a stricter definition of the term certain methods described in this paper may be considered as "dimensionality reduction", rather than "sensing" approaches.

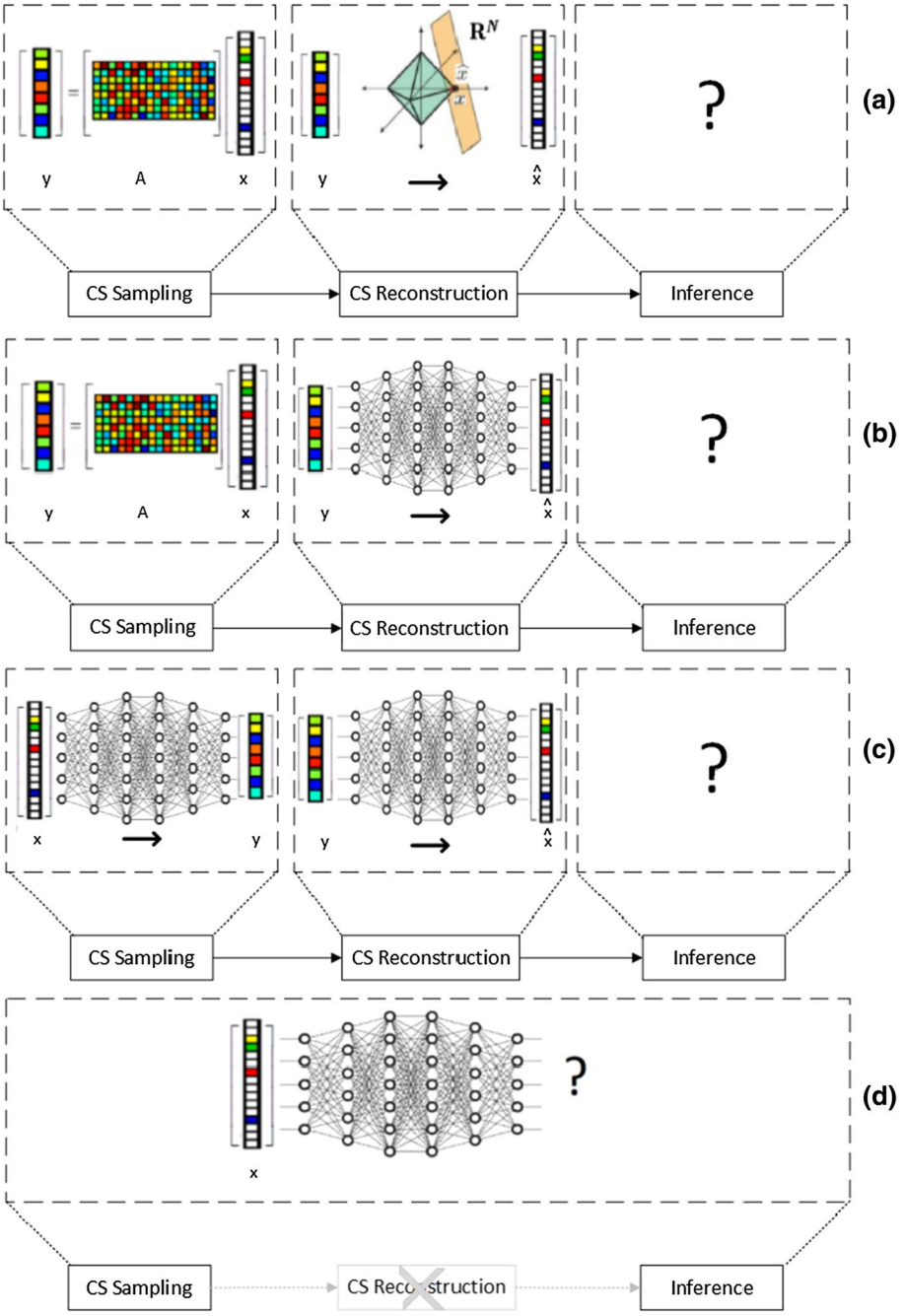

Fig. 1 CS approaches: **a** conventional CS; **b** DL for CS reconstruction; **c** DL for CS sampling and reconstruction; **d** DL for CS direct inference

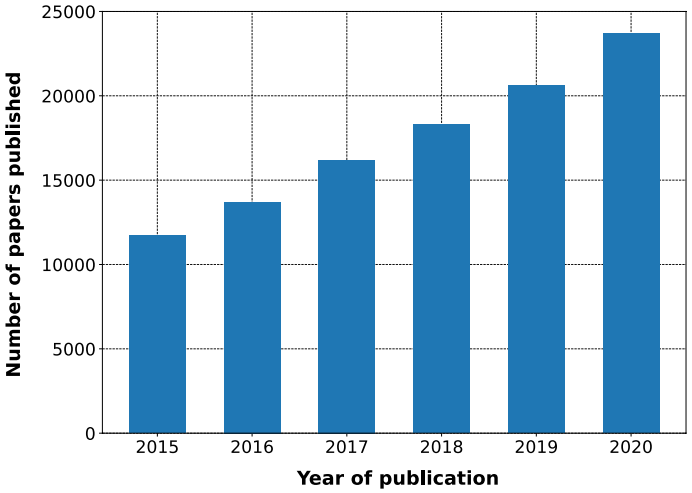

Fig. 2 Number of scientific papers on the topic CS–DL published between 2015 and 2020 (Data from Google Scholar using the search terms "deep learning" and "compressed sensing")

Focused survey papers cover compressive sensing applications in different domains, for instance wireless sensor networks (WSNs) (Wimalajeewa and Varshney 2017), IoT (Djelouat et al. 2018), and EEG processing (Gurve et al. 2020), to name a few. Our survey is orthogonal to these, as we do not focus on a particular domain, but rather merge contributions made in different fields on the common ground of using both the DL techniques and the CS concepts. A summary of the survey articles that are most related to this paper is given in Table 1 and it clearly demonstrates that no published surveys deal with systematically reviewing deep learning for compressive sensing. We opt for this approach with the intention to inform future research in any domain, by providing researchers and practitioners with a toolbox of CS–DL implementations that may be transplanted to any domain. Finally, to the best of our knowledge, ours is the first paper that provides in-detailed consideration of practical issues of CS–DL integration on *ubiquitous* computing devices[3]. Very few papers (even outside surveys) deal with practical implementations of CS on ubiquitous devices. It is our hope that the guidelines on efficient implementations presented in this survey will serve as a foundation for practical realizations of deep learning-supported ubiquitous compressive sensing systems of the future.

An extremely popular research area, deep learning is not short of textbooks, surveys, and tutorial on the topic (e.g. Aggarwal 2018). From a range of DL survey papers, we find (Cheng et al. 2017) and (Choudhary et al. 2020) particularly relevant. Both surveys focus on techniques that prune, compress, quantize, and in other manners reshape powerful DL models so that they can be ran on resource-constrained devices. The expansion of context awareness and artificial intelligence over a wide range of devices and applications is our guiding vision, while reduced sampling rates afforded by CS, together with powerful

[3] In this survey we restrict the term "ubiquitous computing" to devices that integrate computational and sensing capabilities. The devices may or may not be connected to the Internet and may or may not be mobile.

Table 1 Related survey/review papers from the area of compressive sensing

Reference	Year	Domain	Contributions and technical differences
Khosravy et al. (2020)	2020	Health-care	A book on CS applications in healthcare. Focuses on the sensing part and does not discuss potential integration with deep learning.
Wimalajeewa and Varshney (2017)	2017	WSN	A survey on CS for WSN applications. Discusses some challenges that need to be addressed to enable practical implementation, but only with respect to WSN applications.
Djelouat et al. (2018)	2018	IoT	A review on CS for IoT platforms. Surveys the CS-IoT research efforts divided over the different IoT layers. Discusses the efficient hardware implementation of CS reconstruction algorithms, but does not address DL-based reconstruction.
Gurve et al. (2020)	2020	EEG	Summarizes major CS reconstruction algorithms, the sparse basis, and the measurement matrix used in CS for EEG signals. Discusses the advantages and disadvantages of the available (non DL) CS algorithms.

but computationally light inference enabled by intelligently implemented DL pave the path towards our vision.

In this survey we explore *deep learning-supported compressive sensing*, an area that despite the rapidly gaining popularity (see Fig. 2) has not been systematically studied before. Furthermore, we explore it with a particular accent on its real-world applicability within the ubicomp domain. Thus, the objectives of our work remain threefold:

- Present CS fundamentals, DL opportunities, and ubiquitous computing constraints to previously disparate research communities, with a goal of opening discussion and collaboration across the discipline borders;
- Examine a range of research efforts and consolidate DL-based CS advances. In particular, the paper identifies signal sampling, reconstruction, and high-level inference as a major categorization of the reviewed work;
- Recognize major trends in CS–DL research space and derive guidelines for future evolution of CS–DL within the ubicomp domain.

The methodological approach we take focuses on the identification and examination of the most relevant and high impact papers related to the topic of CS–DL, which were published in top scientific journals and renowned international conferences. More specifically, for this survey we:

- Searched Google Scholar with terms including: “deep learning”, “compressive sensing”, “compressed sensing”, “compressed sampling”, “sparse sampling”, “ubiquitous computing” and focused predominantly on well-cited articles (i.e. > 20 citations per year since published) and articles published in 2020 or 2021;
- For journal articles we focused on those published in journals indexed by the Web of Science; for conference articles, we retained those published at conferences supported by a major professional society;
- We manually searched through the proceedings of ubiquitous systems conferences (i.e. ACM MobiSys, ACM UbiComp, ACM SenSys, Asilomar) and machine learning con-

ferences (i.e. NeurIPS, CVPR, ICML, ICLR) for articles related to compressive sensing implementations;
- We identified a small number of very relevant entries on arXiv and opted for including them in the survey, so that the rapid advances in the CS–DL area are not overlooked. Nevertheless, we caution the reader that these entries might not have been peer reviewed yet.

Organization-wise (Figure 3), this paper provides both preliminary material as well as the analysis of recent research trends. With respect to the former, Sect. 2 presents a crash-course overview of compressive sensing, highlighting the necessary conditions for successful sampling, as well as main signal recovery approaches, with an emphasis on algorithm efficiency. Section 3 discusses CS–DL efforts in the area of CS signal reconstruction. The advantages and limitations of different DL flavors with regard to the CS reconstruction challenges are exposed and analyzed, together with the most relevant publications in each case. Table 3 is specifically aimed at practitioners in a need of a quick reference. Machine learning, and deep learning in particular, enables high-level inferences directly from CS-sampled signal without intermediate signal reconstruction. These, so-called, reconstruction-free approaches are presented in Sect. 4. Unique to this survey is also a critical examination of the constraints that CS–DL implementations have to face once deployed in real-world ubiquitous computing environments. These are discussed in Sect. 5, together with key lessons learned from different domains and potential directions future research in the CS–DL for ubiquitous computing. Finally, a coherent frame for our survey is set by the introduction (Sect. 1) and the concluding sections (Sect. 6).

2 Compressive sensing primer

In the first part of this section we aim to bring the area of compressive sensing closer to ubiquitous computing researchers and practitioners. Yet, we focus on the bare essentials and points relevant for real-world implementation of CS and direct an interested reader to more in-depth presentations of the subject, such as (Eldar and Kutyniok 2012). Throughout the section we identify possibilities for deep learning (DL) within the CS domain.

2.1 Theoretical basis

Classical signal processing is based on a notion that signals can be modeled as vectors in a vector space. Nyquist sampling rate requirement was derived based on an assumption that signals may exist anywhere within the given vector space and requires that the sampling frequency is at least as twice as high as the highest frequency component present in the low-pass signal. In reality, however, signals exhibit structure that constrains them to only a subset of possible vectors in a certain geometry, i.e many real-world signals are naturally sparse in a certain basis. Furthermore, if not truly sparse, or even if subject to noise, many signals are compressible—i.e. a limited number of the strongest signal components tends to uniquely describe the signal.

The above observations represent the intuition behind compressive sensing (CS). The idea of joint sensing and compression was theoretically developed in Candès et al. (2006), Donoho (2006) by Emmanuel Candès, Justin Romberg, Terence Tao and David Donoho who also formalized the conditions need for efficient reconstruction of a signal from a

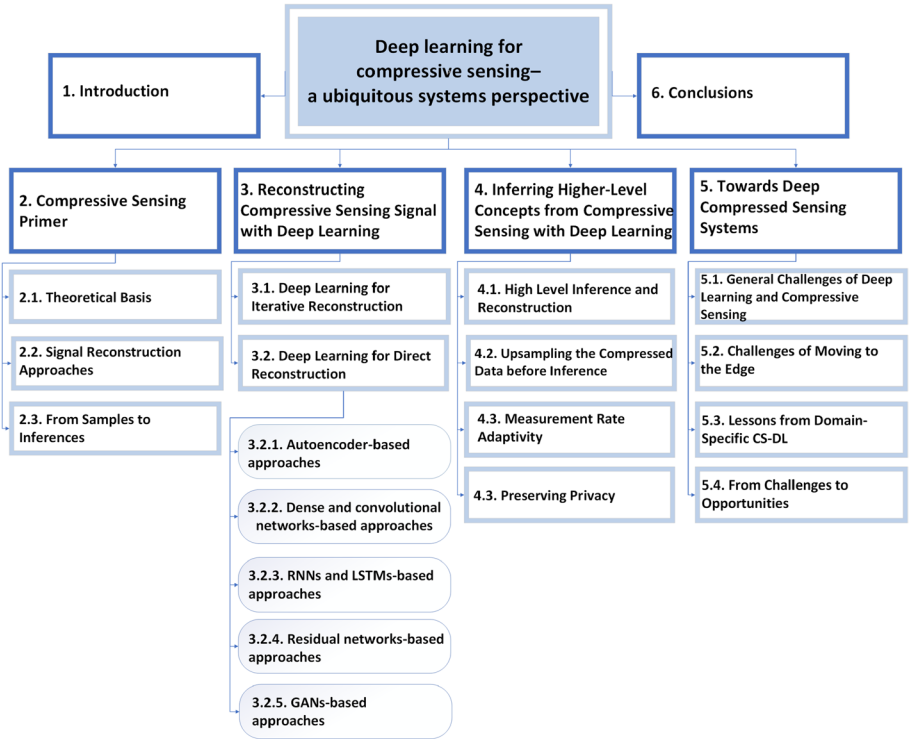

Fig. 3 A schematic illustration of the survey's organization

significantly reduced number of samples, compared to the number of samples assumed under the Nyquist sampling criterion.

