applied sciences

MDPI

Review

An Overview of Open Source Deep Learning-Based Libraries for Neuroscience

Louis Fabrice Tshimanga [1,2,*], Federico Del Pup [1,2,3], Maurizio Corbetta [1,2,4] and Manfredo Atzori [1,2,5,*]

1 Department of Neuroscience, University of Padova, Via Belzoni 160, 35121 Padova, Italy
2 Padova Neuroscience Center, University of Padova, Via Orus 2/B, 35129 Padova, Italy
3 Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy
4 Venetian Institute of Molecular Medicine (VIMM), 35129 Padova, Italy
5 Information Systems Institute, University of Applied Sciences Western Switzerland (HES-SO Valais), 2800 Sierre, Switzerland
* Correspondence: louisfabrice.tshimanga@unipd.it (L.F.T.); manfredo.atzori@unipd.it (M.A.)

Abstract: In recent years, deep learning has revolutionized machine learning and its applications, producing results comparable to human experts in several domains, including neuroscience. Each year, hundreds of scientific publications present applications of deep neural networks for biomedical data analysis. Due to the fast growth of the domain, it could be a complicated and extremely time-consuming task for worldwide researchers to have a clear perspective of the most recent and advanced software libraries. This work contributes to clarifying the current situation in the domain, outlining the most useful libraries that implement and facilitate deep learning applications for neuroscience, allowing scientists to identify the most suitable options for their research or clinical projects. This paper summarizes the main developments in deep learning and their relevance to neuroscience; it then reviews neuroinformatic toolboxes and libraries collected from the literature and from specific hubs of software projects oriented to neuroscience research. The selected tools are presented in tables detailing key features grouped by the domain of application (e.g., data type, neuroscience area, task), model engineering (e.g., programming language, model customization), and technological aspect (e.g., interface, code source). The results show that, among a high number of available software tools, several libraries stand out in terms of functionalities for neuroscience applications. The aggregation and discussion of this information can help the neuroscience community to develop their research projects more efficiently and quickly, both by means of readily available tools and by knowing which modules may be improved, connected, or added.

Keywords: deep learning; machine learning; neuroscience; neuroinformatics; open source

check for updates

Citation: Tshimanga, L.F.; Del Pup, F.; Corbetta, M.; Atzori, M. An Overview of Open Source Deep Learning-Based Libraries for Neuroscience. *Appl. Sci.* **2023**, *13*, 5472. https://doi.org/10.3390/app13095472

Academic Editor: Alexander E. Hramov, Alexander N. Pisarchik, Victor B. Kazantsev

Received: 22 March 2023
Revised: 19 April 2023
Accepted: 22 April 2023
Published: 27 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the last decade, deep learning has taken over most classic approaches in machine learning, computer vision, and Natural Language Processing (NLP) research, showing unprecedented versatility and matching or surpassing the performances of human experts in narrow tasks. The recent growth of deep learning applications in several domains, including neuroscience, consequently offers numerous open-source software opportunities for researchers. Mapping available resources can allow for faster and more precise exploitation. Neuroscience is a diversified field on its own, as much for the objects and scales it focuses on as for the types of data it relies on. The discipline is also historically tied to developments in electrical, electronic, and information technology. Modern neuroscience relies on computerization in many aspects of data generation, acquisition, and analysis. Statistical and machine learning techniques already empower many software packages that have become de facto standards in several subfields of neuroscience, such as Principal and Independent Component Analysis (PCA, ICA) in electroencephalography and neuroimaging, to name a few. Concurrently, the rich and rapidly evolving taxonomy of Deep Neural

Networks (DNNs) is becoming both an opportunity and a hindrance. On the one hand, currently, open-source deep learning libraries allow an increasing number of applications and studies in neuroscience. On the other hand, the adoption of available methods is slowed down by a lack of standards, reference frameworks, and established workflows. Scientific communities whose primary focus or background is not in machine learning engineering may be left partially aside from the ongoing Artificial Intelligence (AI) gold rush. For such reasons, it is fundamental to provide an overview of open-source libraries and toolkits. Framing a panorama could help researchers in selecting ready-made tools and solutions when convenient and aid them in pointing out problems and filling in the blanks with new applications. This work would contribute to advancing the community's possibilities, reducing the workload for researchers to exploit deep learning, and allowing neuroscience to benefit from its most recent advancements.

The rest of the paper is organized as follows: first in the Section 2, a historical perspective of the rise of deep learning in the last decade, then, a general presentation of the vast field of neuroscience followed by a definition of neuroinformatics and the role open source culture and deep learning would serve in it; subsequently, in the Section 3, the methodology to collect and present the libraries collection, and the most prominent features are discussed; the Section 4 shows the tables with libraries information; lastly, discussions and final remarks are offered to the readers (Sections 5 and 6).

2. Background

2.1. Deep Learning

Deep learning has contributed many of the best solutions to problems in its parent field, machine learning, thanks to its theoretical and technological achievements that unlocked its intrinsic versatility. Machine learning is the study of computer algorithms that tackle problems without complete access to predefined rules or analytical, closed-form solutions. The algorithms often require a training phase to adjust parameters and satisfy internal or external constraints (e.g., of exactness, approximation, or generality) on dedicated data for which solutions might be already known. Machine learning comprises a wide array of statistical and mathematical methods, including Artificial Neural Networks (ANNs), biologically inspired systems that connect inputs and outputs through simple computing units (neurons), which act as function approximators. Each unit implements a nonlinear function of the weighted sum of its inputs; thus, the output of the whole ANN is a composite function, as formally intended in mathematics. The networks of neurons are most often layered and "feed-forward", meaning that units from any layer only output results to units in subsequent layers. The width of a layer refers to its neuron count, while the depth of a network refers to its layer count. The typical architecture instantiating the above characteristics is the MultiLayer Perceptron [1] (MLP). Universal approximation theorems [2,3] ensure that whenever a nonlinear network, such as the MLP, is either bound in width and unbound in depth or vice versa, its weights can then be set to represent virtually any function (i.e., a wide variety of function families). The training problem thus consists of building networks with sets of weights so to instantiate or approximate the function that would solve the assigned task or that represents the input-output relation. This search is not trivial: it can be framed as the optimization problem for a function over the ANN weights. Such functions, typically called "loss function", associates the "errors" made on the training data to the neural net parameters (its weights), acting as a total performance score. Approaching local minima of the loss function and improving the network performance on the training data is the prerequisite to generalizing on real-world and unseen data. Deep learning is concerned with the use of deep ANNs, namely characterized by depth, stacking several intermediate (hidden) layers between input and output units. As mentioned above, with other dimensions being equal, the depth increases the representational power of ANNs and, more specifically, aims at modeling complicated functions as meaningful compositions of simpler ones. As with their biological counterparts [4], depth is supposed to manage hierarchies of features from larger input portions, capturing characteristics often inherent to real-world objects and effective in modeling actual data. Overall, depth is one of the key features that allowed us to overcome historical limits [5] of simpler ANNs such as the Perceptron. At the same time, depth comes with numerical and methodological hardships in model training. Part of the difficulties arises as the search space for the optimal set of parameters grows considerably with the number of layers (and their width as well). Other issues are strictly numerical since the training algorithms include long computation chains that may affect the stability of training and learning. Hence, new or rediscovered ideas in training protocols and mathematical optimization (e.g., applying the "backpropagation of errors" algorithm to neural nets [6]) played an important role through times when the scientific interest and hopes in ANNs faded (so-called "AI winters"), paving the way for later advancement. The main drivers for the latest success of deep neural networks are of varied nature and can be schematized as technical and human-related factors. On a technical side, deep learning has profited from [7]:

- The datafication of the world, i.e., the growing availability of (Big) data
- The diffusion of Graphical Processing Units (GPUs) as hardware tools.

To outperform classic machine learning models, deep neural networks often require larger quantities of data samples. Such data hunger and high parameter count contribute to the high requirements of deep models in terms of memory, number of operations, and computation time. Training models with highly parallelized and smartly scheduled computations gained momentum thanks to GPUs. In 2012 a milestone exemplified both the above technical aspects when AlexNet [8], a deep Convolutional Neural Network (CNN) based on ideas from Fukushima [4] and LeCun [9,10], won the ImageNet Large Scale Visual Recognition Challenge after being trained using two GPUs [11]. Since then, deep learning has brought new outstanding results in various tasks and domains, processing different data types. Nowadays, deep networks can work on images, video, audio, text, and speech data, time series and sequences, graphs, and more; the main tasks consist of classification, prediction, or estimating the probability density of data distributions, with the possibility of modifying, completing the input, or even generating new instances. On a more sociological side, the drivers of deep learning success can be related to the synergy of big tech companies, advanced research centers, and developer communities [12]. Investments of economic and scientific resources in relatively independent, collective projects, such as open-source libraries, frameworks, and APIs (Application Programming Interfaces), have offered varied tools adapted to multiple specific situations and objectives, exploiting horizontal organization [13] and mixing top-down and bottom-up approaches. It is difficult to imagine a rapid rise of successful endeavors without both active communities and the technical means to incorporate and manage lower-level aspects. In fact, applying deep learning to a relevant problem in any research field requires, in addition to specific domain knowledge, a vast background of statistical, mathematical, and programming notions and skills. The tools that support scientists and engineers in focusing on their main tasks encompass the languages to express numerical operations on GPUs, such as CUDA [14] and cuDNN [15] by NVIDIA, as well as the frameworks to design models, like TensorFlow [16] and Keras [17] by Google, and PyTorch by Meta [18], or the supporting strategies to build data pipelines. In particular, PyTorch, TensorFlow, and Keras offer the building blocks for model design. These frameworks comprise the mathematical operations and functions that deep learning models perform during training and at test time. The functions can be treated as modular objects, stacked one upon another, or connected in more complex ways. The data are input and processed through these objects-functions chains to return corresponding outputs. On a higher level, one can ignore computational and mathematical details as long as the role and effect of such components are understood. On a lower level, these frameworks allow the experts in the community to introduce and share novel custom objects and operations that push forward deep learning research. Data loading and preprocessing modules are included, as well as many pre-trained deep learning models, enhancing the framework's adaptability and usability. Many deep learning achievements are relevant to biomedical and clinical research, and the above-presented tools have enabled explorations of the capabilities of deep neural networks with neuroscience and biomedical data. Fuller exploitation and routine employment of modern algorithms are yet to come, both in research and clinical practice. This process would accelerate by popularizing, democratizing, and jointly developing models, improving their usability, and expanding their environments, i.e., by wrapping solutions into libraries and shared frameworks.

2.2. Neuroscience

As per the journal *Nature*, «neuroscience is a multidisciplinary science that is concerned with the study of the structure and function of the nervous system. It encompasses the evolution, development, cellular and molecular biology, physiology, anatomy, and pharmacology of the nervous system, as well as computational, behavioral, and cognitive neuroscience» [19]. In summary, neuroscience investigates:

- The evolutionary and individual development of the nervous system;
- The cellular and molecular biology that characterizes neurons and glial cells;
- The physiology of living organisms and the role of the nervous system in the homeostatic function;
- The anatomy, i.e., the identification and description of the system's structures;
- Pharmacology, i.e., the effect of chemicals of external origin on the nervous system, their interactions with endogenous molecules;
- The computational features of the brain and nerves, how information is processed, which mathematical and physical models best predict and approximate the behavior of neurons;
- Cognition, the mental processes at the intersection of psychology and computational neuroscience;
- Behavior as a phenomenon rooted in genetics, development, mental states, and so forth.

Overall, given the wide range of phenomena and the apparatus it investigates, neuroscience research is profoundly multi-modal. Data range from sequences or signals (e.g., electromyography (EMG), electroencephalography (EEG), eye-tracking, genetic sequencing), to 2D/3D images (e.g., Magnetic Resonance Imaging (MRI), X-rays, tomography, histopathology microscopy, eye fundus photography) or videos. Tabular data and text data are also common in this field, from clinical reports and anamneses to surveys, test scores, and inspections of cognitive and sensorimotor functions (e.g., the National Institute of Health (NIH) Stroke Scale test scores [20]), and more.

2.3. Neuroinformatics

Neuroscience is evolving into a data-centric discipline. Modern research heavily depends on human researchers as well as machine agents to store, manage, and process computerized data from the experimental apparatus to the end stage. Before delving into the specifics of artificial neural networks applied to the study of biological neural systems, it is useful to outline the broader concepts of neuroinformatics, regarding data and coding, especially in the light of open culture. According to the International Neuroinformatics Coordinating Facility (INCF), «neuroinformatics is a research field devoted to the development of neuroscience data and knowledge bases together with computational models and analytical tools for sharing, integration, and analysis of experimental data and advancement of theories about the nervous system function.» [21]. Given the relevance of neuroinformatics to neuroscience, supporting open and reproducible science implies and requires attention to standards and best practices regarding open data and code. The INCF itself is an independent organization devoted to validating and promoting such standards and practices, interacting with the research communities [22] and aiming at the "FAIR principles for scientific data management and stewardship" [23]. FAIR principles consist in:

- Being Findable, registered and indexed, searchable, richly described in metadata;
- Being Accessible, through open, free, universally implementable protocols;

- Being Interoperable, with appropriate standards for metadata in the context of knowledge representation;
- Being Reusable, clearly licensed, well described, relevant to a domain, and meeting community standards.

Among free and open resources, several software and organized packages integrating pre-processing and data analysis workflows for neuroimaging and signal processing became the reference for worldwide researchers in neuroscience.
Such tools allow us to perform scientific research in neuroscience easily in solid and repeatable ways. It can be useful to mention, for neuroimaging, Freesurfer (https://surfer.nmr.mgh.harvard.edu/) [24] and FSL (https://fsl.fmrib.ox.ac.uk/fsl/fslwiki) [25] that are standalone softwares, and the MATLAB-connected SPM (https://www.fil.ion.ucl.ac.uk/spm/) [26]. In the domain of signal processing, examples are EEGLAB (https://sccn.ucsd.edu/eeglab/index.php) [27], Brainstorm (https://neuroimage.usc.edu/brainstorm/Introduction) [28], PaWFE (http://ninapro.hevs.ch/node/229) [29], all MATLAB related yet free and open, and MNE (https://mne.tools/stable/index.html) [30], that runs on Python. Regarding applications for neurorobotics and Brain Computer Interfaces (BCIs), a recent open source platform can be found in ROS-neuro (https://github.com/rosneuro) [31]. All URLs accessed at date 22nd of November 2022. Interested readers can find lists of open resources for computational neuroscience (including code, data, models, repositories, textbooks, analysis, simulation, and management software) at Open Computational Neuroscience Resource (https://github.com/asoplata/open-computational-neuroscience-resources) (by Austin Soplata), and at Open Neuroscience (https://open-neuroscience.com/). Additional software resources oriented to neuroinformatics in general, but not necessarily open, can also be found as indexed at "COMPUTATIONAL NEUROSCIENCE on the Web" (https://compneuroweb.com/sftwr.html) (by Jim Perlewitz).

2.4. Bringing Deep Learning to the Neurosciences

The deep learning community is accustomed to open science, as many datasets, models, programming frameworks, and scientific outcomes are publicly released by both academia and companies continuously. However, while deep learning can openly provide state-of-the-art models for old and new problems in neuroscience, theoretical understanding, formalization, and standardization are often yet to be achieved, which may prevent adoption in other research endeavors. From a technical standpoint, deep networks are a viable tool for many tasks involving data from the brain sciences. Image classification has arguably been the task in which deep neural networks have had the highest momentum in terms of pushing the state of the art forward. This translates now into a rich taxonomy of architectures and pre-trained models that consistently maintain interesting performances in pattern recognition across a number of image domains. Pattern recognition is indeed central for diagnostic purposes, in the form of classification of images with pathological features (e.g., types of brain tumors or meningiomas), segmentation of structures (such as the brain, brain tumors, or stroke lesions), classification of signals (e.g., classification of electromyography or electro encephalography data), as well as for action recognition in Human-Computer Interfaces (HCIs) and Brain-Computer Interfaces (BCIs), where the complex systems underlying human behavior and mind must be interpreted, processed and used by artificial systems (see [32] for a larger review of BCIs). The initiatives BRain Tumor Segmentation (BRATS) Challenge (https://www.med.upenn.edu/cbica/brats/) [33], Ischemic Stroke LEsion Segmentation (ISLES) Challenge (https://www.isles-challenge.org/) [34,35], and Ninapro (http://ninaweb.hevs.ch/node/7) [36] are examples of data releases for which above-mentioned tools proved effective. There are models learning image-to-image functions capable of enhancing data, preprocessing it, correcting artifacts and aberrations, allowing smart compression as well as super-resolution, and even expressing cross-modal transformations between different acquisition apparatuses. In the related tasks of object tracking, action recognition, and pose estimation, research results from the automotive sector or crowd analysis have inspired solutions for behavioral neuroscience, especially in animal behavioral studies. When dealing with sequences, deep networks success in computer vision has inspired CNN-based approaches to EEG and EMG studies [37,38], either with or without relying on 2D data, given that mathematical convolution has a 1D version, and 1D signals have 2D spectra. Other architectures more directly instantiate temporal and sequential aspects, e.g., Recurrent Neural Networks (RNNs) such as the Long Short Term Memory (LSTM) [39] and Gated Recurrent Units (GRUs) [40], and they too can be applied to sequence problems and sub-tasks in neuroscience, such as decoding time-dependent brain signals. Although deep neural networks do not explicitly model the nervous system, they are inspired by biological knowledge and mimic some aspects of biological computation and dynamical systems. This has inspired new comparative studies and analogous approaches to learning and perception in a unique way among machine learning algorithms [41]. Many neuroinformatic studies demonstrate how novel deep learning concepts and methods apply to neurological data [12]. However, they often showcase new advanced achievements in performance metrics that do not translate directly to new accepted neuroscience discoveries or clinical best practices.

Such results are very often published together with open code repositories, allowing for reproducibility, yet they may not be explicitly organized for widespread routine adoption in domains different from machine learning. Algorithms are usually written in open programming languages like Python [42], R [43], Julia [44], and deep learning design frameworks such as TensorFlow, PyTorch or Flux [45]. Still, they are more inspiring to the experienced machine learning researcher rather than practically helpful to end users such as neuroscientists. In fact, to successfully build a deep learning application from scratch, vast knowledge is needed in the data science aspect of the task and in coding, as much as in the theoretical and experimental foundations and frontiers of the application domain, here being neuroscience. For the above reasons, the open source and open science domains are promising frames for common development and testing of relevant solutions for neuroscience, as they provide an active flow of ideas and robust diversification, avoiding "reinvention of the wheel", harmful redundancies, or starting from completely blank states. As a contribution to clarifying the current situation and reducing the workload for researchers, this work collects and analyzes several open libraries that implement and facilitate deep learning applications in neuroscience, with the aim of allowing scientists worldwide to identify the most suitable options for their inquiries and clinical tasks.

3. Materials and Methods

The large corpus of available open code makes it useful to specify what qualifies as a coding library or a framework rather than as a model accompanied by utilities for the present scope. In programming, a library is a collection of pre-coded functions and object definitions, often relying on one another and written to optimize programming for custom tasks. The functions are considered useful and unmodified across multiple unrelated programs and tasks. The main program at hand calls the library in the control flow specified by the end users. A framework is a higher level concept, akin to the library, but typically with pre-designed control flows in which custom code from the end users is inserted.

In this review, a repository that simply collects a set of functions that defines and instantiates a deep learning model is not considered a library. On the contrary, a collection of notebooks that allows us to train, retrain, and test models with different architectures, also taking care of data preprocessing and preparation, fully meets the present scopes. The explicit definition given by the authors, their aims, and their level of maintenance were relevant in determining if a repository would be considered a library (or toolkit/toolbox, etc.). Open code for this review comprises code for proprietary languages such as MATLAB, the reasons being the compatibility with free languages such as GNU Octave (https://octave.org/) (where noted), and the general value of open accessing algorithms. For the sake of the review, several resources were queried or scanned. Google Scholar was queried with:

- Allintitle: "deep learning library";
- Allintitle: "deep learning toolbox";
- Allintitle: "deep learning package";
- "deep learning library | toolbox | package" AND "neuroscience | neuroimaging";
- "deep learning library | toolbox | package" AND "EEG | EMG";
- "deep learning library" OR "deep learning toolbox" OR "deep learning package" -"MATLAB deep learning toolbox"

preserving the top 100 search results, ordered for relevance by the engine algorithm. On PubMed the queries were:

- opensource (deep learning) AND (toolbox OR toolkit OR library);
- (EEG OR EMG OR MRI OR (brain (X-ray OR CT OR PT))) (deep learning) AND (toolbox OR toolkit OR library).

Moreover, the site https://open-neuroscience.com/ was scanned specifically for "deep learning" mentions. Stemming citations and automatic recommendations from the engines of the hosting and publishing platforms were also analyzed. The time window was unrestricted in the past, given the recent development of the field, and the search was finished by 22 November 2022 for the selection of library entries. Data regarding library use were updated to 17 April 2023.

The collected libraries were organized according to the principal aim, in the form of data type processed or the supporting function in the workflow, thus dividing:

1. Libraries for sequence data (e.g., EMG, EEG)
2. Libraries for image data (including scalar volumes, 4-dimensional data as in fMRI, video)
3. Libraries and frameworks for further data types and abstractions (including data handling, evaluation, and cloud platforms)

In each category, a set of three tables present separately the results related to the following library characteristics:

1. Domain of application
2. Model engineering
3. Technology and sources

The domain of application comprises the *Neuroscience area* , the *Data types* handled, the provision of *Datasets*, and the machine learning *Task* to which the library is dedicated. When available (73 entries out of 74), a publication is referenced for the library entry in the domain table. Together with the repositories, referenced publications contain valuable information every potential user should check before experimenting, such as the data sets leveraged and use cases intended by the original authors. The model engineering tables include information on the architecture of *Models* manageable in the library, the *DL (Deep Learning) framework* and *Programming language* main dependencies, and the possibility of *Customization* for the model structure or training parameters. Technology and sources refer to the type of *Interface* available for a library, whether it works *Online/Offline*, specifically with real-time data or logged data. *Maintenance* refers to the ongoing activity of releasing features, solving issues and bugs, or offering support through channels (considered active with commits or releases in 2022), *Source* specifies where code files and instructions are made available. *Stars (Forks)* refers to the counts of "stars" and "forks" of the repositories, by which the number of users and possible new developers could be approximately estimated.*Contributors* is the number of people adding code and features to the repository, as declared on site (effective contributors might be differently acknowledged), which is useful to estimate the amount of teamwork, developer support, as well as the inclination to customization that could be expected for the given library. The entry "(*)" signals missing data.