The main idea behind CS is that having a K-sparse signal vector $x \in \mathcal{R}^N$ (i.e. a signal that has only K non-zero components), an accurate reconstruction of x can be obtained from the undersampled measurements taken by a sampling:

$$y = Ax \in \mathcal{R}^M$$

where the $M \times N$ matrix A is called *the sensing matrix* (also *the projection matrix*) and is used for sampling the signal. Since $M < N$, this linear system is typically under-determined, permitting an infinite number of solutions. However, according to the CS theory, due to the sparsity of x, the exact reconstruction is possible, by finding the sparsest signal among all those that produce the measurement y, through a norm minimization approach:

$$\begin{aligned} &\text{minimize} \quad \|x\|_0 \\ &\text{subject to} \quad Ax = y \end{aligned}$$

where $\|\cdot\|_0$ is the l_0-norm and denotes the number of non-zero components in x, i.e. the sparsity of x.

However, this is generally an NP-hard problem. An alternative solution is to minimize the l_1 norm, i.e. the sum of the absolute value of vector components:

$$\begin{aligned} &\text{minimize} && \|x\|_1 \\ &\text{subject to} && Ax = y \end{aligned}$$

Since the l_1-norm minimization-guided solution can be found through iterative tractable algorithms, if the solution to the l_0-norm and l_1-norm conditioned systems were the same, the CS-sensed signal could be perfectly reconstructed from M measurements (where M is roughly logarithmic in the data dimensionality $O(K \log(N/K))$), in a reasonable amount of time.

Candès and Tao show that indeed in certain situations solutions to both problems are equivalent. The condition for the above to hold is that the signal's sparsity K is sufficiently high and that the matrix A satisfies certain properties. One of these properties is the so-called Null Space Property (NSP), a necessary and sufficient condition for guarantying the recovery, requiring that every null space vector of the sensing matrix is not concentrating its energy on any set of entries. A stronger condition on the sensing matrix is the Restricted Isometry Property (RIP), which states that A must behave like an almost orthonormal system, but only for sparse input vectors. More formally, matrix A satisfies K-RIP with restricted isometry constant δ_k if for every K-sparse vector x:

$$(1 - \delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k)\|x\|_2^2$$

where $\|\cdot\|_2$ denotes the l_2-norm.

A uniquely optimal solution for the the l_0-norm and l_1-norm conditioned signal reconstruction systems exists, if $\delta_{2k} + \delta_{3k} < 1$. Intuitively, sampling matrices satisfying this condition preserve signal size and therefore do not distort the measured signal, so the reconstruction is accurate.

In practice, however, assessing RIP is computationally difficult. Another related condition, easier to check, is the incoherence, or the low coherence, meaning than the rows of the sampling matrix should be almost orthogonal to the columns of the matrix representing the basis in which the signal is sparse (often the Fourier basis). Additional mathematical properties that the sensing matrix should satisfy for ensuring the stability of the reconstruction have also been introduced in Donoho (2006). From the real world applications perspective, the sensing matrix should ideally fulfill constraints such as: optimal reconstruction performance (high accuracy), optimal sensing (minimum number of measurements needed), low complexity, fast computation and easy and efficient implementation on hardware. Random sensing matrices such as Gaussian or Bernoulli were shown to satisfy the RIP, however, their unstructured nature raises difficulties for hardware implementation and memory storage, and the processing time can be delayed since no accelerated matrix multiplication is available. Structured matrices, such as Circulant or Toeplitz on the other hand, follow a given structure, which reduces the randomness, memory storage, processing time and energy consumption, subsequently. In an application running in a resource constrained environment, such as those for wearable wireless body sensors, this is of great importance.

Finally, real-world data often hide structures, beyond sparsity, that can be exploited. By learning these regularities from the data through the sensing matrix design, the salient information in the data can be preserved, leading to better reconstruction quality. In addition, most of the existing recovery algorithms, rely on the prior knowledge of the degree of sparsity of the signal to optimal tune their parameters. Difficulties might arise especially when the signal is very large or it exhibits great variations in terms of sparsity. In these cases, the conventional CS approaches cannot perform the optimal reconstruction, but a

data-driven approach could learn important signal features and design the signal sampling to work optimally, even for varying sparsity levels.

2.2 Signal reconstruction approaches

The effective and efficient recovery of the original signal from the compressed one is crucial for CS to become a practical tool. Approaches to signal reconstruction from the undersampled measurements can be roughly grouped into convex optimization, greedy, and non-convex minimization algorithms.

The class of *convex optimization* algorithms solve a convex optimization problem, e.g. the l_1-norm optimization problem, to obtain the reconstruction. The Iterative Shrinkage/Thresholding Algorithm (ISTA) (Daubechies et al. 2004) or the Alternating Direction Method of Multipliers (ADMM) (Boyd et al. 2011), are two examples of such convex optimization algorithms. One of the advantages of the algorithms in this class is the small number of measurements required for achieving an exact reconstruction. However, their computational complexity is high.

The greedy approach for CS involves a step-by-step method for finding the support set of the sparse signal by iteratively adding nonzero components, and reconstructing the signal by using the constrained least-squares estimation method. These algorithms are characterized by lower implementation cost and improved running time. On the downside, their performance is highly constrained by the level of sparsity of the signal and in general, their theoretical performance guarantees remain weak.

The Non Convex Recovery Algorithms imply the use of non convex minimization algorithms to solve the convex optimization problem of signal recovery, by replacing the l_1-norm minimization function with other non-convex, surrogate functions (Zhou and Yu 2019; Fan et al. 2019). These methods can show better recovery probability and request fewer measurements than the convex optimization algorithms, but are more challenging to solve because of their non-convexity. Furthermore, their convergence is not always guaranteed.

A joint property of all the above reconstruction algorithms is their high computational cost dictated by the iterative calculations these algorithms rely on. In order to achieve the goal of incorporating CS in real-world ubiquitous computing applications, fast and efficient reconstruction algorithms need to be developed. Deep Learning emerged as an unexpected candidate for such an algorithm. While DL usually requires substantial computing resources and significant memory space for hundreds of thousands of network parameters, the most burdensome computation is still performed during the algorithm training phase and the inference time remains lower than the time needed for running the conventional iterative reconstruction algorithms.

2.3 From samples to inferences

In the last 15 years, compressive sensing transitioned from a theoretical concept to a practical tool. One of the first demonstrations of CS was the so called *one-pixel camera*. Here, a digital micromirror array is used to optically calculate linear projections of the scene onto pseudorandom binary patterns. A single detection element, i.e. a single "pixel" is then evaluated a sufficient number of times, and from these measurements

the original image can be reconstructed. This early success set the trajectory of practical CS, which is nowadays used for a range of image analysis tasks. Thus, CS is used for fMRI image sampling and reconstruction (Chiew et al. 2018; Li 2020), ultrasound images (Kruizinga 2017; Kim et al. 2020a), remote sensing images (Zhao et al. 2020a; Wang 2017), and other image-related domains. WSN data sub-sampling and recovery represents another significant area for CS (Xiao et al. 2019; Liu et al. 2017; Qie et al. 2020) as does speech compression and reconstruction (Shawky 2017; Al-Azawi and Gaze 2017).

Characteristic for most of the signal processing applications listed above, is the departure from signal reconstruction, as the key aim of CS, towards higher level inference, i.e. detection, classification, or prediction. In such cases, signal reconstruction may represent an unnecessary step and may even be counterproductive. Studies (Lohit et al. 2015) have shown theoretical guarantees that the compressed measurements can be directly used for inference problems without performing the recovery step. Therefore, an increasing number of research works aims to solve the problem of learning directly from sparse samples. This is yet another area where neural networks shine. Driven by the fact is it possible to learn directly in the compressed domain, and that neural networks have a inherent ability to extract hidden features, deep learning can be successfully used to infer from the compressed measurements.

3 Reconstructing compressive sensing signal with deep learning

The reconstruction of compressively sensed signal can be reduced to solving, via convex optimization, an l_1-norm conditioned under-determined system of equations (Sect. 2). Neural networks have been used for solving various optimization problems for the last four decades and different neural network models have been developed to solve convex optimization problems (Wang and Liu 2022; Huang and Cui 2019). Within the CS literature, DL solutions for signal reconstruction (Figure 1b) can be classified into those that follow the general philosophy set by the traditional iterative reconstruction algorithms (discussed in Sect. 3.1) and those that harness the modeling power of DL directly (Sect. 3.2).

3.1 Deep learning for iterative reconstruction

The first group of methods for CS signal reconstruction consists of those methods designed to mimic the iterative CS algorithms using dedicated neural network architecture. Most of these methods are based on the technique called algorithm unrolling (or unfolding) that maps each iteration into a network layer, and stacks a determined number of layers together (Fig. 4). The parameters of the algorithm are weights to be learned and after the unrolling, the training data is fed through the network, and stochastic gradient descent is used to update and optimize its parameters.

The first unfolded approach of the traditional iterative algorithm ISTA, called Learned ISTA (LISTA) (Gregor and LeCun 2010) was proposed in the area of sparse coding, i.e. finding a sparse representation of a given signal in a given dictionary. LISTA uses a deep encoder architecture, trained using stochastic gradient descent to minimize a loss function defined as the squared error between the predicted code and the optimal code averaged over the training set. ISTA-Net, too, proposes a DL network mimicking ISTA, but moves

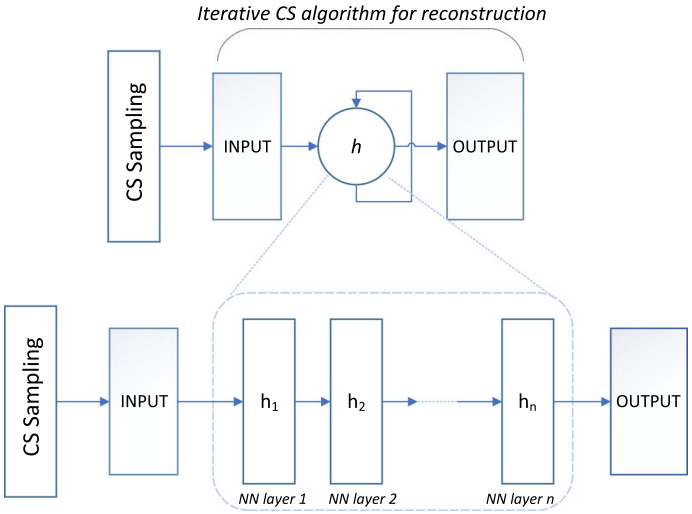

Fig. 4 CS signal reconstruction using an iterative algorithm unrolled over a deep neural network. The CS iterative algorithm's iteration step h is executed a number of times, resulting in the network layers $h_1, h_2, \cdots, h_n$

from the sparse coding to the problem of CS reconstruction (Zhang and Ghanem 2018). In the proposed deep network all parameters (including the transformation matrix) are discriminately learned instead of being hand-crafted or fixed. The authors show that ISTA-Net reduces the reconstruction complexity more than 100 times compared to the traditional ISTA. TISTA is another sparse signal recovery algorithm inspired by ISTA and based on deep learning (Ito et al. 2019). The notable difference between ISTA-Net and TISTA is that the latter uses an error variance estimator, which improves the speed of convergence. The work in Song et al. (2020) also exploits the idea of unfolding the classic iterative algorithm ISTA as a deep neural network, but to deal with nonlinear cases and to solve the sparse nonlinear regression problem, using the Nonlinear Learned Iterative Shrinkage Thresholding Algorithm (NLISTA). For further enhancements to LISTA, a reader is referred to Step-LISTA (Ablin et al. 2019), LISTA-AT (Kim and Park 2020), and GLISTA (Wu et al. 2019b).

A modified version of ISTA, with a better sparsity–undersampling tradeoff, is the Approximate Message Passing (AMP) Donoho et al. (2009) algorithm, inspired by the message passing (or belief propagation) algorithms on graphs. A neural network model for unfolding the iterations of the AMP algorithm was proposed in Borgerding et al. (2017). This AMP variant, called Learned AMP (LAMP) unfolds the AMP algorithm to form a feedforward neural network whose parameters are learned using a variant of back-propagation. The performance of LAMP (as well as AMP) is restrained to i.i.d. Gaussian matrices, hence the authors also propose an additional version, dubbed Learned Vector Approximate Message Passing (LVAMP). LVAMP is build around the Vector Approximate Message Passing (VAMP) algorithm (Schniter et al. 2016), which extends AMP's guarantees from i.i.d. Gaussian and works well with a much larger class of matrices. Another extension of AMP is the Denoising-based Approximate Message Passing algorithm (D-AMP) (Metzler et al. 2016), which is based on the denoising perspective of the AMP algorithm, that considers the non-linear operations in each iteration as a series of denoising processes. A

deep unfolded D-AMP is implemented in Metzler et al. (2017) as the Learned D-AMP (LDAMP), and also in Zhang et al. (2020c) as the AMP-Net. While both implementation are designed as CNNs, AMP-Net has an additional deblocking module (inspired by ResNet He et al. 2016) to eliminate the block-like artifacts in image reconstruction. In addition, AMP-Net also uses a sampling matrix training strategy to further improve the reconstruction performance.

Besides building upon ISTA and AMP, ADMM (Boyd et al. 2011) is another algorithm that can be used for CS reconstruction (Zhang et al. 2020a; Feng et al. 2019). The authors in Yang et al. (2017) propose ADMM-NET, a deep architecture based on CNN and inspired by the ADMM algorithm for reconstructing high-quality magnetic resonance images (MRI) from undersampled data. A more general and powerful unrolled version of the ADMM algorithm, for CS imaging of both MRI and natural images is the ADMM-CSNet (Yang et al. 2020). The ADMM-CSNet discriminatively learns the imaging model and the transform sparsity using a data driven approach, which enhances the image reconstruction accuracy and speed.

Unrolling is not the only method, instead, iterative reconstruction algorithms cam be enhanced by replacing various steps in the algorithm with a NN (Figure 5 shows the borderline case where the whole reconstruction algorithm is replaced by a NN). In Merhej et al. (2011), the correlation step of the Orthogonal Matching Pursuit (OMP) algorithm is replaced with a three-layer fully connected feed forward network trained to give an estimation of the unidentified nonzero entries of the original signal vector. The complexity overhead for training and then integrating the network in the sparse signal recovery is only justified in the case when the signal has an added structure, e.g. the zero coefficients of the sparse signal follow a certain spatial probability density function.

Compared with the conventional iterative reconstruction methods, the DL-supported algorithm unrolling brings a consistent computational speed up (the computational savings were in fact the motivation for unrolling). For example, ADMM-CSNet (Yang et al. 2020) can be about four times faster than the BM3D-AMP algorithm (Metzler et al. 2015). LISTA (Gregor and LeCun 2010) may be 20 times faster than ISTA (Beck and Teboulle 2009) after the training phase, while LDAMP (Metzler et al. 2017) can achieve a 10 times speedup, when compared to BM3D-AMP (Metzler et al. 2015). This is due to the fact that it is faster to process data through neural network layers, especially since special operations such as convolutions can be highly optimized. In addition, the number of layers in a deep network is smaller than the number of iterations required in an iterative algorithm used for CS reconstruction. Interestingly, DL approaches mimicking the unrolled algorithms can be faster even that the classic neural networks implementations aiming to replace the whole algorithm. For example, ADMM-CSNet (Yang et al. 2020) can be about twice as fast as ReconNet (Kulkarni et al. 2016), a pioneering NN reconstruction approach that does not mimic a known iterative algorithm. Nevertheless, the comparison might depend on the efficiency of the implementation of individual network layers.

The true potential of DL for signal reconstruction, however, is observed if we compare the reconstruction accuracy of DL approaches with the accuracy achieved by conventional iterative algorithms. ADMM-CSNet produces the highest recovery accuracy in terms of PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index Measure). AMP-Net (Zhang et al. 2020c) and ISTA-Net (Zhang and Ghanem 2018) improve the reconstruction accuracy over both D-AMP (Metzler et al. 2016) and a NN-approach ReconNet (Kulkarni et al. 2016). By learning parameters in each iteration, instead of keeping them fixed for the whole network, unrolled methods are able to extend the representation

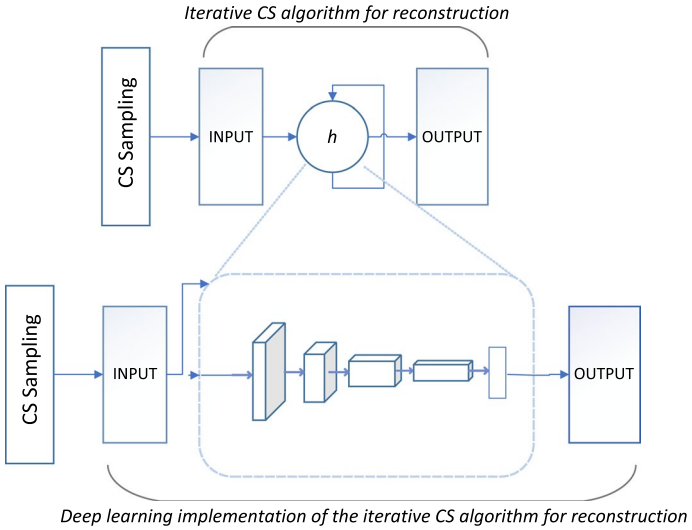

Fig. 5 CS signal reconstruction where a traditional iterative algorithm is fully (or partly) replaced using a deep neural network. The CS iterative algorithm's iteration step h, executed a number of times, is implemented using a certain neural network architecture

capacity over iterative algorithms, thus are more specifically tailored towards target applications.

Finally, if from the efficiency and performance point of view, the unrolled approach often remains superior to both iterative and neural network approaches not based on algorithm unrolling, from other perspectives, such as parameter dimensionality and generalization, unrolled approaches remain in an intermediate spaces between iterative algorithms and more general DL-based solutions (Monga et al. 2021). A summary of the most relevant methods in this category is provided in Table 2.

3.2 Deep learning for direct reconstruction

Harnessing neural networks in unrolled iterative approaches provides a certain level of intuition, yet, such intuition is not necessary for the optimization to work well. In the rest of the section we describe CS reconstruction (and sampling) approaches that were not inspired by the traditional optimization algorithms. These approaches, together with specific affordances they bring, are summarized in Table 3 (due to spatial constraints, not all approaches referred to in the table are described in the main text). Free from any constraints, most of the research we discuss here optimizes both signal reconstruction as well as signal acquisition, i.e. the sampling matrix, essentially reflecting both *approach b)* and *approach c)* in Fig. 1. This brings additional benefits, as many real-world signals, while indeed sparse when projected to a certain space, need not be sparse in the fixed domain we are observing them in—learning the sampling matrix from the data often solves this issue.

Table 2 A summary of CS methods employing algorithm unrolling

NN	Study	Domain	Iterative algorithm
AE	LISTA (Gregor and LeCun 2010)	Images	ISTA
CNN	ISTA-Net (Zhang and Ghanem 2018)	Images	ISTA
FFNN	TISTA (Ito et al. 2019)	Images	ISTA
RNN	NLISTA (Song et al. 2020)	Generic	ISTA
FFNN	LAMP (Borgerding et al. 2017)	Generic	AMP
CNN	AMP-Net (Zhang et al. 2020c)	Images	AMP
CNN	LDAMP (Metzler et al. 2017)	Images	D-AMP
CNN	ADMM-Net (Yang et al. 2017)	MRI	ADMM
CNN	ADMM-CSNet (Yang et al. 2020)	Images	ADMM
CNN	TVINet (Zhang et al. 2020b)	MRI	TV
AE	VN (Hammernik 2018)	MRI	Landweber
FFNN	NNOMP (Merhej et al. 2011)	Images	OMP

3.2.1 Autoencoder-based approaches

A bird's eye perspective on CS reveals that, with its sampling (i.e. dimensionality reduction) and reconstruction pipelines, the method closely resembles a DL autoencoder (AE). Thus, it is not surprising that some of the early forays of DL in CS utilize AEs, so that the encoding process of the AE replaces the conventional compressed sampling process, while the decoding process of the AE replaces an iterative signal reconstruction process in CS. In such an arrangement the AE brings two immediate benefits. First, based on the training data it adapts the CS sampling matrix, which need not be a random matrix any more. Second, it greatly speeds up the reconstruction process. Once trained, the AE performs signal reconstruction through a relatively modest number of DL layers, making it an attractive alternative to iterative reconstruction, even on ubiquitous computing devices, especially those embedded with GPUs.

A pioneering approach in the area of AE-based CS is presented in Mousavi et al. (2015), where Mousavi et al. propose the use of a stacked denoising autoencoder (SDAE). The SDAE is an extension of the standard AE topology, consisting of several layers of DAE where the output of each hidden layer is connected to the input of the successive hidden layer. A DAE is a type of AE that corrupts with noise the inputs to learn robust (denoised) representations of the inputs. The denoising autoencoder's main advantage is that it is robust to noise, being able to reconstruct the original signals from noise-corrupted input. One of the challenges of using SDAE is that its network consists of numerous fully-connected layers. Thus, as the signal size grows, so does the network, imposing a large computational complexity on the training algorithm and risking potential overfitting. The solution proposed in Mousavi et al. (2015) and adopted by similar approaches (Kulkarni et al. 2016; Liu et al. 2019; Pei et al. 2020) is to divide the signal into smaller blocks and then sense/reconstruct each block separately. From the reconstruction time point of view, the simulation results show that this approach beats the other methods, whereas the quality of the reconstruction does not necessarily overshadow that of other state-of-the-art recovery algorithms.

The amount of memory and processing power required by DL may prohibit CS–DL on ubiquitous computing devices. Therefore, reducing the number of DL parameters

Table 3 A summary of the DL approaches for CS direct reconstruction and their key affordances

NN	Study (domain, sensing matrix)	Affordances
AE	Mousavi et al. (2015), images, learned SAECS (Han et al. 2018b), bio signals, Gaussian SSDAE-CS (Zhang et al. 2019), images, learned DCGAN (Bora et al. 2017), images, learned (Bora et al. (2017), images, Gaussian (Iliadis et al. 2016), video, learned (Mangia et al. 2020b), ECG, learned TCSSO (Mangia 2020), ECG, learned	• Adapt the sampling matrix; • Speed up the reconstruction process; • Can also be adapted for learning more robust representations, or for learning representation that follow a certain distribution, etc.;
CNN	ReconNet Kulkarni et al. (2016), images, Gaussian (Liu et al. 2019), images, Gaussian DeepInverse (Mousavi and Baraniuk 2017), images, Gaussian (Mousavi et al. 2017), generic, learned ConvCSNet (Lu et al. 2018), image, learned KCSNet (Canh and Jeon 2018), images, learned WDLReconNet (Lu and Bo 2019), images, random (Wang et al. 2016), MRI, 2D Poisson (Schlemper et al. 2017), MRI, Cartesian	• Minimize number of parameters; • Reduce memory storage; • Increase learning speed; • Can better handle larger inputs; • Capture structure in images;
MLP/FC	Adler et al. (2016b), images, learned (Zur and Adler 2019), images, learned (Shrivastwa et al. 2018), ECoG, learned (Sun et al. 2016a), neural recordings, learned (Iliadis et al. 2018), video, random binary	• Support supervised learning; • Lead to fewer operations, compared to CNNs;
RNN	Li and Wei (2016), generic, Gaussian (Ji et al. 2019), speech, learned CRNN-MRI (Qin 2018), MRI, learned	• Good candidates for temporally correlated data; • Improve reconstruction; • Speed the reconstruction;
LSTM	LSTM-CS (Palangi et al. 2016a), images, random BLSTM-CS (Palangi et al. 2016b), images, random (Han et al. 2017), biological signals, Gaussian CSVideoNet (Xu and Ren 2016), video, learned CSNet (Zhang et al. 2021), ECG, Bernoulli	• Improve training convergence (compared to RNNs); • Can model both past and future information ; • Improve reconstruction; • Speed the reconstruction;
Res	DR2-Net (Yao 2019), images, Gaussian (Du 2019), images, learned CSNet (Shi et al. 2020), images, binary learned (Han et al. 2016), CT, - FBPConvNet (Jin et al. 2017), CT, Cartesian (Lee et al. 2017), MRI, uniform random (Han et al. 2018a), CT, MRI, radial DRL-CNN (Ouchi and Ito 2020) MRI, random (Zhao et al. 2020b), video, learned ResCNN (Kim et al. 2020b), spectroscopy, 2D filter-array	• Alleviate the vanishing gradient problem, overfitting and accuracy saturation; • Are easier to train as opposed to deep CNNs; • Provide balance between number of parameters and performance; • Accelerate training process; • Improve quality of the reconstruction; • Can be used for denoising or de-aliasing;
GAN	ReconNet(2) (Lohit et al. 2018a), images, learned DAGAN (Yang 2018), MRI, Gaussian GANCS (Mardani 2018), MRI, radial (Yu et al. 2017), MRI, Fourier encoding matrix CSGAN (Kabkab et al. 2018), images, Gaussian	• Can be tailored towards a certain task; • More useful perceptual details preserved; • Provide sharper, more realistic reconstructions;

necessary for CS is highly desired. A sparse autoencoder compressed sensing (SAECS) approach is proposed in Han et al. (2018b). The sparse autoencoder's loss function is constructed in a way that activations are penalized within a layer, resulting in fewer non-zero parameters, thus a "lighter" encoder, that is also less likely to overfit. Furthermore, combining SDAE and SAECS, a stacked sparse denoising autoencoder for CS (SSDAE CS) is proposed in Zhang et al. (2019). The proposed model, consists of an encoder sub-network which performs nonlinear measurements (unlike the conventional CS approach that involves linear measurements) on the original signal and a decoder sub-network that reconstructs the original de-noised signals by minimizing the reconstruction error between the input and the output.

Signals in the AE-compressed space, by default, need not exhibit any regularities. The AE merely ensures that the encoding-decoding process is efficient. The variational autoencoder (VAE) is trained to ensure that the latent space exhibits suitable properties enabling generative decoder behavior, which could improve the CS recovery problem, as shown in Bora et al. (2017). A novel Uncertainty autoencoder (UAE) structure is proposed by Grover and Ermon (2019). The UAE is another AE-based framework for unsupervised representation learning where the compressed measurements can be interpreted as the latent representations. Unlike the VAE, the UAE does not explicitly impose any regularization over the latent space to follow a prior distribution, but optimizes the mutual information between the input space and the latent representations, being explicitly designed to preserve as much information as possible. While not discussed in Grover and Ermon (2019) it would be interesting to examine whether UAE enables faster training with less data—a property that down the road could enable privacy-preserving on-device training in ubicomp environments.