4. Results

The analysis of the literature allowed us to select a total of 74 entries for the tables, with publications that describe libraries implementing or empowering deep learning applications for neuroscience. Despite open source and effectiveness, several publications did not provide an ecosystem of reusable functions. Proofs of concept and single-shot experiments were discarded. Please, refer to the Abbreviations section for acronyms from the entire paper and specifically the following tables.

4.1. Libraries for Sequence Data

Libraries and frameworks for sequence data are shown in Table 1 (domains of application), Table 2 (models characteristics), Table 3 (technologies and sources). The majority of models process EEG signals, which are among the most common types of sequential data in neuroscience research. A common objective is deducing the activity or state of the subject based on temporal or spectral (2D) patterns. Deep Learning is capable of bypassing some of the preprocessing steps often required by other common statistical and engineering techniques, and it comprises both 1D and 2D approaches through MLPs, CNNs, or RNNs architectures. An example of a sequence-oriented library is gumpy, whose intended area of application is that of BCIs, where decoding a signal is the first step towards communication and interaction with a computer or robotic system. Given the setting, gumpy allows working with EEG or EMG data and suits them with specific defaults, e.g., 1-D CNNs, or LSTMs. Similarly to ExBrainable, it was validated on data from the BCI Competition IV (https://www.bbci.de/competition/iv/).

Notable mentions in the sequence category are the library Traja and the VARDNN toolbox, as they depart from the common scenarios of previous examples. Traja stands out as an example of less usual sequential data, namely trajectory data (sequences of coordinates in 2 or 3 dimensions, through time). Moreover, in Traja sequences are modeled and analyzed employing the advanced architectures of Variational AutoEncoders (VAEs) and Generative Adversarial Networks (GANs), usually encountered in image tasks. With different theoretical backgrounds, both architectures allow simulation and characterization of data through their statistical properties. The VARDNN toolbox enables analyses on blood-oxygen-level-dependent (BOLD) signals in the established domain of functional Magnetic Resonance Imaging (fMRI) but uses a unique approach to autoregressive processes mixed with deep neural networks, allowing to perform causal analysis and to study functional connections between brain regions through their patterns of activity in time. It was developed from the data set ADNI-2 (73 subjects) from the Alzheimer's Disease Neuroimaging Initiative (https://adni.loni.usc.edu/).

Overall, the libraries oriented to sequence data analysis are mainly directed at the classification of EEG signals, whose variety of acquisition settings and downstream applications could be largely approached with the aid of deep models as a part of the pipeline. Other types of sequence data in neuroscience could be processed by newer or harder-to-retrieve libraries. Despite the fact that preprocessing and domain-specific features require special care, sequence model can still be applied in principle to these data to perform several machine learning tasks on magnetoencephalography (MEG), electrocorticography (ECoG), spike train data, and more. The expert end user may apply or adapt the libraries mentioned above to new domains, or develop new applications, possibly leveraging the open source of available code.

Table 1. Domains of applications for the libraries and frameworks processing sequence data.

Name	Neuroscience Area	Data Type	Datasets	Task
braindecode [46]	General	EEG, MEG	External	Classification
DeepEEG	Electrophysiology	EEG	No	Classification
DeLINEATE [47]	General	Images, sequences	External	Classification
DN3 [48]	BCI	EEG	No	Classification
EEG-DL [49]	BCI	EEG	No	Classification
ExBrainable [50]	Electrophysiology	EEG	External	Classification, XAI
gumpy [51]	BCI	EEG, EMG	No	Classification
SANTIA [52]	Electrophysiology	Local Field Potentials	No	Processing
Traja [53]	Behavioural neuroscience	Trajectories	No	Prediction, Classification, Synthesis
VAME [54]	Behavioral neuroscience	Trajectories	No	Embedding, Clustering
VARDNN toolbox [55]	Connectomics (Functional Connectivity)	Sequences (BOLD signal)	No	Time series causal analysis

Table 2. Model engineering specifications for the libraries and frameworks processing sequence data.

Name	Models	DL Framework	Customization	Programming Language
braindecode	1-D CNN	PyTorch	Yes (weights, model)	Python
DeepEEG	MLP, 1,2,3-D CNN, LSTM	Keras, TensorFlow	Yes (weights)	Python
DeLINEATE	CNN	Keras, TensorFlow	Yes (weights, model)	Python
DN3	MLP	PyTorch	Yes	Python
ExBrainable	CNN	PyTorch	Yes (weights)	Python
EEG-DL	Miscellaneous	TensorFlow	Yes (weights, model)	Python, MATLAB
gumpy	CNN, LSTM	Keras, Theano	Yes (weights, model)	Python
SANTIA	MLP, LSTM, 1-D CNN	Deep Learning Toolbox	Yes (weights, model)	MATLAB
Traja	LSTM, VAE, GAN	PyTorch	Yes (weights, model)	Python
VAME	VAE	PyTorch	Yes (weights, size)	Python
VARDNN toolbox	Vector Auto-Regressive DNN	TensorFlow	Yes (weights)	Python

Table 3. Technological aspects and code sources for the libraries and frameworks processing sequence data. The entry "(*)" signals missing data.

Name	Interface	Online/Offline	Maintenance	Source	Stars (Forks)	Contributors
braindecode	CLI	Offline	Active	https://github.com/braindecode/braindecode	463 (123)	26
DeepEEG	Colab Notebooks	Offline	Inactive	https://github.com/kylemath/DeepEEG	213 (55)	2
DeLINEATE	GUI, Colab Notebooks	Offline	Active	https://bitbucket.org/delineate/delineate	(*)	3
DN3	CLI	Offline	Inactive	https://github.com/SPOClab-ca/dn3	50 (16)	4
EEG-DL	CLI	Offline	Active	https://github.com/SuperBruceJia/EEG-DL	640 (179)	1
ExBrainable	GUI	Offline	Active	https://github.com/CECNL/ExBrainable	1 (2)	4
gumpy	CLI	Online, Offline	Inactive	https://github.com/gumpy-bci	61 (22)	4
SANTIA	GUI	Offline	Inactive	https://github.com/IgnacioFabietti/SANTIAtoolbox	4 (2)	1
Traja	CLI	Offline	Active	https://github.com/traja-team/traja	71 (23)	10
VAME	CLI	Offline	Active	https://github.com/LINCellularNeuroscience/VAME	130 (43)	5
VARDNN toolbox	CLI	Offline	Active	https://github.com/takuto-okuno-riken/vardnnpy	2 (0)	2

4.2. Libraries for Image Data

Libraries and frameworks for image data are shown in Table 4 (domains of application), Table 5 (models characteristics), Table 6 (technologies and sources). Computer vision and 2D image processing are arguably the fields in which deep learning has achieved the most impressive and state-of-art defining results, often inspiring and translating breakthroughs in other domains. Classification and segmentation (i.e., the separation of parts of the image based on their classes) are the most common tasks addressed by image processing libraries. Magnetic resonance is the primary source of data; however, various deep learning libraries are built for microscopic and eye-tracking data as well. Most of the libraries collected in our analysis took advantage of classical CNN architectures for classification, Convolutional AutoEncoders (CAEs) for segmentation, and GANs for synthesis. It is common to employ transfer learning to lessen the computational and memory burden during the training phase and take advantage of pre-trained models. Transfer learning consists of initializing models with parameters learned on usually larger data sets, possibly from different domains and tasks, with varying amounts of further training in the target domain. The best such examples are pose-estimation libraries extending the DeepLabCut system, arguably the most relevant project on the topic. DeepLabCut is an interactive framework for labeling, training, testing, and refining models that originally exploits the weights learned from ResNets (or newer architectures) on the ImageNet data. The results match human annotation using quite a few training samples, holding for many (human and non-human) animals and settings. Validated data sets comprise TRI-MOUSE (161 data points), Parenting Mouse (542), MARMOSET (7600), FISH (100), and HORSE (8114). The documentation, demonstrative notebooks, and tools offered by the Mathis Lab allow different levels of understanding and customization of the process with high levels of robustness. Among the considered libraries, two set apart from the majority given the type of tasks they perform: GaNDLF addresses eXplainable AI (XAI), i.e., Artificial Intelligence whose decisions and outputs can be understood by humans through more transparent mental models; ANTsX performs both the co-registration step and super-resolution as a quality enhancing step for neuroimages, with the former being usually performed by traditional algorithms. GaNDLF sets its goal as the provision of deep learning resources in different layers of abstraction, allowing medical researchers with virtually no ML knowledge to perform robust experiments with models trained on carefully split data, with augmentations and preprocessing, under standardized protocols that can easily integrate interpretability tools such as Grad-CAM [56] and attention maps, which highlight the parts of an image according to how they influenced a model outcome. It was validated on 7 data sets comprising from 371 up to 180,000 images of different systems ranging from brain MRI to dental X-ray and eye fundus. The ANTsX ecosystem is of similar wide scope and is intended to build workflows on quantitative biology and medical imaging data, both in Python and R languages. Packages from the same ecosystem perform registration of brain structures (by classical methods) as well as brain extraction by deep networks, aggregating structural MRI data sets for over 1200 subjects.

Table 4. Domains of applications for the libraries and frameworks processing image data.