The versatility of the autoencoder approach to CS led to its adaptation to a range of domains. Such adaptation is evident in Adler et al. (2016b), one of the early approaches geared towards image reconstruction. High dimensionality, and consequently high memory and computation load, incurred by the sheer size of images called for block-based reconstruction. This was later enhanced with a modified loss function—in Zur and Adler (2019) the authors moved from a generally-applicable mean squared error to image-specific structural similarity index measure (SSIM) as a training loss function for the autoencoder. SSIM is a widely used loss function adopted for training many recent image restoration DL models (Jun et al. 2021; Ramzi et al. 2020) because its ability to produce visually pleasing images. Domain adaptation is also evident in Iliadis et al. (2016) where Iliadis et al. developed a DL architecture for compressive sensing and reconstruction of temporal video frames. Another adaptation (Mangia et al. 2020a) was designed for the particularities of the biological signals. In the case of ECG or EEG for example, short acquisition windows are beneficial for reducing the computational complexity, the storage requirements and the latency at the encoder side. However, short windows are also out of the reach of classical CS mechanisms, because the sparsity constraint is no longer fulfilled in small sized measurements. To make the CS acquisition and recovery feasible even for short windows, the authors step away from the classical approach that directly reconstructs the input signal and propose a two stage approach: first guessing which components are non-zero in the sparse signal to recover and then computing their magnitudes.

3.2.2 Dense and convolutional networks-based approaches

Dense networks based on a standard multilayer perceptron (MLP) have also been considered a reasonable choice for CS reconstruction problem, because their ability to learn a nonlinear function that maps the compressed measurements to the the original signal, in a supervised manner, unlike the autoencoders. With the introduction of convolutional filters and pooling layers, CNNs can achieve improved performance for reduced memory storage and increased learning speed, compared to vanilla dense networks, thus, represent an attractive architectural choice especially for image reconstruction tasks. Furthermore, compared to autoencoders, these networks can often handle larger inputs, due to the reduced dimensionality of convolutional and sparse connected layers.

ReconNet (Kulkarni et al. 2016) is considered to be the first work that employs CNN for compressive sensing. Inspired by the success of the CNN-based approach for image super-resolution, the authors proposed a fully connected layer along with a CNN, which takes in CS measurements of an image as input and outputs the reconstructed image. The quality of the reconstruction is superior to those of the traditional iterative CS reconstruction algorithms. From the time complexity perspective, this approach is about three orders of magnitude faster than traditional reconstruction algorithms. One of its drawbacks is the fact that this approach uses a blocky measurement matrix, to reduce the network complexity and hence, the training time, and therefore, ReconNet does not exploit potentially strong dependencies that may exist between the reconstructions of different blocks. CombNet provides improved quality of reconstruction by using a deeper network structure and a smaller convolution core (Liu et al. 2019).

Going a step further from these block based approaches (Kulkarni et al. 2016; Liu et al. 2019), the work in Mousavi and Baraniuk (2017) introduces the first DL framework that works for images acquired with non-blocky measurement matrices. This network, named DeepInverse, is a CNN architecture, with a fully connected linear layer, whose weights are designed to implement the adjoint operator (transpose) of the measurement matrix. Including this auxiliary information into the reconstruction process simplifies the ill-posed reconstruction problem, and allows the network to reconstruct signals, without subdivision, using just four layers. DeepCodec (Mousavi et al. 2017) is a variant of DeepInverse (Mousavi and Baraniuk 2017) that instead of using random linear undersampled measurements learns the measurement matrix through a sequence of convolutional layers. ConvCSNet (Lu et al. 2018) uses one convolutional layer for the sensing part and a two branches network for reconstruction. Another approach meant to alleviate the complexity of the high dimension measurements is presented in Canh and Jeon (2018), where the sampling and the initial reconstruction are performed with convolutional Kronecker layers that decompose the large weight matrices of the convolutional layer into combinations of multiple Kronecker products of smaller matrices, thus reducing the number of parameters and the computation time.

Domain knowledge can greatly enhance signal reconstruction. In Lu and Bo (2019) a wireless deep learning reconstruction network (WDLReconNet) aiming to recover signals from compressive measurements transmitted over a WSN (Wireless Sensor Network) is proposed. To counter the effect of wireless distortions, the authors prefix a CNN with a dictionary learning-based feature enhancement layer. MRI is one of the key application areas of CS in general, thus a number of CNN-based solutions have been adapted to this domain. Schlemper et al. (2017) propose a framework for reconstructing dynamic sequences of MR images from undersampled data using a deep cascade of CNNs to accelerate the data

acquisition process. A simple CNN could show signs of overfitting, when not enough training data is available—a situation often encountered in the medical imagery field. Therefore, the authors proposed to concatenate a new CNN on the output of the previous CNN to create a DNN that iterates between intermediate de-aliasing and the data consistency reconstruction. CNNs are also used in Wang et al. (2016), where a three-layer CNN is designed and trained to define a relationship between zero filled solution and high-quality MRI.

Finally, while CNNs remain the predominant approach for CS–DL integration, standard MLP-based dense networks also have their place in the literature. Iliadis et al. use such networks for the problem of video CS reconstruction (Iliadis et al. 2018). The authors motivated their choice of not using convolutional layers to the fact that the measurement blocks used here are preventing convolutions from being effective. Different variants of MLP networks are tested in Shrivastwa et al. (2018) and shown to perform well for compressed electrocorticography (ECoG) signal reconstruction. High compression rates demonstrated in this work open way for interesting ubiquitous computing applications, such as using remotely sampled and compressed brain signals for remote prosthetic control.

3.2.3 Recurrent neural networks (RNNs) and long short-term memory networks (LSTMs)-based approaches

RNNs, to the best of our knowledge first used for CS signal reconstruction in Li and Wei (2016), found suitable ground in domains where the temporal dependencies in the data are essential for achieving a fast and accurate reconstruction. Consequently, domain-adapted versions of RNNs and LSTMs can be found mostly in video processing and speech analysis, but extend to other domains as well. In speech processing, RNNs were exploited for shaping the structural differences between voiced and unvoiced speech (Ji et al. 2019). To mitigate the noisy distortions caused by unvoiced speech, the authors build an RNN dictionary learning module that learns structured dictionaries for both voiced and unvoiced speech. These learned codebooks further contribute to improving the overall reconstruction performance of compressed speech signals. RNNs are also appropriate for processing sequences of images, particularly if the images are temporally correlated. An RNN model adapted for the image domain can be found in Qin (2018), where a convolutional recurrent neural network (CRNN) is used to improve the reconstruction accuracy and speed by exploiting the temporal dependencies in CS cardiac MRI data. By enabling the propagation of the contextual information across time frames, the CRNN architecture makes the reconstruction process more dynamic, generating less redundant representations.

LSTMs are similarly adapted to particular CS domains. In Zhang et al. (2021) a LSTM is used to extract time features from ECG signals compressed measurements, initially reconstructed with a CNN, for further improving the quality of the reconstruction. The field of distributed compressed sensing or Multiple Measurement Vectors (MMV) was also targeted in Palangi et al. (2016a) and in Palangi et al. (2016b), where an LSTM and a Bidirectional Long Short-Term Memory (BLSTM), respectively were proposed for reconstruction. The LSTM and BLSTM models are good candidates for reconstructing multiple jointly sparse vectors, because of their ability to model difficult sequences and to capture dependencies (using both past and future information, in the case of BLSTM). In the field of biological signals processing, Han et al. (2017) took advantage of the natural sparsity of the clothing pressure time sequences measurements of a region of the human body and targeted a LSTM-based CS reconstruction. In the video compression domain, Xu and Ren

(2016) employed a LSTM for combining the temporal coherence in adjacent (compressed sampled) video frames with the spatial features extracted by the CNN, for an enhanced reconstruction quality. The comparison with and without the LSTM network shows that exploiting the temporal correlations between adjacent frames can significantly enhance the CS performance of video applications in terms of the trade-off between compression rates and reconstruction quality.

Building upon these results, the field of ubiquitous computing can be further enriched with novel applications, such as those targeting smart city sensing or remote health telemonitoring with compressed electroencephalogram (EEG) signals, where learning and incorporating temporal dependencies in the CS reconstruction algorithm might improve both the reconstruction accuracy and the response time. In addition, using network models that exploit the inherent temporal structure of the signals may also reduce the number of measurements needed, which is again very important for applications using sensors with low power consumption and limited battery life.

3.2.4 Residual networks-based approaches

Increasing the depth of network by adding more layers may improve the performance of the CS reconstruction networks discussed above. However, training very deep networks poses difficulties due to the vanishing gradient problem, overfitting, and accuracy saturation. The solution to these problems came with the introduction of the residual learning. Residual networks prove to be able to alleviate this challenges, being easier to train, better optimized and having higher performances, as opposed to very deep CNN architectures. Moreover, residual learning architecture proved to be very useful not only to speed up the training process, but also for denoising and superresolution, thus improving the quality of the reconstruction.

A residual network architecture was used by Yao et al. (2019). This network (called the Deep Residual Reconstruction Network, or DR2-Net) has a fully connected layer (to obtain a preliminary reconstruction) and several residual learning blocks (to infer the residual between the ground truth image and the preliminary reconstruction). The experimental results showed that the DR2-Net outperforms other deep learning and iterative methods, being more robust for the CS measurement at higher compression rates.

Learning the sampling as well further improves the quality of the reconstruction. A jointly optimized sampling and reconstruction approach is presented in Du (2019) where Du et al. propose a similar architecture, the main difference being the use of a convolutional layer for getting the adaptive measurements. A fully convolutional neural network can deal with images of any size, breaking the limitation of fully-connected layers that are only capable of sampling fixed size images. Another approach that learns both the sampling and the reconstruction, is the one in Shi et al. (2020). The sampling network is designed to learn binary and bipolar sampling matrices tailored for easy storage and hardware implementation. Such matrices are then suitable for a range of ubicomp applications, e.g. for WSNs used for critical infrastructure health monitoring where signals are collected, compressed, and wireless transmitted by low-power and memory-limited WSN nodes.

Incorporating domain knowledge can improve the performance of the reconstructing model that is based on a residual network. In the field of computed tomography (CT), the use of residual networks was motivated mainly by the inherent presence of striking artifacts caused by the sparse projection that are difficult to remove with vanilla CNNs architectures.

Multi-scale residual learning networks based on the U-Net (Ronneberger et al. 2015) architecture aiming to remove these streaking artifacts were proposed in Han et al. (2016) and Jin et al. (2017). Aliasing artifacts are also common in CS MRI, so several approaches (Lee et al. 2017; Ouchi and Ito 2020; Han et al. 2018a) introduced residual learning to enhance the reconstruction accuracy by learning and removing the aliasing artifacts patterns. This also accelerates the training process, since learning the aliasing artifacts is easier and faster than learning to predict the original aliasing-free MR images which possess a more complex topological structure. Another issue specific to the MRI field, namely the limited data available for training was addressed by Han et al. (2018a) who propose the technique of domain adaptation (pre-training the network with CT data for MRI), that also expands the applicability of the model. Domain knowledge integration is also evident in Kim et al. (2020b) where a residual network adapted for CS spectroscopy reconstruction (ResCNN) is proposed. The main challenge in this field is to identify a sparsifying basis that would work for the great variety of spectra available. Having a residual connection between the input and the output of a CNN, ResCNN learns the spectral features and recovers the fine details for various kinds of spectra, without requiring a priori knowledge about the sparsifying basis or about the spectral features of the signal. Finally, in the field of video compressed sensing (Zhao et al. 2020b), a residual block was used for improving the recovery procedure in terms of both the quality and the speed of the reconstruction.

These promising results achieved by residual networks, in ensuring a good balance between the number of network parameters and the model performance, can open new perspectives in the ubiquitous computing domain for new applications, such as using compressed sampled images for license plate recognition directly on edge devices with limited computing resources. In light of the proliferation of IoT devices in today's smart cities, increased efforts are underway towards on edge device processing without transferring the data (Chen et al. 2020, 2021); as such, a CS–DL residual architecture has great potential in complementing these on device approaches. For example, the promising results achieved by Kim et al. (2020b) could be capitalized in the emerging field of small satellites on-board applications. Given their moderate depth and reduced number of parameters, residual CS networks can be successfully deployed in such resource-constrained environments. In addition, due to their proven ability to discriminate among various spectral signatures, residual CS networks could consequently provide real-time information in applications such as optical discrimination of vegetation species (Maimaitijiang 2020) or marine algal cultures (Deglint et al. 2019).

3.2.5 Generative adversarial networks-based approaches

Generative Adversarial Networks (GANs) can facilitate the reconstruction and synthesis of realistic images and opened the door to innovative ways to approach challenging image analysis tasks such as image denoising, reconstruction, segmentation or data simulation. Inspired by the success of GANs in computer vision tasks, a modified version of the ReconNet, with adversarial loss (in addition to the Euclidean loss) and a jointly learning approach of both the measurement matrix and the reconstruction network is proposed in Lohit et al. (2018a). In this network architecture, ReconNet acts as the generator, while another network, the discriminator, is trained to classify the input received as being a real image or a ReconNet reconstructed one. This GAN approach has sharper reconstructions, especially at higher measurement rates, compared to the original ReconNet.

GANs ability to provide realistic, high texture quality images was exploited in Mardani (2018) for CS MR image reconstruction. A deep residual network with skip connections is trained as the generator that learns to remove the aliasing artifacts by projecting it onto a low-dimensional manifold containing the desired, high-quality data. The discriminator, in the form of a CNN-based architecture, is trained to assess the projection quality, scoring one if the image is of diagnostic quality, and, zero if it contains artifacts. This network model, dubbed GANCS, scores superior results in terms of both diagnostic quality of the reconstructed images, and running time, relative to the alternative, conventional CS algorithms.