Name	Neuroscience Area	Data Type	Datasets	Task
Allen Cell Structure Segmenter [57]	Microbiology, Histology	3D-fluorescence microscopy	No	Segmentation
ALMA [58]	Behavioral neuroscience	Video	External	Pose estimation, Classification
ANTsX [59] (ANTsPyNet, ANTsRNet)	Neuroimaging	MRI	No	Classification, Segmentation, Registration, Superresolution
ATLASS [60]	Medical Imaging	Images	No	Annotation, Classification
AxonDeepSeg [61]	Microbiology, Histology	SEM, TEM	External	Segmentation
BART [62]	Medical Imaging	MRI	No	Reconstruction
Brainstorm [63]	Medical Imaging	MRI	No	Synthesis, Augmentation
CASCADE [64]	Electrophys.	2-photon calcium video, sequences	Yes	Event detection
CDeep3M2 [65]	Microbiology, Histology	Microscopy	Yes	Segmentation
CERR [66]	Oncology, Radiomics	Images	No	Segmentation, Outcome prediction
ClinicaDL [67]	Neuroimaging	MRI, PET	External	Classification, Segmentation
DANNCE [68]	Behavioral neuroscience	Video	Yes	Pose estimation
DeepBehavior [69]	Behavioral neuroscience	Video	Yes	Pose estimation
DeepBhvTracking [70]	Behavioral neuroscience	Video	No	Pose estimation
DeepCINAC [71]	Electrophys.	2-photon calcium video	No	Classification
DeepInfer [72]	Medical Imaging	Images (3D)	No	Classification, Segmentation
DeepLabCut [73]	Behavioral neuroscience	Video	No	Pose estimation
DeepLabStream [74]	Behavioral neuroscience	Video	No	Pose estimation
DeepNeuro [75]	Neuroimaging	Images (fMRI, miscellaneous)	No	Classification, Segmentation, Synthesis
DeepNeuron [76]	Morphology	Images (2D, 3D)	No	Classification, Segmentation
DeepPoseKit [77]	Behavioral neuroscience	Video	No	Pose estimation
DeLINEATE [47]	Medical Imaging	Images, sequences	External	Classification
DeepVOG [78]	Oculography	Images, Video	Demo	Segmentation
DLTK [79]	Medical Imaging	Images	No	Classification, Segmentation
DNNBrain [80]	Brain mapping	Images	No	Classification
FastSurfer [81]	Neuroimaging	MRI	No	Segmentation
fetal-code [82]	Neuroimaging	rs-fMRI	External	Segmentation
GaNDLF [83]	Medical Imaging	Images (2D, 3D)	External	Segmentation, Regression, XAI
hipotalamus_seg [84]	Neuroimaging	MRI	No	Segmentation
ivadomed [85]	Neuroimaging	Images (2D, 3D)	No	Classification, Segmentation
LEAP [86], SLEAP [87]	Behavioral neuroscience	Video	No	Pose estimation
MARS, BENTO [88]	Behavioral neuroscience	Video	Yes	Pose estimation, Classification, Action recognition, Tag
MesoNet [89]	Neuroimaging	Images (fluoresc. microscopy)	External	Segmentation, Registration

Table 4. *Cont.*

Name	Neuroscience Area	Data Type	Datasets	Task
MEYE [90]	Oculography	Images, Video	Yes	Segmentation
MIScnn [91]	Medical Imaging	Images (2D, 3D)	No	Segmentation
Neurite, Neuron [92]	Neuroimaging	Images	No	Segmentation
NiftyNet [93]	Medical Imaging	MRI, CT	No	Classification, Segmentation, Synthesis
NiftyTorch [94]	Neuroimaging	Images (2D, 3D)	No	Classification, Segmentation, Synthesis
nnU-Net [95]	Medical Imaging	Images (2D, 3D)	No	Segmentation
PyTC [96]	Connectomics	Images (2D, 3D)	No	Segmentation
ScLimbic [97]	Neuroimaging	MRI	External	Segmentation
SimBA [98]	Behavioral neuroscience	Video	No	Pose estimation
SynthStrip [99]	Neuroimaging	Images (3D)	No	Segmentation, Extraction
VesicleSeg [100]	Microbiology, Histology	EM	No	Segmentation
Visual Fields Analysis [101]	Eye tracking, Behavioral neuroscience	Video	No	Pose estimation, Classification
Volume Segmantics [102]	Neuroimaging	Images (3D)	No	Segmentation
VoxelMorph [103], HyperMorph [104]	Neuroimaging	MRI	No	Registration

Table 5. Model engineering specifications for the libraries and frameworks processing image data.

Name	Models	DL Framework	Customization	Programming Language
Allen Cell Structure Segmenter	CAE	PyTorch	No	Python
ALMA	CNN	Unspecified	No	Python
ANTsX (ANTsPyNet, ANTsRNet)	CNN, CAE, GAN	Keras, TensorFlow	Yes	Python, R, C++
ATLASS	CNN	FastAI	Yes	Python
AxonDeepSeg	CAE	TensorFlow	Yes (weights)	Python
BART	CAE, VAE	BART, TensorFlow	Yes	Python
Brainstorm	3D-CAE	Keras, TensorFlow	Yes (weights)	Python
CASCADE	1-D CNN	TensorFlow	Yes (weights)	Python
CDeep3M2	CAE	TensorFlow	Yes (weights)	Python
CERR	CAE	CERR	Yes	Octave, MATLAB
ClinicaDL	CNN, CAE	PyTorch	Yes	Python
DANNCE	3D-CNN	PyTorch	Yes (weights)	Python, MATLAB
DeepBehavior	CNN	TensorFlow	Yes (weights)	Python, MATLAB

Table 5. *Cont.*

Name	Models	DL Framework	Customization	Programming Language
DeepBhvTracking	CNN	TensorFlow	Yes (weights)	Python, MATLAB
DeepCINAC	DeepCINAC (CNN+LSTM)	Keras, TensorFlow	Yes (weights)	Python
DeepInfer	3D-CNN, CAE	Unspecified	Yes (weights)	Python, C++ (3D Slicer)
DeepLabCut	CNN	TensorFlow	Yes (weights)	Python
DeepLabStream	CNN	TensorFlow	Yes (weights)	Python
DeepNeuro	CNN, CAE, GAN	Keras, TensorFlow	Yes (weights, model)	Python
DeepNeuron	2,3-D CNN	Unspecified (Vaa3D)	Yes (weights)	C++
DeepPoseKit	CNN	Keras, TensorFlow	Yes (weights)	Python
DeLINEATE	CNN	Keras, TensorFlow	Yes (weights, model)	Python
DeepVOG	CAE	TensorFlow	No	Python
DLTK	CNN, CAE	Tensorflow	Yes (weights)	Python
DNNBrain	CNN	PyTorch	Yes (model)	Python
FastSurfer	CNN	PyTorch	Yes (weights)	Python (FreeSurfer)
fetal-code	2-D CNN	TensorFlow	No	Python
GaNDLF	CNN, CAE	PyTorch	Yes	Python
hipotalamus_seg	3D-CNN	Keras, TensorFlow	Yes (weights)	Python
ivadomed	2,3-D CNN, CAE	PyTorch	Yes (weights, model)	Python
LEAP, SLEAP	CNN, CAE	TensorFlow	Yes (weights, model)	Python
MARS, BENTO	CNN	TensorFlow	Yes (weights)	Python
MesoNet	CNN, CAE	Keras, TensorFlow	No	Python
MEYE	CAE, CNN	TensorFlow	Yes (model)	Python
MIScnn	2,3-D CNN	TensorFlow	Yes (weights)	Python
Neurite, Neuron	VAE	Keras, TensorFlow	Yes (weights)	Python
NiftyNet	CNN	TensorFlow	Yes	Python
NiftyTorch	CNN, CAE, GAN	PyTorch	Yes	Python
nnU-Net	2,3-D CAE	PyTorch	Yes	Python
PyTC	2,3-D CAE	PyTorch	Yes	Python
ScLimibic	3-D CAE	Neurite, TensorFlow	No	Python (FreeSurfer)
SimBA	CNN	TensorFlow	Yes (weights)	Python
SynthStrip	3-D CAE	PyTorch	No	Python
VesicleSeg	CNN	PyTorch	No	Python
Visual Fields Analysis	DeepLabCut	TensorFlow, DeepLabCut	Yes (weights)	Python
Volume Segmantics	CAE	PyTorch	Yes (weights)	Python
VoxelMorph, HyperMorph	CAE	TensorFlow	Yes (weights)	Python

Table 6. Technological aspects and code sources for the libraries and frameworks processing image data. The entry "(*)" signals missing data.

Name	Interface	Online/Offline	Maintenance	Source	Stars (Forks)	Contributors
Allen Cell Structure Segmenter	GUI, Jupyter Notebooks	Offline	Active	https://github.com/AllenCell/aics-ml-segmentation	24 (4)	2
ALMA	GUI	Offline	Active	https://github.com/sollan/alma	7 (1)	2
ANTsX (ANTsPyNet, ANTsRNet)	CLI	Offline	Active	https://github.com/ANTsX	961 (354)	55
ATLASS	Jupyter Notebooks	Offline	Inactive	https://github.com/adines/ATLASS	2 (1)	2
AxonDeepSeg	Jupyter Notebooks	Offline	Active	https://github.com/axondeepseg/axondeepseg	106 (28)	21
BART	CLI	Offline	Active	https://github.com/mrirecon/deep-deep-learning-with-bart	4 (0)	1
Brainstorm	CLI	Offline	Inactive	https://github.com/xamyzhao/brainstorm	387 (93)	1
CASCADE	GUI, Colab Notebooks	Offline	Active	https://github.com/HelmchenLabSoftware/Cascade	74 (26)	7
CDeep3M2	GUI, Colab Notebooks	Offline	Active	https://github.com/CRBS/cdeep3m2	4 (2)	2
CERR	GUI	Offline	Active	https://github.com/cerr/CERR	170 (95)	7
ClinicaDL	GUI, Colab Notebooks	Offline	Active	https://github.com/aramis-lab/clinicadl		
DANNCE	GUI	Offline	Inactive	https://github.com/spoonsso/dannce/	162 (23)	7
DeepBehavior	GUI	Offline	Inactive	https://github.com/aarac/DeepBehavior	29 (17)	1
DeepBhvTracking	GUI	Offline	Inactive	https://github.com/SunGL001/DeepBhvTracking	3 (0)	1
DeepCINAC	GUI, Colab Notebooks	Offline	Active	https://gitlab.com/cossartlab/deepcinac	7 (0)	10
DeepInfer	GUI	Offline	Active	http://www.deepinfer.org/	24 (14)	4
DeepLabCut	GUI, Colab Notebooks	Offline	Active	https://github.com/DeepLabCut/DeepLabCut	3600 (1500)	102
DeepLabStream	GUI	Online	Active	https://github.com/SchwarzNeuroconLab/DeepLabStream	45 (8)	5
DeepNeuro	CLI	Offline	Active	https://github.com/QTIM-Lab/DeepNeuro	113 (35)	3
DeepNeuron	GUI	Offline	Inactive	https://github.com/Vaa3D/vaa3d_tools/tree/master/hackathon/MK/DeepNeuron	92 (69)	69
DeepPoseKit	GUI	Offline	Inactive	https://github.com/jgraving/DeepPoseKit	353 (83)	6
DeLINEATE	GUI, Colab Notebooks	Offline	Active	bitbucket.org/delineate/delineate	(*)	3
DeepVOG	CLI	Offline	Inactive	https://github.com/pydsgz/DeepVOG	131 (58)	4
DLTK	CLI	Offline	Inactive	https://github.com/DLTK/DLTK	1400 (408)	6
DNNBrain	CLI	Offline	Active	https://github.com/BNUCNL/dnnbrain	37 (39)	10
FastSurfer	CLI	Offline	Active	https://github.com/Deep-MI/FastSurfer	338 (83)	12
fetal-code	GUI, Colab Notebooks	Offline	Active	https://github.com/saigerutherford/fetal-code	12 (5)	2
GaNDLF	GUI	Offline	Active	https://github.com/CBICA/GaNDLF	85 (53)	30
hipotalamus_seg	CLI	Offline	Active	https://github.com/BBillot/hypothalamus_seg	18 (5)	1
ivadomed	CLI	Offline	Active	https://github.com/ivadomed/ivadomed	146 (151)	34
LEAP, SLEAP	CLI, GUI, Colab Notebooks	Online	Active	https://github.com/talmolab/sleap	278 (61)	26
MARS, BENTO	GUI, MATLAB GUI, Jupyter Notebooks	Offline	Active	https://github.com/neuroethology	37 (8)	3
MesoNet	GUI, Colab Notebooks	Offline	Active	osf.io/svztu	(*)	3

Table 6. *Cont.*

Name	Interface	Online/Offline	Maintenance	Source	Stars (Forks)	Contributors
MEYE	Web app	Online, Offline	Active	pupillometry.it	21 (5)	2
MIScnn	Jupyter Notebook	Offline	Inactive	https://github.com/frankkramer-lab/MIScnn	357 (116)	6
Neurite, Neuron	CLI	Offline	Active	https://github.com/adalca/neurite	279 (59)	11
NiftyNet	CLI	Offline	Inactive	https://github.com/NifTK/NiftyNet	1300 (408)	41
NiftyTorch	CLI	Offline	Active	https://github.com/NiftyTorch/NiftyTorch.doc	34 (8)	3
nnU-Net	CLI	Offline	Active	https://github.com/MIC-DKFZ/nnUNet	3500 (1200)	38
PyTC	CLI	Offline	Active	https://github.com/zudi-lin/pytorch_connectomics	139 (67)	28
ScLimbic	CLI	Offline	Active	https://surfer.nmr.mgh.harvard.edu/fswiki/ScLimbic	(*)	(*)
SimBA	GUI	Offline	Active	https://github.com/sgoldenlab/simba	201 (115)	9
SynthStrip	CLI	Offline	Active	https://github.com/freesurfer/freesurfer/tree/dev/mri_synthstrip	(*)	(*)
VesicleSeg	GUI	Offline	Active	https://github.com/Imbrosci/synaptic-vesicles-detection	4 (3)	2
Visual Fields Analysis	GUI	Offline	Active	https://github.com/mathjoss/VisualFieldsAnalysis	1 (3)	4
Volume Segmantics	CLI, API	Offline	Active	https://github.com/DiamondLightSource/volume-segmantics	7 (3)	1
VoxelMorph, HyperMorph	CLI	Offline	Active	https://github.com/voxelmorph/voxelmorph	1800 (534)	13

4.3. Libraries for Further Data Types and Abstractions

Libraries for further data types and abstractions are shown in Table 7 (domains of application), Table 8 (models characteristics), Table 9 (technologies and sources). There are two ways libraries listed in this section differ from the previous ones. On the one hand, they can process data types that do not fall into the sequence or image category. On the other hand, they are the product of projects that transcend the training of deep learning models, i.e., larger scope frameworks, deep learning support functions (e.g., preprocessing and data augmentations pipelines), and services or infrastructure to host computational experiments. NeuroCAAS is an ambitious project that both standardizes experimental schedules and analyses and offers computational resources on the cloud. The platform lifts the burden of configuring and deploying data analysis tools, also guaranteeing replicability and readily available usage of pre-made pipelines with high efficiency. Other platform-oriented libraries are concerned with federated learning, i.e., the training of deep learning models on separated and private datasets, a relevant issue in healthcare. MONAI is a project that brings deep learning tools to many health and biology problems. The paradigm builds on PyTorch and aims at unifying healthcare AI practices throughout both academia and enterprise research, not only in the model development but also in the creation of shared annotated datasets. It also focuses on deployment and work in real-world clinical production, settling as a strong candidate for being the standard solution in the domain. Importantly, it is a commonly used framework for the 3D variations of UNet [105] lately dominating the yearly BraTS challenge [33] (see at http://braintumorsegmentation.org/). In this regard, nnU-Net is a narrower scope framework explicitly for building UNet-like models, focused on data-driven self-configuration of training hyperparameters, reducing the burden on researchers and practitioners. It was validated on 23 public data sets of biomedical interest with great success. PsychRNN, PyCog and THINGvision, as well as NeuroGym are libraries that bridge deep learning research and computational neuroscience. They are concerned with studying how artificial neural systems solve the same tasks that animal and human brains are subject to. The aims are those of simulating, modeling, learning representations, and reverse engineering cognition and behavior.

Table 7. Domains of applications for the libraries and frameworks for special applications.

Name	Neuroscience Area	Data Type	Datasets	Task
DANCE [106]	Single cell analysis	Gene sequences	External availability	Clustering, Classification, Prediction
PsychRNN [107]	Computational neuroscience	Sequences	No	Classification, Prediction, Cognitive tasks
PyCog [108]	Computational neuroscience	Sequences	No	Classification, Prediction, Cognitive tasks
THINGvision [109]	Computational neuroscience	Images, Text	External availability	Classification
TorchDIVA [110]	Speech production	Sequences	No	Audio synthesis
COINSTAC [111]	Neuroimaging	Img	No	Federated learning, Classification, Segmentation
Fed-BioMed [112]	Neuroimaging	Img	No	Federated learning, Classification, Aggregation
FeTS [113]	Neuroimaging	Img	No	Federated learning, Segmentation
MeDaS [114]	Neuroimaging	Img	No	Utilities, Classification, Segmentation, Object detection
MONAI [115]	General	General	External availability	General
NeuroCAAS [116]	General	General	External availability	General
NeuroGym [117]	Behavioral, cognitive neuroscience	General	Internal	Behavioral, cognitive task generation, evaluation
NiftyNet [93]	Neuroimaging	Img	No	Utilities, Classification, Segmentation, Regression, Synthesis
OpenFL [118]	General	General	No	Federated learning, Segmentation
pymia [119]	General	Img	No	Utilities (data handling, evaluations)
TorchIO [120]	Imaging	All images	No	Augmentation

Table 8. Model engineering specifications for the libraries and frameworks for special applications.

Name	Models	DL Framework	Customization	Programming Language
DANCE	GNN	PyTorch	Yes	Python
PsychRNN	RNN	Tensorflow	Yes	Python
PyCog	RNN	Theano	Yes	Python
THINGvision	CNN, RNN, Transformers	PyTorch, TensorFlow	No	Python
TorchDIVA	CNN	PyTorch	Yes (weights)	Python
COINCSTAC	CNN	COINSTAC	Yes	JavaScript, Python
Fed-BioMed	VAE	PyTorch	Yes (weights)	Python
FeTS	3D-ResUNet	PyTorch	Yes (weights)	Python
MeDaS	2,3-D CNN, CAE	PyTorch, TensorFlow	Yes	Python

Table 8. *Cont.*

Name	Models	DL Framework	Customization	Programming Language
MONAI	General	PyTorch	Yes	Python
NeuroCAAS	CNN	TensorFlow	Yes	Python
NeuroGym	RNN, General	Keras, TensorFlow, PyTorch	Yes	Python
NiftyNet	CNN, CAE, GAN	TensorFlow	Yes	Python
OpenFL	General	Keras, TensorFlow, PyTorch	Yes	Python
pymia	General	General	Yes	Python
TorchIO	CNN	PyTorch	Yes	Python

Table 9. Technological aspects and code sources for the libraries and frameworks for special applications. The entry "(*)" signals missing data.

Name	Interface	Online/Offline	Maintenance	Source	Stars (Forks)	Contributors
DANCE	CLI	Offline	Active	https://github.com/OmicsML/dance	206 (15)	8
PsychRNN	None	Offline	Active	https://github.com/murraylab/PsychRNN	122 (38)	8
PyCog	None	Offline	Inactive	https://github.com/xjwanglab/pycog	45 (29)	1
THINGvision	None	Offline	Active	github.com/ViCCo-Group/THINGSvision	108 (17)	12
TorchDIVA	CLI	Offline	Active	https://github.com/skinahan/DIVA_PyTorch	12 (0)	1
COINSTAC	GUI	Offline	Active	https://github.com/trendscenter/coinstac	35 (19)	19
Fed-BioMed	CLI	Offline	Active	https://gitlab.inria.fr/fedbiomed	6 (0)	26
FeTS	GUI, CLI	Offline	Active	https://github.com/FETS-AI/Front-End	55 (6)	2
MeDaS	GUI	Offline	Inactive	https://medas.bnc.org.cn/	(*)	(*)
MONAI	GUI, Colab Notebooks	Offline	Active	github.com/Project-MONAI/MONAI	4000 (776)	151
NeuroCAAS	GUI, Jupyter Notebooks	Offline	Active	github.com/cunningham-lab/neurocaas	25 (22)	6
NeuroGym	Colab Notebooks	Offline	Active	https://neurogym.github.io	(*)	4
NiftyNet	CLI	Offline	Inactive	https://github.com/NifTK/NiftyNet	1300 (408)	41
OpenFL	CLI	Offline	Active	https://github.com/securefederatedai/openfl	484 (134)	49
pymia	CLI	Offline	Active	https://github.com/rundherum/pymia	54 (12)	4
TorchIO	GUI, CLI	Offline	Active	torchio.rtfd.io	1700 (204)	49

5. Discussion

The use of deep learning in neuroscience research requires us to frame a study, or a part of it, as a machine learning problem or task. The setting may or may not require researchers to develop a novel learning architecture and algorithm, train an existing model on the data, or directly apply a trained model to the scientific use case. These situations require different levels of machine learning knowledge and data literacy. Consequently, different libraries and frameworks would be of use. In general, researchers should be aware of what mathematical objects and data objects can represent their experimental data: deep learning typically processes vectors, matrices, and tensors, which could represent signals, images, scalar and tensor fields, and more. Scientists should also consider the nature of data used to train models and how similar they are to use case data, hence how the model would perform on the latter. Importantly, there must be a specific task the model would perform in processing input data and providing the output, e.g., many problems are formulated as classification tasks, while others are very specific to the neuroscience domain. Deep learning models may retain general information from the training data but generally cannot apply it to different tasks without design adaptations. The application of deep learning to neuroscience is thus challenging. Working in teams with different scientific backgrounds is a possible solution for the problem of specialized expertise that deep learning requires. Leveraging the platforms and practices of open science and open source communities offers the support that a working team would still need. Deep learning has the potential hindrance of thriving on large data sets and powerful hardware, but transfer learning and the availability of "model zoos" in many libraries allow researchers to build on systems with great knowledge or efficacy from the start. These systems are not an end in themselves and can be integrated into larger workflows, allowing domain expertise and researchers' creativity to be enhancing and enhanced.

The tables in this work help to evaluate candidate tools for neuroscience research problems, providing information on the type and domain of data they process, the machine learning task they perform, and whether models are trainable, customizable, or frozen and ready for use.