These promising results are not easy to achieve, since training two competing neural networks is a challenging task, that requires extra care for designing the model and tuning the parameters, in order to ensure the stability and the proper convergence of the model. A solution for this issues was proposed in Yu et al. (2017), where the authors use refinement learning to stabilize the training of a GAN model for MRI CS. The generator was trained to generate only the missing details in the image, which proved to reduce the complexity of the network and lead to a faster convergence. In addition, the loss function was enriched with a perceptual loss and a content loss incorporating both pixel and frequency domain information to improve the reconstructed image quality in terms of anatomical or pathological details.

The authors of Kabkab et al. (2018) introduce the task-aware GANs for CS, a model that allows optimizing the generator specifically for a desired task, be it reconstruction, classification, or super-resolution. Their solution improves the objective function by introducing the optimization of the latent code, or in other words of the compressed representation of the data, in addition to the optimizations of the generator and of the discriminator. In this manner, the compressed data obtained is more tailored towards a certain task and more beneficial for training the generator and the discriminator. Unlike other deep learning-based approaches, a GAN can emphasize what is indeed relevant for the end-user, and therefore their reconstructed images preserve more useful perceptual details. In the case of MRI, anatomical or pathological details for diagnosis, e.g., more detailed texture, sharper organ edges, better defined tumor textures and boundaries, and other factors beyond the commonly used image quality metrics (such as the Signal-to-Noise Ratio, Mean Squared Error, etc.) are elicited. Finally, this approach (Kabkab et al. 2018) also addresses the cases where no or very little non-compressed data is available, and for that, the model is trained using a combination of compressed and non-compressed training data. This requires another discriminator to be added, hence the proposed model has one generator and two discriminators—one for distinguishing between actual training data and generated data, and another for distinguishing between actual compressed training data and generated data.

Although most of the existing GAN based CS approaches addressed the MRI domain, other yet unexplored CS fields may also benefit from the advantages brought by the GANs architectures. In particular, the ability of GANs to reconstruct data with relevant features to the end-user application can be extremely valuable in ubiquitous computing applications with budgeted acquisition and reconstruction speed, where not the accuracy of the reconstruction, but the usefulness for subsequent processing prevails. One such example is the use of drones equipped with hyperspectral cameras for capturing compressed images (Oiknine et al. 2018) and using the salient information of the scene for wildfire monitoring.

4 Inferring higher-level concepts from compressive sensing with deep learning

The recovery of data acquired through compressive sensing need not always be necessary, as we might be interested, not in the original signal, but certain inferences stemming from it. For instance, we might be concerned about whether certain objects are present in an image, what kind of a modulation technique is used in a wireless electromagnetic wave, or whether a certain pathology is present in a compressed ECG signal.

Such a CS approach, without the actual signal reconstruction, is termed *compressive learning*, and was first introduced in Calderbank et al. (2009), where the authors demonstrated that learning using compressed data need not produce a substantial accuracy loss, compared to learning from the full uncompressed data. Many compressed learning approaches have been proposed since, expanding over domains such as compressed object tracking (Kwan 2020; Vargas et al. 2018), compressed hyperspectral image classification (Hahn et al. 2014), or reconstruction-free single pixel image classification (Latorre-Carmona et al. 2019). The main advantage of compressive learning is that by skipping the reconstruction phase and extracting features from compressed measurements directly, the computation complexity and the processing time get significantly reduced. From the systems point of view, additional benefits are achieved by keeping the whole inference pipeline on a single device: the data transmission costs are reduced, the inference latency decreases, and data privacy is maintained. Furthermore, compressive learning may perform well even at very high compression rates where reconstruction based approaches would fail. This allows on-device learning implementations on resource-constrained systems that otherwise would not be able to support the inference task. Finally, in some cases the reconstruction phase may introduce artifacts and errors that can distort the reconstructed signal and therefore also the inference result. In many cases, compressed samples contain most of the relevant information, and thus can be considered as a comprehensive feature representation. This points to an interesting parallel between the traditional use of the encoder part of an autoencoder as a feature extraction tool, and its use for the sampling matrix adaptation in Sect. 3.2.

The direct high-level inference from CS data using DL is depicted in Fig. 1d). Free from the need to reconstruct the signal, we can work directly in the compressed domain and harness the neural network's inherent ability to extract discriminative non-linear features, for which an intuitive explanation is not needed. Thus Lohit et al. (2016) proposed a DL approach for image classification compressive learning approach. This approach employed CNNs and random Gaussian sensing matrices (Lohit et al. 2016). Building upon this work, Adler et al. (2016a) demonstrate that by jointly learning both the sensing matrix as well as the inference pipeline, the image classification error can be reduced, which is especially evident when high compression rates are used.

4.1 High level inference and reconstruction

In certain situations, however, signal reconstruction may be desired even when high-level inference remains the main goal of the CS system. For instance, a security camera may need to detect an intruder, yet, it would be desirable to reconstruct the original signal as a potential evidence of intrusion. A joint inference-reconstruction pipeline is proposed in Xuan and Loffeld (2018), where the authors optimize the DL pipeline, so that after a jointly learned sensing matrix, the two branches—image reconstruction and image

labeling—continue their separate ways. Such a configuration is, in sum, more efficient than a solution that relies on separate pipelines for reconstruction and labeling. The work by Singhal et al. (2017) also combines the two stages in a single one and classifies compressed EEG and ECG signals at sensor nodes. Two different experiments are conducted, the first ones involves seizure detection from EEG compressed samples and the second one arrhythmia classification from ECG compressed samples. The authors show that by eliminating a separate reconstruction stage, upon which the inference would be done, the errors and artifacts are minimized and hence the results are improved. A joint construction of the two pipelines opens interesting opportunities for adaptive CS–DL deployment in heterogeneous systems. Different pipelines may have different processing and memory requirements, and the application packages could be made so that either the inference or the signal reconstruction, or both, are supported at different (edge) platforms the application is deployed to.

4.2 Upsampling the compressed data before inference

High compression rates are crucial in certain domains. Distributed video surveillance and UAV-based imaging are just two examples of applications that generate enormous volumes of data whose storage and wireless transmission is impractical. Instead, high compression rates are used, which leads to poor signal reconstruction quality. Optimizing the inference process over the compressed data, however, can provide better results compared to the case when the inference is performed after decompressing severely compressed data. A data-driven reconstruction-free CS framework for action recognition from video data is proposed in Gupta et al. (2019). This architecture consists of two separable networks: the convolutional encoder, which performs the sensing and generates undersampled measurements, and the classifier trained to classify the undersampled measurements. To ensure compatibility between the encoder and an existing DL classifier, an upsampling layer is added after the encoding part. In this manner, the customized encoding sub-network can be jointly trained with a validated classifier for better results. This approach of resizing the encoded data before inference was also explored in Bacca et al. (2020), where a CS reconstruction-free deep learning implementation for single pixel camera is proposed. Two network architectures are evaluated: one that re-projects the measurements to the original image size to preserve the image size for classification, and another that extracts features directly from the compressed measurements, without learning an image size re-projection operator. Although the first approach achieves slightly better accuracy results on average, the second one has advantages in terms of smaller number of parameters and faster computation times with results comparable with those achieved by the first approach.

4.3 Measurement rate adaptivity

A common feature of most deep learning-based compressive sensing approaches is that they work only for the measurement rate that they have been trained on and cannot be used on other measurement rates without retraining. However, real-world scenarios often impose time-varying constraints on the measurement rates (e.g. the sparsity of the data fluctuate, the memory/energy/bandwidth limitations vary, the content is dynamically changing, etc.). This is especially symptomatic for mobile solutions, where the context of usage changes with the location of a smartphone, smartwatch, or any other device a user is carrying. Thus, is of practical importance to enable dynamically adaptive measurement

rates. In practice, this could be realized via several different network models each trained for a separate measurement rate. Yet, this would incur potentially prohibitive additional storage and computation costs associated with training and storing multiple network configurations. Especially for the devices with limited resources, it is crucial to enable a single neural network to perform inference over a range of measurement rates. To circumvent these challenges, several recent studies addressed neural network model adaptation to variations in the dimensions of their inputs (Malekzadeh et al. 2021; Gilton et al. 2021).

In CS–DL, the first challenge for rate-changing scenarios, is finding the optimal measurement rate under fluctuating conditions. A preliminary approach addressing this issue was developed in Sekine and Ikada (2019) where deep learning algorithms are used to estimate the optimal compression rate according to the data sparsity. This solution manages to maximize the data transmission efficiency, being thus very suitable for edge devices. The system was tested on a Raspberry Pi 3 Model B+, using vertical acceleration data of a domestic bridge. However, this solution relies on conventional CS algorithms for sampling and reconstruction that can easily incorporate sparsity priors as their input parameters. The second challenge towards rate-adaptive CS–DL algorithms is developing neural networks that can work with different measurement rates.

When it comes to the actual adaptable network implementations, such a solution was first proposed by Lohit et al. (2018b). The authors first train the entire network for the highest desired measurement rate, and then in the second stage, all the parameters are frozen and the network is trained for the lowest measurement rate. Finally, in the third stage, the network is again optimized over a subset of parameters corresponding to adding an additional row at a time to the measurement matrix with the rest of parameters frozen. In the end, any subset of consecutive rows of the measurement matrix represents a valid measurement matrix corresponding to a different measurement rate in the range between the highest and the lowest specified measurement rates. To map the size-varying compressed inputs to the same inference network, an additional conversion layer is needed that maps the inputs back in the original space by applying the pseudo-inverse of the measurement matrix. This approach was tested in an object tracking scenario in video sequences and the measurement rate was adapted based on the content evaluation among successive frames. The adaptivity in the context of compressed learning has also been addressed by Xu et al. (2020). In this work, CS measurement vectors of different lengths, corresponding to different measurement rates, and randomly shuffled, are provided as inputs to the neural network with a fixed input layer size and the network is trained on them. For handling the size mismatch between the size-varying CS measurement vectors and input layer two approaches are explored: one that zero-pads the measurement vectors so they are all of the maximum length, which is also the dimension of the input layer of network; and the other that projects back into the original space dimension the measurements, to get a pseudo-inverse of the measurement matrix, like in the previous work (Lohit et al. 2018b). CS rate adaptivity was also addressed in Machidon and Pejović (2022), where a zero padding strategy was also proposed and combined with context awareness to intelligently adapt the sampling rate according to the nature of the signal at the input.

4.4 Preserving privacy

Compressive learning can also be performed over a distributed system, such as a cloud computing network, raising concerns regarding the privacy protection of sensitive data. While compressive sensing can be seen as a form of data encryption, the compressed

data is vulnerable to privacy attack, since the compressed sensing matrix could easily be decoded, by using a brute force attack of trial and error method, and that would expose the original data. Thapaliya et al. (2020) build a privacy-preserving predictive compressed learning model based on using a strong transformation matrix, instead of the attack vulnerable classic compression matrix. Unlike other privacy preserving approaches, that are based on hiding the patterns existing in the data, the proposed approach perturbs the data using patterns that are not present in the data. In this manner, the data is perturbed enough to be robust to privacy attacks but the predictive accuracy of the model, trained to recognize the patterns of the data is still preserved.

5 Towards deep compressed sensing systems

The CS–DL approach has created hopes for the implementation of many practical ubiquitous computing applications, nevertheless, to the best of our knowledge, a large majority of the existing CS–DL approaches use pre-collected data (mostly images) and run on desktop computers or servers. In this section we first discuss real-world limitations that CS–DL is facing, especially in the ubicomp domain, and potential solutions addressing some of these limitations. We then, in selected application domains, present a few existing research efforts tackling particular domain challenges. Finally, we present a few interesting opportunities for future research in deep compressed sensing systems.

5.1 General challenges of deep learning and compressive sensing

Both building blocks of CS–DL carry their unique challenges. For compressive sensing, assessing an unknown signal's sparsity is one of the major practical hurdles, as the number of samples needed for signal reconstruction directly depends on the signal's sparsity. Currently, having access to fine-grained sampled signal that will then be transformed to its sparse basis represents the only means of reliably addressing the issue. This, however, defies the purpose of sub-Nyquist sampling that CS is based on. The problem is even more difficult in case the basis in which the signal is sparse is unknown. Significant reconstruction algorithm improvements brought by deep learning may offer a solution, as one could use large amounts of data and train various pipelines (in accordance to Fig. 1b or c) to reconstruct the signal from varying amounts of undersampled measurements, finally selecting the one that leads to satisfactory results.

Deep learning comes with its own challenges, such as a need for the large amount of data, occasional difficulties with training convergence, or high computational demands of model training[4]. A challenge that is likely to be increasingly pronounced in the area of CS–DL is the lack of well-defined standard architectures for different problem subdomains. While years of research, common datasets, and open competitions have lead to a clear identification of the most suitable neural network architectures for natural language processing (e.g. BERT) or image segmentation (e.g. U-nets), it is unclear which architectures are the most suitable for, say, wireless signal reconstruction from sparse spectrum measurements. We hope that our survey represents a good starting point for further consolidation of CS–DL architecture search efforts.

[4] We refer a reader to Aggarwal (2018) for additional details on DL challenges.

The integration of CS and DL introduces new challenges. For instance, adversarial learning can be harnessed to tune neural network models so that the *reconstruction* from CS samples leads to more realistic, high texture quality images (see Sect. 3). While the diagnostic quality of such images (i.e. when used in the medical domain) often outperforms vanilla reconstruction approaches, caution should be exercised when images are used for the exploratory purposes. Due to the way adversarial training tunes the network, unexpected findings might be obscured by the reconstruction process that favors realistic appearance of the image.