The panorama of open-source libraries dedicated to deep learning applications in neuroscience is quite rich and diversified. There is a corpus of organized packages that integrate preprocessing, training, testing, and performance analyses of deep neural networks for neurological research. Most of these projects are tuned to specific data modalities and formats, but some libraries are quite versatile and customizable, and there are projects that encompass quantitative biology and medical analysis as a whole. There is a common tendency to develop GUIs, enhancing the user-friendliness of toolkits for non-programmers and researchers unacquainted with the command line interfaces, for example. Visualizations of building blocks, operations, and results allow us to focus on them without further allocating cognitive and time resources to producing the respective codes. Moreover, for the many libraries developed in Python, the (Jupyter) Notebook format appears as a widespread tool both for tutorials, documentation and as an interface to cloud computational resources (e.g., Google Colab [121]). Through notebooks, experimental templates can be modified and adapted in complete environments, where text instructions, dynamic visualization, and code are blended efficiently, can be shared and reproduced, Although learning curves depend on subjective experiences, the availability of such interfaces and instruments reduces the burden on the end user, enhancing accessibility. Apart from specific papers and documentation, and outside of deep learning per se, it is important to make researchers and developers aware of the main topics and initiatives in open culture and neuroinformatics in order to sustain the development of the field. For this reason, the interested reader is invited to rely on competent institutions (e.g., INCF) and databases of open resources (e.g., open-neuroscience) dedicated to neuroscience. Among the possibly missing technologies, the queries employed did not retrieve results in Natural Language Processing libraries dedicated to neuroscience, nor toolkits specifically employing Graph Neural Networks (GNNs), although available in EEG-DL. NLP is actually fundamental in healthcare since medical reports often come in non-standardized forms. Large Language Models (LLMs), Named Entity Recognition (NER) systems and text mining approaches in biomedical research exist [122,123]. GNNs comprise recent architectures that are extremely promising in a variety of fields [124], including biomedical research and particularly neuroscience [125,126]. Even if promising, their application is still less mature than that of computer vision models or time series analysis.

Future Directions

Overall, current libraries cover a wide range of applications, but it seems unlikely that a single deep learning framework could dominate the entire neuroscience field in the near future. Nonetheless, projects such as the PyTorch-based MONAI are strong candidates in unifying ecosystems for deep learning in medicine and biology. DeepLabCut and nnU-Net are also worth mentioning since they are widely applied as blueprints for newer applications, respectively, in pose estimation and segmentation. Investing efforts in open tools and data, accessible documentation, and modularity may guarantee useful results regardless of the specific application field and are indeed foundational to transferring successful tools across domains. Such effort should be paired with data literacy and basic knowledge of machine learning best practices (e.g., how to avoid data leakage between train and test sets, how to assess model performance), paired with digital frameworks with strong priors towards these experimental practices, such as NeuroCAAS. Interpretability and explainability of models depend not only on the researchers' theoretical understanding but also on transparent models and effective tools to open and shed light on black boxes. In fact, XAI is essential for systems that support decision-making in healthcare or experiments in biomedical sciences. Tools and libraries such as GaNDLF can be expected to gain value. Another relevant aspect of biomedicine is that of privacy and sensitive data. Records cannot be processed anywhere; personal data cannot be published anyhow. Standards and protocols to protect people's privacy and machine learning models and frameworks that respect them, such as federated learning, are expected to gain momentum. All developments should hold against the intrinsic multimodality of data from the field and the multidisciplinarity required to analyze them. In the end, the interplay of common practices and flexible models can be expected to be of central importance.

6. Conclusions

Although a large and growing number of repositories offer code to build specific models, as published in experimental papers, these resources seldom aim to constitute proper libraries or frameworks for research or clinical practice. Both deep learning and neuroscience gain much value even from sophisticated proofs of concept. In parallel, organized packages are spreading and starting to provide and integrate pre-processing, training, testing, and performance analyses of deep neural networks for neurological and biomedical research. This paper has offered both a historical and a technical context for the use of deep neural networks in neuroinformatics, focusing on open-source tools that scientists can comprehend and adapt to their necessities. At the same time, this work underlines the value of the open culture and points to relevant institutions and platforms for neuroscientists. Although the aim is not restricted to making clinicians develop their own deep models without coding or machine learning background, as was the case in [127], the overall effect of these libraries and sources is to democratize deep learning applications and results, as well as standardize such complex and varied models, supporting the research community in obtaining a proper means to an end and in envisioning then realizing collectively new projects and tools.

Author Contributions: Conceptualization and methodology, M.A. and L.F.T.; validation, L.F.T. and F.D.P.; writing—original draft preparation, L.F.T.; writing—review and editing, M.A., L.F.T., F.D.P., M.C.; supervision, M.A. and M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the "Department of excellence 2018–2022" initiative of the Italian Ministry of education (MIUR) awarded to the Department of Neuroscience—University of Padua.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANN	Artificial Neural Network
BCI	Brain-Computer Interface
BOLD	Blood-Oxygen-Level-Dependent
CAE	Convolutional AutoEncoder
CLI	Command Line Interface
CNN	Convolutional Neural Network
CT	Computed Tomography
DNN	Deep Neural Network
EEG	Electroencephalography
EM	Electron Microscopy
EMG	Electromiography
GAN	Generative Adversarial Network(s)
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
LSTM	Long-Short Term Memory
MLP	MultiLayer Perceptron
(rs-f)MRI	(resting state functional) Magnetic Resonance Imaging
NLP	Natural Language Processing
PET	Positron Emission Tomography
RNN	Recurrent Neural Network
SEM	Scanning Electron Microscopy
TEM	Transmission Electron Microscopy
VAE	Variational AutoEncoder
XAI	eXplainable Artificial Intelligence

References

1. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 386–408. [CrossRef] [PubMed]
2. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
3. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
4. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **1980**, *36*, 193–202. [CrossRef]
5. Minsky, M.; Papert, S. *Perceptrons: An Introduction to Computational Geometry*; MIT Press: Cambridge, MA, USA, 1969.
6. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Representations by Back-propagating Errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
7. Meijering, E. A bird's-eye view of deep learning in bioimage analysis. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 2312–2325. [CrossRef]
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Morehouse Lane Red Hook, NY, USA, 2012; pp. 1097–1105.
9. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]

10. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
11. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *arXiv* **2014**, arXiv:1409.0575.
12. Valliani, A.A.A.; Ranti, D.; Oermann, E.K. Deep Learning and Neurology: A Systematic Review. *Neurol. Ther.* **2019**, *8*, 351–365. [CrossRef]
13. Raymond, E.S. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, 2nd ed.; O'Reilly Media: Beijing, China; Cambridge, MA, USA; Farnham, UK; Köln, Germany; Paris, France; Sebastopol, CA, USA; Taip, Taiwan, 2001; p. 241.
14. Nvidia; Vingelmann, P.; Fitzek, F.H. CUDA, release: 10.2.89, 2020. Available online: https://developer.nvidia.com/cuda-toolkit (accessed on 22 November 2022).
15. Chetlur, S.; Woolley, C.; Vandermersch, P.; Cohen, J.M.; Tran, J.; Catanzaro, B.; Shelhamer, E. cuDNN: Efficient Primitives for Deep Learning. *arXiv* **2014**, arXiv:1410.0759.
16. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Available online: tensorflow.org (accessed on 22 November 2022).
17. Chollet, F. keras, 2015. Available online: https://github.com/fchollet/keras (accessed on 22 November 2022).
18. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Morehouse Lane Red Hook, NY, USA, 2019; pp. 8024–8035.
19. Nature Neuroscience. Available online: https://www.nature.com/subjects/neuroscience (accessed on 18 August 2022).
20. Brott, T.; Adams, H.; Olinger, C.; Marle, J.; Barsan, W.; Biller, J.; Spilker, J.; Holleran, R.; Eberle, R.; Hertzberg, V.; et al. Measurements of acute cerebral infarction: A clinical examination scale. *Stroke* **1989**, *20*, 864–870. [CrossRef] [PubMed]
21. What Is Neuroinformatics? Available online: https://www.incf.org/about/what-is-neuroinformatics (accessed on 18 August 2022).
22. Abrams, M.B.; Bjaalie, J.G.; Das, S.; Egan, G.F.; Ghosh, S.S.; Goscinski, W.J.; Grethe, J.S.; Kotaleski, J.H.; Ho, E.T.W.; Kennedy, D.N.; et al. A Standards Organization for Open and FAIR Neuroscience: the International Neuroinformatics Coordinating Facility. *Neuroinformatics* **2021**, *20*, 25–36. [CrossRef] [PubMed]
23. Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.W.; da Silva Santos, L.B.; Bourne, P.E.; et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **2016**, *3*, 160018. [CrossRef]
24. Fischl, B. FreeSurfer. *Neuroimage* **2012**, *62*, 774–781. [CrossRef]
25. Smith, S.M.; Jenkinson, M.; Woolrich, M.W.; Beckmann, C.F.; Behrens, T.E.J.; Johansen-Berg, H.; Bannister, P.R.; Luca, M.D.; Drobnjak, I.; Flitney, D.; et al. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage* **2004**, *23*, S208–S219. [CrossRef]
26. Flandin, G.; Friston, K. Statistical Parametric Mapping. *Scholarpedia* **2008**, *3*, 6232. [CrossRef]
27. Delorme, A.; Makeig, S. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J. Neurosci. Methods* **2004**, *134*, 9–21. [CrossRef]
28. Tadel, F.; Baillet, S.; Mosher, J.C.; Pantazis, D.; Leahy, R.M. Brainstorm: A User-Friendly Application for MEG/EEG Analysis. *Comput. Intell. Neurosci.* **2011**, *2011*, 1–13. [CrossRef]
29. Atzori, M.; Müller, H. PaWFE: Fast Signal Feature Extraction Using Parallel Time Windows. *Front. Neurorobot.* **2019**, *13*, 74. [CrossRef]
30. Gramfort, A.; Luessi, M.; Larson, E.; Engemann, D.A.; Strohmeier, D.; Brodbeck, C.; Goj, R.; Jas, M.; Brooks, T.; Parkkonen, L.; et al. MEG and EEG Data Analysis with MNE-Python. *Front. Neurosci.* **2013**, *7*, 1–13. [CrossRef] [PubMed]
31. Luca, T.; Gloria, B.; Stefano, T.; Emanuele, M. ROS-Neuro: An Open-Source Platform for Neurorobotics. *Front. Neurorobot.* **2022**, *16*, 886050. [CrossRef]
32. Hramov, A.E.; Maksimenko, V.A.; Pisarchik, A.N. Physical principles of brain–computer interfaces and their applications for rehabilitation, robotics and control of human brain states. *Phys. Rep.* **2021**, *918*, 1–133. [CrossRef]
33. Menze, B.H.; Jakab, A.; Bauer, S.; Kalpathy-Cramer, J.; Farahani, K.; Kirby, J.; Burren, Y.; Porz, N.; Slotboom, J.; Wiest, R.; et al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Trans. Med. Imaging* **2015**, *34*, 1993–2024. [CrossRef] [PubMed]
34. Winzeck, S.; Hakim, A.; McKinley, R.; Pinto, J.; Alves, V.; Silva, C.; Pisov, M.; Krivov, E.; Belyaev, M.; Monteiro, M.; et al. ISLES 2016 and 2017-benchmarking ischemic stroke lesion outcome prediction based on multispectral MRI. *Front. Neurol.* **2018**, *9*, 679.
35. Petzsche, M.R.H.; de la Rosa, E.; Hanning, U.; Wiest, R.; Pinilla, W.E.V.; Reyes, M.; Meyer, M.I.; Liew, S.L.; Kofler, F.; Ezhov, I.; et al. ISLES 2022: A multi-center magnetic resonance imaging stroke lesion segmentation dataset. *Sci. Data* **2022**, *9*, 762. [CrossRef]
36. Atzori, M.; Gijsberts, A.; Heynen, S.; Hager, A.G.M.; Deriaz, O.; van der Smagt, P.; Castellini, C.; Caputo, B.; Muller, H. Building the Ninapro database: A resource for the biorobotics community. In Proceedings of the 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), Rome, Italy, 24–27 June 2012; pp. 1258–1265.
37. Park, K.H.; Lee, S.W. Movement intention decoding based on deep learning for multiuser myoelectric interfaces. In Proceedings of the 2016 4th International Winter Conference on Brain-Computer Interface (BCI), Gangwon, Republic of Korea, 22–24 February 2016; pp. 1–2. [CrossRef]