Reliability of deep learning-based *inference* is an active research topic (Jiang et al. 2018; Guo et al. 2017). Soon after the initial excitement brought by the success of deep learning models for computer vision, examples (often minimial tweaks) have been constructed to fool popular models into misclassifying images (Kurakin et al. 2018). Recent theoretical advances improve our understanding of DL model robustness for certain classes of models and in certain situations. Yet, it is unknown how the stochastic nature of compressive sensing, and especially the variation of the amount of sampled data brought by varying sampling rates, could impact the reliability of the CS–DL inference pipeline.

5.2 Challenges of moving to the edge

Energy is the most critical resource on ubiquitous computing devices. Portability implies that devices often run on limited capacity batteries with few opportunities for charging. Both sensor sampling and deep learning incur a significant energy cost. Limited *communication* capacity is another constraint. Wirelessly connected devices have to cope with intermittent and varying quality links. For CS–DL applications this calls for a careful adaptation of the sampling rate, as higher sampling rates produce more data, which might need to be processed remotely, while lower sampling rates may reduce the quality of the reconstructed signal. *Changing the sampling rate* in the case of CS–DL based algorithms is another challenge, since a neural network is usually trained for one measurement rate, and unlike the traditional CS reconstruction algorithms, only works for that sampling rate. Difficulties also arise when trying to devise CS inspired sensor control strategies, since most sensors only support by default uniform sampling strategies. *Non-uniform sampling*, unless natively supported by the OS (Operating System), is not trivial to achieve since it implies a cyclic switching of the sampling periods of the sensor and must consider feedback control, tasks scheduling, synchronization, input saturation, etc. Currently there is no native support for non-uniform sampling across the OSs used in ubiquitous computing, and research in this field targets custom-designed controllers (Chang et al. 2018). *Computational and storage* limitations, especially in terms of the GPU support for DL models on embedded architectures and the ability to store large deep networks in memory, are characteristic for ubiquitous computing devices. Moreover, from the software side, DL often harnesses dedicated libraries, which can be difficult to migrate to such devices.

The above challenges did not prevent researchers from proposing practical CS–DL solutions. For instance, Shen (2018) proposed incorporating the theory of compressive sensing at the input layer of a CNN model to reduce the resources consumption for IoT applications, but the authors offer no actual evaluation of their model on resource constrained devices. Lee et al. (2019) developed a joint transmission-recognition CS–DL framework with low complexity for IoT devices to effectively transmit data to a server for recognition. Nevertheless, no actual implementation using IoT devices is provided. Sun et al. (2016a) present a deep compressed sensing framework for wireless neural recording, with potential

applications for real-time wireless neural recording and for low-power wireless telemonitoring of physiological signals. However, the framework is only assessed from the time and accuracy point of view, without any energy/power usage evaluation. Related to CS–DL efforts are solutions proposed for resource-efficient DL. To enable models running on edge devices, efforts are currently being made towards devising optimization techniques which aim to trim down the network complexity and reduce the redundancy without significant degeneration in performance. Such neural networks optimization techniques include: quantization (Gupta et al. 2015), low-rank compression (Novikov et al. 2015), pruning (Han et al. 2015), slimmable neural networks (Yu et al. 2018), and early exiting (Scardapane et al. 2020), to name a few. Crucial from the mobile systems perspective is the fact that these techniques can also be used in a dynamic context and by taking advantage of the variations in the input's complexity for example, important resources can be saved with minimal impact on the accuracy (Laskaridis et al. 2020).

5.3 Lessons from domain-specific CS–DL

Wearable computing devices enable in-situ sampling of *physiological signals*, such as breathing and heartbeat activity, skin temperature, skin conductance, and others. Recent advancements in the sampling and processing of ECG and EEG signals show that *CS–DL methods can be used to lower the power consumption*, with respect to the classical sampling approach, and to enable real-time signal reconstruction or high level inferences in cases where such execution was infeasible with classical sampling methods. For example, Shrivastwa et al. (2018) use an MLP network for ECoG (Electrocorticography) signals compression and reconstruction. In Singhal et al. (2017), the authors go a step further and directly classify ECG and EEG signals in their compressive measurements space, achieving satisfying results with minimal computation, by skipping the reconstruction phase. Mangia et al. (2020a) use support identification through DNN-based oracles, to first guess which components are non-zero in the sparse signal to recover, and then compute their magnitudes, thus decreasing the complexity of the computation.

Wireless connectivity is a defining characteristic of ubiquitous computing. In the field of WSNs, CS–DL based techniques are motivated by not only the sparsity of the signals, but also by the requirement of efficiency in processing in terms of energy consumption and *communication bandwidth utilization*. Moreover, the compressive measurements are transmitted over wireless channels, so *the impact of channel noise and fading on reconstruction are important factors to be taken into consideration*. Some of the most important contributions in the field of CS–DL based methods for WSN are the WDLReconNet, and Fast-WDLReconNet networks (Lu and Bo 2019), validated for the transmission of remote sensing images. Interestingly, the number of parameters in the denoising layer accounts for about 94% of the total number of parameters in the proposed CNN, underlying the impact of the noise on the transmitted data. Energy efficiency is another major aspect in WSN and quantization proved to be able to ensure an efficient wireless transmission (Sun et al. 2016a).

Magnetic resonance imaging (MRI) has revolutionized medical diagnosis in the late 20^{th} century. Nevertheless, conducting MRI scans requires significant time when a subject needs to be immobilized. CS–DL has already proved to be able to reduce the scanning time by reducing the number of samples while simultaneously improving the image

reconstruction quality. Some of the CS–DL MRI methods, use deep neural networks to learn the traditional optimization algorithms, by unrolling them (Sun et al. 2016b; Yang et al. 2017; Zhang et al. 2020b; Ramzi et al. 2020). Another category of methods uses deep networks to mitigate noise and aliasing artifacts in the MRI reconstruction. SDAE (Majumdar 2015), CNNs (Wang et al. 2016; Jun et al. 2021), residual networks (Lee et al. 2017; Han et al. 2018a; Sun et al. 2018; Ouchi and Ito 2020), or GANs (Mardani et al. 2017; Yu et al. 2017) were successfully validated as suitable architectures for CS–DL MRI and have all shown great potential, outperforming conventional CS techniques for under sampled MRI. The main advantage of CS–DL methods in MRI lies in the capability of a DNN to capture and make use of the patterns learned within the data in both image and frequency domain, to improve the quality of the reconstruction which is of high importance for medical diagnosis. In this process of reconstruction, the deep learning-based CS solutions go beyond the standard metrics that are usually used for mathematically evaluating the quality of an image, and put more emphasis on the anatomical or pathological details of an image. Also, by shifting the computational efforts to an offline training stage, CS–DL algorithms for MRI are able to provide a fast reconstruction, up to real-time, which is crucial in clinical practice. However, there are also concerns regarding use of deep learning models for image reconstruction in critical domains, such as MRI, because of the hallucinatory effects (mimicking normal structures that are either absent or abnormal), which are sometimes introduced in the reconstructed images, as shown at the 2020 fastMRI challenge (Muckley 2021).

Despite not being directly related to ubiquitous computing, advances in using CS–DL for MRI uncover the importance of *going beyond signal sparsity* and using other, expected, structure of the signal for practical reconstruction. In addition, CS–DL approaches for MRI indicate that signal reconstruction *quality should be measured from an end-to-end perspective*: the quality of the reconstruction is only good if the result is properly interpreted by a human user. With ubiquitous devices being closely integrated into human everyday environments, it is important that sensing results' usability gets precedence over the simple mathematical formalization of the reconstruction error.

5.4 From challenges to opportunities

Distributed computing A good place for using CS–DL approaches is in the field of distributed computing; instead of performing the inference either solely on edge devices or exclusively in the cloud, distributed computing proposes an alternative approach based on splitting the computation between edge devices and the cloud. Partitioning data between different devices implies using an efficient compression technique to minimize the offload transmission overhead. Yao et al. (2020) integrated the compressive sensing theory with deep neural networks, for providing offloading functionality for various applications. This system, called Deep Compressive Offloading or DeepCOD, includes a lightweight encoder on the mobile side to compress the to-be-transferred data and a decoder on the edge server side to reconstruct the transferred data. The data for offloading is encoded on the local device and decoded on the edge server, trading edge computing resources for data transmission time, and thus significantly reducing the offloading latency with minimal accuracy loss.

The most important challenge related to computation offloading is to decide whether, what, and how to offload, thus finding the best splitting point with the least latency among all possible neural network partitions candidates. DeepCOD addresses this issue

by using a performance predictor that estimates the execution time of the NN operations on the local device and on the edge server, and a runtime partition decision maker to find the optimal partition point (from the latency perspective) for offloading. This system was implemented on Android mobile devices and a Linux edge server with GPUs and reduced the offloading latency by a factor of 2 to 35 with at most 1% accuracy loss under various mobile-edge-network configurations. Adding accuracy-variable distributed NN execution, for instance in the form of early NN exiting, as proposed by SPINN (Laskaridis et al. 2020), would yield an interesting compression-splitting-approximation optimization space.

Finally, new distributed computing opportunities can also arise by intertwining compressive sensing with federated learning (Yang et al. 2019). In a standard federated learning scenario, multiple client devices collaboratively train a model, with local data. After decentralized local training on edge devices, the network parameters, such as weights or gradients are exchanged, which can cause communication bottlenecks and delays. A CS–DL approach can be used to efficiently encode the network parameters on the edge device and decode them on the server side. In this manner, only the compressed version of the gradient, for example, needs to be shared, which can reduce the communication bandwidth.

Given the increasingly crucial role that NN play in IoT systems applications, further implementations that combine CS with DL for distributed computing are likely to offer solutions in domains such as task distribution in a wireless network, mobile edge deep learning, and satellite-terrestrial computation partitioning.

Heterogeneous architectures CS–DL is ideally positioned to efficiently utilize heterogeneous architectures of today's ubiquitous computing landscape, such as the ARM big.LITTLE architecture (2012), which integrate slower power-efficient processor cores with faster power-hungry cores. By mapping an application to the optimal cores, considering the performance demands and power availability, important power savings can be achieved. Nevertheless, matching dynamic computational requirements with the underlying hardware capabilities is not easy. The DeepX framework (Lane et al. 2016) dynamically decomposes a neural network architecture into segments that can each be executed across different processors to maximize energy-efficiency and execution time. DeepSense (Huynh et al. 2016) leverages the architecture of mobile GPU devices and implies optimization techniques for optimal offloading a neural network' layers and operations on on the CPU/GPU memory and processors to achieve the best accuracy-latency trade-off.

With its adaptability, afforded by sampling rate adjustment and NN compression, CS–DL can be tuned to maximally utilize heterogeneous hardware. An adaptable CS–DL pipeline was explored in Shrivastwa et al. (2018). The authors aimed at porting a deep neural network for ECoG signals compression and reconstruction to a lightweight device and explores three architectural options: using a greedy algorithm (Orthogonal Matching Pursuit), signal compression and reconstruction using a MLP with all layers implemented in the FPGA logic, and finally a heterogeneous architecture consisting of an ARM CPU and FPGA fabric, with just a single layer of the NN being deployed in the FPGA re-configurable logic. Measurements demonstrate that the third, heterogeneous architecture, stands out as the most efficient, since it requires significant less multipliers, and thus has lower overhead compared to implementing the full NN in the FPGA. This system was realized using a Zynq processing system (ARM core) in Zedboard, and opens the door for future explorations of efficient task mapping of CS–DL implementations on heterogeneous architectures.

Perhaps the most promising avenue for research lies in energy-efficient task scheduling of a CS–DL pipeline on a mobile device equipped with a heterogeneous hardware architecture and hardware-software codesign for CS–DL. The scheduling would address the optimal task to processor assignment for achieving minimum energy consumption, which is especially important as we expect a range of advanced mobile sensing functionalities, such as speech recognition and live video processing, from our battery powered devices. The hardware-software codesign would ensure that sensors are built with compressive sensing in mind, while the processing pipeline matches the needs of the CS–DL algorithms. With respect to the latter, FPGAs stand out as likely candidate for initial implementations, due to the processing capabilities—flexibility balance they afford.

6 Conclusions

Key takeaway ideas

- CS–DL methods exhibit consistent speed-up, often being two orders of magnitude faster than the traditional CS algorithms, thus allowing real-time ubiquitous computing applications.
- Especially at the very aggressive undersampling rates often required by resource-constrained devices, the CS–DL methods are capable of better reconstructions than most of the classical methods.
- Data-driven measurement matrix does not only improve CS reconstruction/inference results, but is also more suitable for on-device storage compared to conventionally used random measurement matrices.
- The trade-off between model performance and the number of network parameters in CS–DL can be addressed using residual blocks.
- Training CS–DL pipelines requires significant computing and data resources, rendering on-device training impractical for a range of device; this issue could be alleviated with transfer and federated learning.
- New opportunities arise for distributed CS–DL computing, where a new balance can be struck between on-device sensing, partial inference, and compression, and partitioning between edge devices and the cloud; data transmission overhead, energy use, and inference delay can be optimized in this process.

The move from centralized storage and processing towards distributed and edge computing indicates that the intelligence that is expected from future ubiquitous computing environments needs to be realized as close to the physical world as possible. Consequently, sensing and learning from the collected data need to be implemented directly on the ubiquitous sensing devices, and with the support for adaptive, dynamic distributed processing. Reduced rate sampling enabled by compressive sensing (CS) represents a viable solution enabling the reduction of the amount of generated sensing data, yet CS alone does not solve the issue of complex processing that may be overwhelming for ubicomp devices' limited computational resources. Deep learning (DL) naturally complements CS in the ubicomp domain, as it reduces the computational complexity of signal reconstruction and enables full sensing-learning pipelines to be implemented.