38. Atzori, M.; Cognolato, M.; Müller, H. Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands. *Front. Neurorobot.* **2016**, *10*, 9. [CrossRef]
39. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
40. Cho, K.; van Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv* **2014**, arXiv:1409.1259. https://doi.org/10.48550/ARXIV.1409.1259.
41. Yamins, D.L.K.; DiCarlo, J.J. Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* **2016**, *19*, 356–365.
42. Van Rossum, G.; Drake, F.L. *Python 3 Reference Manual*; CreateSpace: Scotts Valley, CA, USA, 2009.
43. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2022.
44. Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V.B. Julia: A fresh approach to numerical computing. *SIAM Rev.* **2017**, *59*, 65–98. [CrossRef]
45. Innes, M. Flux: Elegant Machine Learning with Julia. *J. Open Source Softw.* **2018**, *3*, 602. [CrossRef]
46. Heilmeyer, F.A.; Schirrmeister, R.T.; Fiederer, L.D.J.; Völker, M.; Behncke, J.; Ball, T. A large-scale evaluation framework for EEG deep learning architectures. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 1039–1045.
47. Kuntzelman, K.M.; Williams, J.M.; Lim, P.C.; Samal, A.; Rao, P.K.; Johnson, M.R. Deep-Learning-Based Multivariate Pattern Analysis (dMVPA): A Tutorial and a Toolbox. *Front. Hum. Neurosci.* **2021**, *15*, 638052. [CrossRef] [PubMed]
48. Kostas, D.; Rudzicz, F. DN3: An open-source Python library for large-scale raw neurophysiology data assimilation for more flexible and standardized deep learning. *Neuroscience* **2020**. [CrossRef]
49. Hou, Y.; Zhou, L.; Jia, S.; Lun, X. A novel approach of decoding EEG four-class motor imagery tasks via scout ESI and CNN. *J. Neural Eng.* **2020**, *17*, 016048. [CrossRef] [PubMed]
50. Huang, Y.L.; Hsieh, C.Y.; Huang, J.X.; Wei, C.S. ExBrainable: An Open-Source GUI for CNN-based EEG Decoding and Model Interpretation. *arXiv* **2022**, arXiv:2201.04065.
51. Tayeb, Z.; Waniek, N.; Fedjaev, J.; Ghaboosi, N.; Rychly, L.; Widderich, C.; Richter, C.; Braun, J.; Saveriano, M.; Cheng, G.; et al. Gumpy: A Python toolbox suitable for hybrid brain–computer interfaces. *J. Neural Eng.* **2018**, *15*, 065003.
52. Fabietti, M.; Mahmud, M.; Lotfi, A.; Kaiser, M.S.; Averna, A.; Guggenmos, D.J.; Nudo, R.J.; Chiappalone, M.; Chen, J. SANTIA: a Matlab-based open-source toolbox for artifact detection and removal from extracellular neuronal signals. *Brain Inform.* **2021**, *8*, 14. [CrossRef]
53. Shenk, J.; Byttner, W.; Nambusubramaniyan, S.; Zoeller, A. Traja: A Python toolbox for animal trajectory analysis. *J. Open Source Softw.* **2021**, *6*, 3202. [CrossRef]
54. Luxem, K.; Mocellin, P.; Fuhrmann, F.; Kürsch, J.; Miller, S.R.; Palop, J.J.; Remy, S.; Bauer, P. Identifying behavioral structure from deep variational embeddings of animal motion. *Commun. Biol.* **2022**, *5*, 1267. [CrossRef]
55. Okuno, T.; Woodward, A. Vector Auto-Regressive Deep Neural Network: A Data-Driven Deep Learning-Based Directed Functional Connectivity Estimation Toolbox. *Front. Neurosci.* **2021**, *15*, 764796. [CrossRef] [PubMed]
56. Selvaraju, R.R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2016**, *128*, 336–359. [CrossRef]
57. Chen, J.; Ding, L.; Viana, M.P.; Lee, H.; Sluezwski, M.F.; Morris, B.; Hendershott, M.C.; Yang, R.; Mueller, I.A.; Rafelski, S.M. The Allen Cell and Structure Segmenter: A new open source toolkit for segmenting 3D intracellular structures in fluorescence microscopy images. *Cell Biol.* **2018**. [CrossRef]
58. Aljovic, A.; Zhao, S.; Chahin, M.; de la Rosa, C.; Van Steenbergen, V.; Kerschensteiner, M.; Bareyre, F.M. A deep learning-based toolbox for Automated Limb Motion Analysis (ALMA) in murine models of neurological disorders. *Commun. Biol.* **2022**, *5*, 131. [CrossRef]
59. Tustison, N.J.; Cook, P.A.; Holbrook, A.J.; Johnson, H.J.; Muschelli, J.; Devenyi, G.A.; Duda, J.T.; Das, S.R.; Cullen, N.C.; Gillen, D.L.; et al. The ANTsX ecosystem for quantitative biological and medical imaging. *Sci. Rep.* **2021**, *11*, 9068. [CrossRef]
60. Inés, A.; Domínguez, C.; Heras, J.; Mata, E.; Pascual, V. Biomedical image classification made easier thanks to transfer and semi-supervised learning. *Comput. Methods Programs Biomed.* **2021**, *198*, 105782. [CrossRef]
61. Zaimi, A. AxonDeepSeg: Automatic axon and myelin segmentation from microscopy data using convolutional neural networks. *Sci. Rep.* **2018**, *8*, 3816. [CrossRef]
62. Blumenthal, M.; Luo, G.; Schilling, M.; Holme, H.C.M.; Uecker, M. Deep, deep learning with BART. *Magn. Reson. Med.* **2023**, *89*, 678–693. [CrossRef]
63. Zhao, A.; Balakrishnan, G.; Durand, F.; Guttag, J.V.; Dalca, A.V. Data augmentation using learned transformations for one-shot medical image segmentation. *arXiv* **2019**, arXiv:1902.09383.
64. Rupprecht, P.; Carta, S.; Hoffmann, A.; Echizen, M.; Blot, A.; Kwan, A.C.; Dan, Y.; Hofer, S.B.; Kitamura, K.; Helmchen, F.; et al. A database and deep learning toolbox for noise-optimized, generalized spike inference from calcium imaging. *Nat. Neurosci.* **2021**, *24*, 1324–1337.
65. Haberl, M.G.; Churas, C.; Tindall, L.; Boassa, D.; Phan, S.; Bushong, E.A.; Madany, M.; Akay, R.; Deerinck, T.J.; Peltier, S.T.; et al. CDeep3M—Plug-and-Play cloud-based deep learning for image segmentation. *Nat. Methods* **2018**, *15*, 677–680.
66. Apte, A.P.; Iyer, A.; Thor, M.; Pandya, R.; Haq, R.; Jiang, J.; LoCastro, E.; Shukla-Dave, A.; Sasankan, N.; Xiao, Y.; et al. Library of deep-learning image segmentation and outcomes model-implementations. *Phys. Medica* **2020**, *73*, 190–196. [CrossRef] [PubMed]