Despite its potential, the CS–DL area remains only sporadically explored. In this survey we identified the current trends in the CS–DL area and reviewed some of the most significant recent efforts. We systematically examined how DL can be used to speed up CS signal reconstruction by alleviating the need for iterative algorithms. Furthermore, classic CS methods were not designed to go beyond sparsity and exploit structure present in the data. DL enables for the sampling matrix to be designed according to the hidden data structure that can further be exploited in the reconstruction phase. The trade-off between model performance and the number of network parameters represents a major issue in CS–DL. It has been shown that deeper network architectures can result in better network performance, yet increasing model complexity requires more intensive computational and memory requirements. Residual blocks represent a viable solution for addressing this trade-off (Yao 2019; Du 2019). Regarding the compression rate, studies (Kulkarni et al. 2016; Schlemper et al. 2017; Yao 2019; Shrivastwa 2020) showed that at very aggressive undersampling rates, the DL based methods are capable of better reconstructions than most of the classical methods. For example, the ReconNet network outperforms other methods by large margins at measurement rates of up to 0.01. Finally, one of the drawbacks of accurately reconstructing signals from few measurements using DL, is the high requirements in terms of time and data for training. Transfer learning might be a solution for this issue, as shown in Han et al. (2018a).

Although compressive sensing is a relatively new field, being around for less than two decades, with deep learning being an even newer addition, CS–DL is characterized by a burgeoning community that produces a growing body of freely available online educational resources are available. A broader collection of resources ranging from conference or journal papers and tutorials to blogs, software tools and video talks can be found at http://dsp.rice.edu/cs/). In addition, novel ideas and methods in this area are often accompanied by free and open-source code of the implementations. A useful repository containing a collection of reproducible Deep Compressive Sensing source code can be found at https://github.com/ngcthuong/Reproducible-Deep-Compressive-Sensing.

Only recently, conventional CS methods have begun to be integrated in commercial products, e.g., Compressed Sensing (Siemens) in 2016, Compressed SENSE (Philips) in 2014, and HyperSense (GE) in 2016, all three for CS MRI. The maturity level of CS–DL methods is much lower than that of conventional CS methods, and to the best of our knowledge no commercial/industry products that use CS–DL methods have yet been marketed. However, due to the promising potential that CS–DL showed in supporting various commercial applications, during the following years CS–DL will come of age and the challenges will shift from proving the concept towards integrating it into commercial products. Already, Facebook developed a DL-based faster MRI system that is currently undergoing a study in collaboration with market-leading MRI scanner vendors Siemens, General Electric, and Philips Healthcare (Marks 2021).

In this survey we presented mostly academic works at the intersection of CS and DL, aiming to provide a valuable resource for future researchers and practitioners in this domain. Furthermore, the survey aims to attract new audience to CS–DL, primarily ubiquitous systems researchers. Such expansion is crucial, as challenges identified in this manuscript, including the realization of distributed CS–DL on heterogeneous architectures and with support for dynamically adaptive sampling rates need to be addressed in order to ensure the further proliferation of sensing systems' intelligence.

Acknowledgements The research presented in this paper was funded by projects "Bringing Resource Efficiency to Smartphones with Approximate Computing" (N2-0136), "Context-Aware On-Device Approximate

Computing" (J2-3047), and by the Slovenian Research Agency research core funding No. P2-0098 and P2-0426.

Funding Funding was provided by Javna Agencija za Raziskovalno Dejavnost RS (N2-0136, J2-3047, P2-0098, P2-0426).

Data availibility Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

Ablin P, Moreau T, Massias M, Gramfort A (2019) Learning step sizes for unfolded sparse coding. In: Advances in neural information processing system, pp 13100–13110

Adler A, Elad M, Zibulevsky, M (2016a) Compressed learning: a deep neural network approach. arXiv:1610.09615

Adler A, Boublil D, Elad M, Zibulevsky M (2016b) A deep learning approach to block-based compressed sensing of images. arXiv:1606.01519

Aggarwal CC et al (2018) Neural networks and deep learning, vol 10. Springer, New York, pp 978–983

Al-Azawi MKM, Gaze AM (2017) Combined speech compression and encryption using chaotic compressive sensing with large key size. IET Signal Proc 12(2):214–218

Bacca J, Galvis L, Arguello H (2020) Coupled deep learning coded aperture design for compressive image classification. Opt Express 28(6):8528–8540

Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imag Sci 2(1):183–202

Bora A, Jalal A, Price E, Dimakis AG (2017) Compressed sensing using generative models. In: International conference on machine learning, pp 537–546

Borgerding M, Schniter P, Rangan S (2017) AMP-inspired deep networks for sparse linear inverse problems. IEEE Trans Signal Process 65(16):4293–4308

Boyd S, Parikh N, Chu E (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc, South Holland

Calderbank R, Jafarpour S, Schapire R (2009) Compressed learning: universal sparse dimensionality reduction and learning in the measurement domain. arXiv preprint

Candes EJ, Romberg J (2006) Quantitative robust uncertainty principles and optimally sparse decompositions. Found Comput Math 6(2):227–254

Candes EJ, Romberg JK, Tao T (2006) Stable signal recovery from incomplete and inaccurate measurements. Commun Pure Appl Math 59(8):1207–1223

Candès EJ, Romberg J, Tao T (2006b) Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans Inf Theory 52(2):489–509

Canh TN, Jeon B (2018) Deep learning-based Kronecker compressive imaging. In: IEEE international conference on consumer electronic-Asia (ICCE-A), pp 1–4

Chang W, Goswami D, Chakraborty S, Hamann A (2018) OS-aware automotive controller design using non-uniform sampling. ACM Trans Cyber-Phys Syst 2(4):1–22

Chen Y, Yang T, Li C, Zhang Y (2020) A binarized segmented resnet based on edge computing for re-identification. Sensors 20(23):6902

Chen C, Liu B, Wan S, Qiao P, Pei Q (2021) An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. IEEE Trans Intell Transp Syst 22(3):1840–1852

Cheng Y, Wang D, Zhou P, Zhang T (2017) A survey of model compression and acceleration for deep neural networks. arXiv:1710.09282

Chiew M, Graedel NN, Miller KL (2018) Recovering task fMRI signals from highly under-sampled data with low-rank and temporal subspace constraints. Neuroimage 174:97–110
Choudhary T, Mishra V, Goswami A, Sarangapani J (2020) A comprehensive survey on model compression and acceleration. Artif Intell Rev 1–43
Daubechies I, Defrise M, De Mol C (2004) An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun Pure Appl Math 57(11):1413–1457
Deglint JL, Jin C, Wong A (2019) Investigating the automatic classification of algae using the spectral and morphological characteristics via deep residual learning. In: International conference on image analysis and recognition, pp 269–280
Djelouat H, Amira A, Bensaali F (2018) Compressive sensing-based IoT applications: a review. J Sens Actuator Netw 7(4):45
Donoho DL (2006) Compressed sensing. IEEE Trans Inf Theory 52(4):1289–1306
Donoho DL, Maleki A, Montanari A (2009) Message-passing algorithms for compressed sensing. Proc Natl Acad Sci 106(45):18914–18919
Du J et al (2019) Fully convolutional measurement network for compressive sensing image reconstruction. Neurocomputing 328:105–112
Eldar YC, Kutyniok G (2012) Compressed sensing: theory and applications. Cambridge University Press, Cambridge
Fan Y-R, Buccini A, Donatelli M, Huang T-Z (2019) A non-convex regularization approach for compressive sensing. Adv Comput Math 45(2):563–588
Feng L, Sun H, Zhu J (2019) Robust image compressive sensing based on half-quadratic function and weighted Schatten-p norm. Inf Sci 477:265–280
Gilton D, Ongie G, Willett R (2021) Model adaptation for inverse problems in imaging. IEEE Trans Comput Imaging 7:661–674
Gregor K, LeCun Y (2010) Learning fast approximations of sparse coding. In: Proceedings of the 27th international conference on machine learning, pp 399–406
Grover A, Ermon S (2019) Uncertainty autoencoders: learning compressed representations via variational information maximization. In: The 22nd international conference on artificial intelligence and statistics, pp 2514–2524
Guo C, Pleiss G, Sun Y, Weinberger KQ (2017) On calibration of modern neural networks. In: International conference on machine learning, pp 1321–1330
Gupta S, Agrawal A, Gopalakrishnan K, Narayanan P (2015) Deep learning with limited numerical precision. In: International conference on machine learning, pp 1737–1746
Gupta R, Anand P, Kaushik V, Chaudhury S, Lall B (2019) Data driven sensing for action recognition using deep convolutional neural networks. In: International conference on pattern recognition and machine intelligence, pp 250–259
Gurve D, Delisle-Rodriguez D, Bastos-Filho T, Krishnan S (2020) Trends in compressive sensing for EEG signal processing applications. Sensors 20(13):3703
Hahn J, Rosenkranz S, Zoubir AM (2014) Adaptive compressed classification for hyperspectral imagery. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 1020–1024
Hammernik K et al (2018) Learning a variational network for reconstruction of accelerated MRI data. Magn Reson Med 79(6):3055–3071
Han S, Pool J, Tran J, Dally WJ (2015) Learning both weights and connections for efficient neural networks. arXiv:1506.02626
Han YS, Yoo J, Ye JC (2016) Deep residual learning for compressed sensing CT reconstruction via persistent homology analysis. arXiv:1611.06391
Han T, Hao K, Ding Y, Tang X (2017) A new multilayer LSTM method of reconstruction for compressed sensing in acquiring human pressure data. In: 2017 11th Asian control conference (ASCC) 2001–2006
Han Y et al (2018a) Deep learning with domain adaptation for accelerated projection-reconstruction MR. Magn Reson Med 80(3):1189–1205
Han T, Hao K, Ding Y, Tang X (2018b) A sparse autoencoder compressed sensing method for acquiring the pressure array information of clothing. Neurocomputing 275:1500–1510
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
Huang X, Cui B (2019) A neural dynamic system for solving convex nonlinear optimization problems with hybrid constraints. Neural Comput Appl 31(10):6027–6038

Huynh LN, Balan RK, Lee Y (2016) Deepsense: a GPU-based deep convolutional neural network framework on commodity mobile devices. In: Proceedings of the 2016 workshop on wearable systems and applications, pp 25–30
Iliadis M, Spinoulas L, Katsaggelos AK (2016) Deepbinarymask: learning a binary mask for video compressive sensing. arXiv:1607.03343
Iliadis M, Spinoulas L, Katsaggelos AK (2018) Deep fully-connected networks for video compressive sensing. Digit Signal Process 72:9–18
Ito D, Takabe S, Wadayama T (2019) Trainable ISTA for sparse signal recovery. IEEE Trans Signal Process 67(12):3113–3125
Ji Y, Zhu W-P, Champagne B (2019) Recurrent neural network-based dictionary learning for compressive speech sensing. Circuit Syst Signal Process 38(8):3616–3643
Jiang H, Kim B, Guan MY, Gupta M (2018) To trust or not to trust a classifier. In: Proceedings of the 32nd international conference on neural information processing systems, pp 5546–5557
Jin KH, McCann MT, Froustey E, Unser M (2017) Deep convolutional neural network for inverse problems in imaging. IEEE Trans Image Process 26(9):4509–4522
Jun Y, Shin H, Eo T, Hwang D (2021) Joint deep model-based MR image and coil sensitivity reconstruction network (joint-ICNet) for fast MRI, pp 5270–5279
Kabkab M, Samangouei P, Chellappa R (2018) Task-aware compressed sensing with generative adversarial networks. In: Thirty-second AAAI conference on artificial intelligence
Khosravy M, Dey N, Duque CA (2020) Compressive sensing in healthcare. Elsevier, New York
Kim D, Park D (2020) Element-wise adaptive thresholds for learned iterative shrinkage thresholding algorithms. IEEE Access 8:45874–45886
Kim Y, Park J, Kim H (2020a) Signal-processing framework for ultrasound compressed sensing data: envelope detection and spectral analysis. Appl Sci 10(19):6956
Kim C, Park D, Lee H-N (2020b) Compressive sensing spectroscopy using a residual convolutional neural network. Sensors 20(3):594
Kruizinga P et al (2017) Compressive 3D ultrasound imaging using a single sensor. Sci Adv 3(12):e1701423
Kulkarni K, Lohit S, Turaga P, Kerviche R, Ashok A (2016) Reconnet: non-iterative reconstruction of images from compressively sensed measurements. Proc IEEE Conf Comput Vis Pattern Recognit 449–458
Kurakin A, Goodfellow IJ, Bengio S (2018) Adversarial examples in the physical world. Chapman and Hall, Boca Raton, pp 99–112
Kwan C et al (2020) Detection and confirmation of multiple human targets using pixel-wise code aperture measurements. J Imaging 6(6):40
Lane ND et al (2016) Deepx: a software accelerator for low-power deep learning inference on mobile devices. In: 2016 15th ACM/IEEE international conference on information processing in sensor networks (IPSN), pp 1–12
Laskaridis S, Venieris SI, Almeida M, Leontiadis I, Lane ND (2020) SPINN: synergistic progressive inference of neural networks over device and cloud. In: Proceedings of the 26th annual international conference on mobile computing and networking, pp 1–15
Latorre-Carmona P, Traver VJ, Sánchez JS, Tajahuerce E (2019) Online reconstruction-free single-pixel image classification. Image Vis Comput 86:28–37
Lee D, Yoo J, Ye JC (2017) Deep residual learning for compressed sensing MRI. In: 2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017), pp 15–18
Lee C-H, Lin J-W, Chen P-H, Chang Y-C (2019) Deep learning-constructed joint transmission-recognition for internet of things. IEEE Access 7:76547–76561
Li S (2020) Tensorflow lite: on-device machine learning framework. J Comput Res Dev 57(9):1839
Li Y-M, Wei D (2016) Signal reconstruction of compressed sensing based on recurrent neural networks. Optik 127(10):4473–4477
Li X et al (2020) Deep residual network for highly accelerated fMRI reconstruction using variable density spiral trajectory. Neurocomputing 398:338–346
Liu J, Huang K, Zhang G (2017) An efficient distributed compressed sensing algorithm for decentralized sensor network. Sensors 17(4):907
Liu Y, Liu S, Li C, Yang D (2019) Compressed sensing image reconstruction based on convolutional neural network. Int J Comput Intell Syst 12(2):873–880
Lohit S, Kulkarni K, Turaga P, Wang J, Sankaranarayanan AC (2015) Reconstruction-free inference on compressive measurements. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 16–24

Lohit S, Kulkarni K, Turaga P (2016) Direct inference on compressive measurements using convolutional neural networks. In: 2016 IEEE international conference on image processing (ICIP), pp 1913–1917
Lohit S, Kulkarni K, Kerviche R, Turaga P, Ashok A (2018a) Convolutional neural networks for noniterative reconstruction of compressively sensed images. IEEE Trans Comput Imaging 4(3):326–340
Lohit S, Singh R, Kulkarni K, Turaga P (2018b) Rate-adaptive neural networks for spatial multiplexers. arXiv:1809.02850
Ltd A (2012) Arm big.LITTLE technology.howpublished. https://www.arm.com/why-arm/technologies/big-little. Accessed 6 Sept 2021
Lu H, Bo L (2019) Wdlreconnet: compressive sensing reconstruction with deep learning over wireless fading channels. IEEE Access 7:24440–24451
Lu X, Dong W, Wang P, Shi G, Xie X (2018) ConvCSNet: a convolutional compressive sensing framework based on deep learning. arXiv:1801.10342
Ma T et al (2017) The extraction of motion-onset VEP BCI features based on deep learning and compressed sensing. J Neurosci Methods 275:80–92
Machidon A, Pejović V (2022) Enabling resource-efficient edge intelligence with compressive sensing-based deep learning. In: Proceedings of the 19th ACM international conference on computing frontiers, pp 141–149
Maimaitijiang M et al (2020) Soybean yield prediction from UAV using multimodal data fusion and deep learning. Remote Sens Environ 237:111599
Majumdar A (2015) Real-time dynamic MRI reconstruction using stacked denoising autoencoder. arXiv:1503.06383
Malekzadeh M, Clegg R, Cavallaro A, Haddadi H (2021) Dana: dimension-adaptive neural architecture for multivariate sensor data. Proc ACM Interact Mobile Wearable Ubiquitous Technol 5(3):1–27
Mangia M et al (2020a) Deep neural oracles for short-window optimized compressed sensing of biosignals. IEEE Trans Biomed Circuits Syst 14(3):545–557
Mangia M et al (2020b) Low-power ECG acquisition by compressed sensing with deep neural oracles. In: 2020 2nd IEEE international conference on artificial intelligence circuits and systems (AICAS), pp 158–162
Mardani M et al (2017) Deep generative adversarial networks for compressed sensing automates MRI. arXiv:1706.00051
Mardani M et al (2018) Deep generative adversarial neural networks for compressive sensing MRI. IEEE Trans Med Imaging 38(1):167–179
Marks P (2021) Deep learning speeds MRI scans. Commun ACM 64(4):12–14
Merhej D, Diab C, Khalil M, Prost R (2011) Embedding prior knowledge within compressed sensing by neural networks. IEEE Trans Neural Netw 22(10):1638–1649
Metzler CA, Maleki A, Baraniuk RG (2015) BM3D-AMP: a new image recovery algorithm based on BM3D denoising. In: 2015 IEEE international conference on image processing (ICIP), pp 3116–3120
Metzler CA, Maleki A, Baraniuk RG (2016) From denoising to compressed sensing. IEEE Trans Inf Theory 62(9):5117–5144
Metzler C, Mousavi A, Baraniuk R (2017) Learned D-AMP: principled neural network based compressive image recovery. In: Advances in neural information processing systems, pp 1772–1783
Monga V, Li Y, Eldar YC (2021) Algorithm unrolling: interpretable, efficient deep learning for signal and image processing. IEEE Signal Process Mag 38(2):18–44
Mousavi A, Baraniuk RG (2017) Learning to invert: signal recovery via deep convolutional networks. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 2272–2276
Mousavi A, Patel AB, Baraniuk RG (2015) A deep learning approach to structured signal recovery. In: 2015 53rd annual Allerton conference on communication, control, and computing (Allerton), pp 1336–1343
Mousavi A, Dasarathy G, Baraniuk RG (2017) Deepcodec: adaptive sensing and recovery via deep convolutional neural networks. In: IEEE 55th annual Allerton conference on communication, control, and computing, p 744
Muckley MJ et al (2021) Results of the 2020 fastmri challenge for machine learning mr image reconstruction. IEEE Trans Med Imaging 40(9):2306–2317
Novikov A, Podoprikhin D, Osokin A, Vetrov DP (2015) Tensorizing neural networks. Adv Neural Inf Process Syst 28:442–450
Oiknine Y, August I, Farber V, Gedalin D, Stern A (2018) Compressive sensing hyperspectral imaging by spectral multiplexing with liquid crystal. J Imaging 5(1):3

Ouchi S, Ito S (2020) Reconstruction of compressed-sensing MR imaging using deep residual learning in the image domain. In: Magnetic resonance in medical sciences mp–2019
Palangi H, Ward R, Deng L (2016a) Distributed compressive sensing: a deep learning approach. IEEE Trans Signal Process 64(17):4504–4518
Palangi H, Ward R, Deng L (2016b) Reconstruction of sparse vectors in compressive sensing with multiple measurement vectors using bidirectional long short-term memory. In: 2016 IEEE global conference on signal and information processing (GlobalSIP), pp 192–196
Pei Y, Liu Y, Ling N (2020) Deep learning for block-level compressive video sensing. In: 2020 IEEE international symposium on circuits and systems (ISCAS), pp 1–5
Polania LF, Barner KE (2017) Exploiting restricted boltzmann machines and deep belief networks in compressed sensing. IEEE Trans Signal Process 65(17):4538–4550
Pramanik PKD et al (2019) Power consumption analysis, measurement, management, and issues: a state-of-the-art review of smartphone battery and energy usage. IEEE Access 7:182113–182172
Qie Y, Hao C, Song P (2020) Wireless transmission method for large data based on hierarchical compressed sensing and sparse decomposition. Sensors 20(24):7146
Qin C et al (2018) Convolutional recurrent neural networks for dynamic MR image reconstruction. IEEE Trans Med Imaging 38(1):280–290
Ramzi Z, Ciuciu P, Starck J-L (2020) Benchmarking mri reconstruction neural networks on large public datasets. Appl Sci 10(5):1816
Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, pp 234–241
Scardapane S, Scarpiniti M, Baccarelli E, Uncini A (2020) Why should we add early exits to neural networks? Cogn Comput 12(5):954–966
Schlemper J, Caballero J, Hajnal JV, Price AN, Rueckert D (2017) A deep cascade of convolutional neural networks for dynamic MR image reconstruction. IEEE Trans Med Imaging 37(2):491–503
Schniter P, Rangan S, Fletcher AK (2016) Vector approximate message passing for the generalized linear model. In: 2016 50th Asilomar conference on signals, systems and computers 1525–1529
Sekine M, Ikada S (2019) LACSLE: lightweight and adaptive compressed sensing based on deep learning for edge devices. In: 2019 IEEE global communications conference (GLOBECOM), pp 1–7
Shawky H et al (2017) Efficient compression and reconstruction of speech signals using compressed sensing. Int J Speech Technol 20(4):851–857
Shen Y et al (2018) CS-CNN: Enabling robust and efficient convolutional neural networks inference for internet-of-things applications. IEEE Access 6:13439–13448
Shi W, Jiang F, Liu S, Zhao D (2020) Image compressed sensing using convolutional neural network. IEEE Trans Image Process 29:375–388
Shrivastwa RR, Pudi V, Chattopadhyay A (2018) An FPGA-based brain computer interfacing using compressive sensing and machine learning. In: 2018 IEEE computer society annual symposium on VLSI (ISVLSI), pp 726–731
Shrivastwa RR et al (2020) A brain-computer interface framework based on compressive sensing and deep learning. IEEE Consum Electron Mag 9(3):90–96
Singhal V, Majumdar A, Ward RK (2017) Semi-supervised deep blind compressed sensing for analysis and reconstruction of biomedical signals from compressive measurements. IEEE Access 6:545–553
Song Y, Cao Z, Wu K, Yan Z, Zhang C (2020) Learning fast approximations of sparse nonlinear regression. arXiv:2010.13490
Sun B, Feng H, Chen K, Zhu X (2016a) A deep learning framework of quantized compressed sensing for wireless neural recording. IEEE Access 4:5169–5178
Sun J, Li H, Xu Z et al (2016b) Deep ADMM-Net for compressive sensing MRI. Adv Neural Inf Process Syst 29:10–18
Sun L, Fan Z, Huang Y, Ding X, Paisley JW (2018) Compressed sensing MRI using a recursive dilated network. In: AAAI, pp 2444–2451
Thapaliya N, Goluguri L, Suthaharan S (2020) Asymptotically stable privacy protection technique for fMRI shared data over distributed computer networks. In: Proceedings of the 11th ACM international conference on bioinformatics, computational biology and health informatics, pp 1–8
Vargas H, Fonseca Y, Arguello H (2018) Object detection on compressive measurements using correlation filters and sparse representation. In: 2018 26th European signal processing conference (EUSIPCO), pp 1960–1964
Wang D, Liu X-W (2022) A gradient-type noise-tolerant finite-time neural network for convex optimization. Neurocomputing

Wang S et al (2016) Accelerating magnetic resonance imaging via deep learning. In: 2016 IEEE 13th international symposium on biomedical imaging (ISBI), pp 514–517
Wang Y et al (2017) Compressive sensing of hyperspectral images via joint tensor tucker decomposition and weighted total variation regularization. IEEE Geosci Remote Sens Lett 14(12):2457–2461
Wang G, Niu M-Y, Fu F-W (2019) Deterministic constructions of compressed sensing matrices based on codes. Cryptogr Commun 11(4):759–775
Wimalajeewa T, Varshney PK (2017) Application of compressive sensing techniques in distributed sensor networks: a survey. arXiv:1709.10401
Wu C-J et al (2019a) Machine learning at facebook: understanding inference at the edge. In: 2019 IEEE international symposium on high performance computer architecture (HPCA), pp 331–344
Wu K, Guo Y, Li Z, Zhang C (2019b) Sparse coding with gated learned ISTA. In: International conference on learning representations
Xiao S, Li T, Yan Y, Zhuang J (2019) Compressed sensing in wireless sensor networks under complex conditions of internet of things. Clust Comput 22(6):14145–14155
Xu Y, Liu W, Kelly KF (2020) Compressed domain image classification using a dynamic-rate neural network. IEEE Access 8:217711–217722
Xu K, Ren F (2016) Csvideonet: a recurrent convolutional neural network for compressive sensing video reconstruction. arXiv:1612.05203
Xuan VN, Loffeld O (2018) A deep learning framework for compressed learning and signal reconstruction. In: 5th international workshop on compressed sensing applied to radar, multimodal sensing, and imaging (CoSeRa), pp 1–5
Yang Y, Sun J, Li H, Xu Z (2017) ADMM-Net: a deep learning approach for compressive sensing MRI. arXiv:1705.06869
Yang G et al (2018) DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. IEEE Trans Med Imaging 37(6):1310–1321
Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: concept and applications. ACM Trans Intell Syst Technol 10(2):1–19
Yang Y, Sun J, Li H, Xu Z (2020) ADMM-CSNet: a deep learning approach for image compressive sensing. IEEE Trans Pattern Anal Mach Intell 42(3):521–538
Yu S et al (2017) Deep de-aliasing for fast compressive sensing MRI. IEEE Trans Med Imaging. arXiv:1705.07137
Yao H et al (2019) DR2-Net: deep residual reconstruction network for image compressive sensing. Neurocomputing 359:483–493
Yao S et al (2020) Deep compressive offloading: speeding up neural network inference by trading edge computation for network latency. In: Proceedings of the 18th conference on embedded networked sensor systems, pp 476–488
Yu J, Yang L, Xu N, Yang J, Huang T (2018) Slimmable neural networks. In: International conference on learning representations
Zhang J, Ghanem B (2018) ISTA-Net: interpretable optimization-inspired deep network for image compressive sensing. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1828–1837
Zhang Z, Wu Y, Gan C, Zhu Q (2019) The optimally designed autoencoder network for compressed sensing. EURASIP J Image Video Process 2019(1):1–12
Zhang Y, Li X, Zhao G, Lu B, Cavalcante CC (2020a) Signal reconstruction of compressed sensing based on alternating direction method of multipliers. Circuits Syst Signal Process 39(1):307–323
Zhang X, Lian Q, Yang Y, Su Y (2020b) A deep unrolling network inspired by total variation for compressed sensing MRI. Digit Signal Process 107:102856
Zhang Z, Liu Y, Liu J, Wen F, Zhu C (2020c) AMP-Net: denoising based deep unfolding for compressive image sensing. In: IEEE transactions on image processing: a publication of the IEEE signal processing society
Zhang H, Dong Z, Wang Z, Guo L, Wang Z (2021) Csnet: a deep learning approach for ECG compressed sensing. Biomed Signal Process Control 70:103065
Zhao X, Li W, Zhang M, Tao R, Ma P (2020a) Adaptive iterated shrinkage thresholding-based Lp-norm sparse representation for hyperspectral imagery target detection. Remote Sens 12(23):3991
Zhao Z, Xie X, Liu W, Pan Q (2020b) A hybrid-3D convolutional network for video compressive sensing. IEEE Access 8:20503–20513
Zhou Z, Yu J (2019) A new nonconvex sparse recovery method for compressive sensing. Front Appl Math Stat 5:14
Zur Y, Adler A (2019) Deep learning of compressed sensing operators with structural similarity loss. arXiv:1906.10411