67. Thibeau-Sutre, E.; Diaz, M.; Hassanaly, R.; Routier, A.; Dormont, D.; Colliot, O.; Burgos, N. ClinicaDL: An open-source deep learning software for reproducible neuroimaging processing. *Comput. Methods Programs Biomed.* **2022**, *220*, 106818. [CrossRef] [PubMed]
68. Dunn, T.W.; Marshall, J.D.; Severson, K.S.; Aldarondo, D.E.; Hildebrand, D.G.C.; Chettih, S.N.; Wang, W.L.; Gellis, A.J.; Carlson, D.E.; Aronov, D.; et al. Geometric deep learning enables 3D kinematic profiling across species and environments. *Nat. Methods* **2021**, *18*, 564–573. [CrossRef] [PubMed]
69. Arac, A.; Zhao, P.; Dobkin, B.H.; Carmichael, S.T.; Golshani, P. DeepBehavior: A Deep Learning Toolbox for Automated Analysis of Animal and Human Behavior Imaging Data. *Front. Syst. Neurosci.* **2019**, *13*, 20. [CrossRef]
70. Sun, G.; Lyu, C.; Cai, R.; Yu, C.; Sun, H.; Schriver, K.E.; Gao, L.; Li, X. DeepBhvTracking: A Novel Behavior Tracking Method for Laboratory Animals Based on Deep Learning. *Front. Behav. Neurosci.* **2021**, *15*, 750894. [CrossRef]
71. Denis, J.; Dard, R.F.; Quiroli, E.; Cossart, R.; Picardo, M.A. DeepCINAC: A Deep-Learning-Based Python Toolbox for Inferring Calcium Imaging Neuronal Activity Based on Movie Visualization. *eNeuro* **2020**, *7*, ENEURO.0038–20.2020. [CrossRef]
72. Mehrtash, A.; Pesteie, M.; Hetherington, J.; Behringer, P.A.; Kapur, T.; Wells, W.M.; Rohling, R.; Fedorov, A.; Abolmaesumi, P. *DeepInfer: Open-Source Deep Learning Deployment Toolkit for Image-Guided Therapy*; SPIE: Bellingham, WA, USA, 2017; p. 101351K. [CrossRef]
73. Nath, T.; Mathis, A.; Chen, A.C.; Patel, A.; Bethge, M.; Mathis, M.W. Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nat. Protoc.* **2019**, *14*, 2152–2176.
74. Schweihoff, J.F.; Loshakov, M.; Pavlova, I.; Kück, L.; Ewell, L.A.; Schwarz, M.K. DeepLabStream enables closed-loop behavioral experiments using deep learning-based markerless, real-time posture detection. *Commun. Biol.* **2021**, *4*, 130. [CrossRef]
75. Beers, A.; Brown, J.; Chang, K.; Hoebel, K.; Patel, J.; Ly, K.I.; Tolaney, S.M.; Brastianos, P.; Rosen, B.; Gerstner, E.R.; et al. DeepNeuro: An open-source deep learning toolbox for neuroimaging. *Neuroinformatics* **2021**, *19*, 127–140. [CrossRef]
76. Zhou, Z.; Kuo, H.C.; Peng, H.; Long, F. DeepNeuron: An open deep learning toolbox for neuron tracing. *Brain Inform.* **2018**, *5*, 3. [CrossRef] [PubMed]
77. Graving, J.M.; Chae, D.; Naik, H.; Li, L.; Koger, B.; Costelloe, B.R.; Couzin, I.D. DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife* **2019**, *8*, e47994. [CrossRef] [PubMed]
78. Yiu, Y.H.; Aboulatta, M.; Raiser, T.; Ophey, L.; Flanagin, V.L.; zu Eulenburg, P.; Ahmadi, S.A. DeepVOG: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *J. Neurosci. Methods* **2019**, *324*, 108307. [CrossRef] [PubMed]
79. Pawlowski, N.; Ktena, S.I.; Lee, M.C.H.; Kainz, B.; Rueckert, D.; Glocker, B.; Rajchl, M. DLTK: State of the Art Reference Implementations for Deep Learning on Medical Images. *arXiv* **2017**, arXiv:1711.06853.
80. Chen, X.; Zhou, M.; Gong, Z.; Xu, W.; Liu, X.; Huang, T.; Zhen, Z.; Liu, J. DNNBrain: A Unifying Toolbox for Mapping Deep Neural Networks and Brains. *Front. Comput. Neurosci.* **2020**, *14*, 580632. [CrossRef]
81. Henschel, L.; Conjeti, S.; Estrada, S.; Diers, K.; Fischl, B.; Reuter, M. FastSurfer—A fast and accurate deep learning based neuroimaging pipeline. *NeuroImage* **2020**, *219*, 117012. [CrossRef]
82. Rutherford, S.; Sturmfels, P.; Angstadt, M.; Hect, J.; Wiens, J.; van den Heuvel, M.I.; Scheinost, D.; Sripada, C.; Thomason, M. Automated Brain Masking of Fetal Functional MRI with Open Data. *Neuroinformatics* **2021**, *20*, 173–185. [CrossRef]
83. Pati, S.; Thakur, S.P.; Bhalerao, M.; Thermos, S.; Baid, U.; Gotkowski, K.; Gonzalez, C.; Guley, O.; Hamamci, I.E.; Er, S.; et al. GaNDLF: A Generally Nuanced Deep Learning Framework for Scalable End-to-End Clinical Workflows in Medical Imaging. *arXiv* **2021**, arXiv: 2103.01006.
84. Billot, B.; Bocchetta, M.; Todd, E.; Dalca, A.V.; Rohrer, J.D.; Iglesias, J.E. Automated segmentation of the hypothalamus and associated subunits in brain MRI. *NeuroImage* **2020**, *223*, 117287. [CrossRef]
85. Gros, C.; Lemay, A.; Vincent, O.; Rouhier, L.; Bucquet, A.; Cohen, J.P.; Cohen-Adad, J. ivadomed: A Medical Imaging Deep Learning Toolbox. *arXiv* **2020**, arXiv: 2010.09984.
86. Pereira, T.D.; Aldarondo, D.E.; Willmore, L.; Kislin, M.; Wang, S.S.H.; Murthy, M.; Shaevitz, J.W. Fast animal pose estimation using deep neural networks. *Nat. Methods* **2019**, *16*, 117–125. [CrossRef] [PubMed]
87. Pereira, T.D.; Tabris, N.; Matsliah, A.; Turner, D.M.; Li, J.; Ravindranath, S.; Papadoyannis, E.S.; Normand, E.; Deutsch, D.S.; Wang, Z.Y.; et al. SLEAP: A deep learning system for multi-animal pose tracking. *Nat. Methods* **2022**, *19*, 486–495. [CrossRef] [PubMed]
88. Segalin, C.; Williams, J.; Karigo, T.; Hui, M.; Zelikowsky, M.; Sun, J.J.; Perona, P.; Anderson, D.J.; Kennedy, A. The Mouse Action Recognition System (MARS) software pipeline for automated analysis of social behaviors in mice. *eLife* **2021**, *10*, e63720.
89. Xiao, D.; Forys, B.J.; Vanni, M.P.; Murphy, T.H. MesoNet allows automated scaling and segmentation of mouse mesoscale cortical maps using machine learning. *Nat. Commun.* **2021**, *12*, 5992. [CrossRef] [PubMed]
90. Mazziotti, R.; Carrara, F.; Viglione, A.; Lupori, L.; Lo Verde, L.; Benedetto, A.; Ricci, G.; Sagona, G.; Amato, G.; Pizzorusso, T. MEYE: Web App for Translational and Real-Time Pupillometry. *eNeuro* **2021**, *8*, ENEURO.0122–21.2021. [CrossRef] [PubMed]
91. Müller, D.; Kramer, F. MIScnn: A framework for medical image segmentation with convolutional neural networks and deep learning. *BMC Med. Imaging* **2021**, *21*, 12. [CrossRef]
92. Dalca, A.V.; Guttag, J.; Sabuncu, M.R. Anatomical Priors in Convolutional Networks for Unsupervised Biomedical Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 9290–9299. [CrossRef]

93. Gibson, E.; Li, W.; Sudre, C.; Fidon, L.; Shakir, D.I.; Wang, G.; Eaton-Rosen, Z.; Gray, R.; Doel, T.; Hu, Y.; et al. NiftyNet: A deep-learning platform for medical imaging. *Comput. Methods Programs Biomed.* **2018**, *158*, 113–122. [CrossRef]
94. Subramanian, A.; Lan, H.; Govindarajan, S.; Viswanathan, L.; Choupan, J.; Sepehrband, F. NiftyTorch: A Deep Learning framework for NeuroImaging. *Neuroscience* **2021**. [CrossRef]
95. Isensee, F.; Jaeger, P.F.; Kohl, S.A.A.; Petersen, J.; Maier-Hein, K.H. nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. *Nat. Methods* **2021**, *18*, 203–211. [CrossRef]
96. Lin, Z.; Wei, D.; Lichtman, J.; Pfister, H. PyTorch Connectomics: A Scalable and Flexible Segmentation Framework for EM Connectomics. *arXiv* **2021**, arXiv:2112.05754.
97. Greve, D.N.; Billot, B.; Cordero, D.; Hoopes, A.; Hoffmann, M.; Dalca, A.V.; Fischl, B.; Iglesias, J.E.; Augustinack, J.C. A deep learning toolbox for automatic segmentation of subcortical limbic structures from MRI images. *NeuroImage* **2021**, *244*, 118610. [CrossRef] [PubMed]
98. Nilsson, S.R.; Goodwin, N.L.; Choong, J.J.; Hwang, S.; Wright, H.R.; Norville, Z.C.; Tong, X.; Lin, D.; Bentzley, B.S.; Eshel, N.; et al. Simple Behavioral Analysis (SimBA)—an open source toolkit for computer classification of complex social behaviors in experimental animals. *bioRxiv* **2020**.
99. Hoopes, A.; Mora, J.S.; Dalca, A.V.; Fischl, B.; Hoffmann, M. SynthStrip: Skull-stripping for any brain image. *NeuroImage* **2022**, *260*, 119474. [CrossRef] [PubMed]
100. Imbrosci, B.; Schmitz, D.; Orlando, M. Automated Detection and Localization of Synaptic Vesicles in Electron Microscopy Images. *eNeuro* **2022**, *9*, ENEURO.0400-20.2021.
101. Josserand, M.; Rosa-Salva, O.; Versace, E.; Lemaire, B.S. Visual Field Analysis: A reliable method to score left and right eye use using automated tracking. *Behav. Res. Methods* **2021**, *54*, 1715–1724. [CrossRef]
102. King, O.N.F.; Bellos, D.; Basham, M. Volume Segmantics: A Python Package for SemanticSegmentation of Volumetric Data Using Pre-trained PyTorch Deep LearningModels. *J. Open Source Softw.* **2022**, *7*, 4691. [CrossRef]
103. Balakrishnan, G.; Zhao, A.; Sabuncu, M.R.; Guttag, J.; Dalca, A.V. VoxelMorph: A Learning Framework for Deformable Medical Image Registration. *IEEE Trans. Med. Imaging* **2019**, *38*, 1788–1800.
104. Hoopes, A.; Hoffmann, M.; Fischl, B.; Guttag, J.; Dalca, A.V. HyperMorph: Amortized Hyperparameter Learning for Image Registration. *arXiv* **2021**, arXiv:2101.01035.
105. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
106. Ding, J.; Wen, H.; Tang, W.; Liu, R.; Li, Z.; Venegas, J.; Su, R.; Molho, D.; Jin, W.; Zuo, W.; et al. DANCE: A Deep Learning Library and Benchmark for Single-Cell Analysis. *bioRxiv* **2022**. [CrossRef]
107. Ehrlich, D.B.; Stone, J.T.; Brandfonbrener, D.; Atanasov, A.; Murray, J.D. PsychRNN: An Accessible and Flexible Python Package for Training Recurrent Neural Network Models on Cognitive Tasks. *eNeuro* **2021**, *8*, ENEURO.0427–20.2020. [CrossRef] [PubMed]
108. Song, H.F.; Yang, G.R.; Wang, X.J. Training Excitatory-Inhibitory Recurrent Neural Networks for Cognitive Tasks: A Simple and Flexible Framework. *PLoS Comput. Biol.* **2016**, *12*, e1004792. [CrossRef] [PubMed]
109. Muttenthaler, L.; Hebart, M.N. THINGSvision: A Python Toolbox for Streamlining the Extraction of Activations from Deep Neural Networks. *Front. Neuroinform.* **2021**, *15*, 679838. [CrossRef] [PubMed]
110. Kinahan, S.; Liss, J.; Berisha, V. TorchDIVA: An Extensible Computational Model of Speech Production built on an Open-Source Machine Learning. *PLoS ONE* **2023**, *18*, e0281306. [CrossRef]
111. Rootes-Murdy, K.; Gazula, H.; Verner, E.; Kelly, R.; DeRamus, T.; Plis, S.; Sarwate, A.; Turner, J.; Calhoun, V. Federated Analysis of Neuroimaging Data: A Review of the Field. *Neuroinformatics* **2022**, *20*, 377–390. [CrossRef] [PubMed]
112. Silva, S.; Altmann, A.; Gutman, B.; Lorenzi, M. Fed-BioMed: A General Open-Source Frontend Framework for Federated Learning in Healthcare. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*; Lecture Notes in Computer Science; Albarqouni, S., Bakas, S., Kamnitsas, K., Cardoso, M.J., Landman, B., Li, W., Milletari, F., Rieke, N., Roth, H., Xu, D., et al.; Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 12444, pp. 201–210.
113. Pati, S.; Baid, U.; Edwards, B.; Sheller, M.J.; Foley, P.; Anthony Reina, G.; Thakur, S.; Sako, C.; Bilello, M.; Davatzikos, C.; et al. The federated tumor segmentation (FeTS) tool: An open-source solution to further solid tumor research. *Phys. Med. Biol.* **2022**, *67*, 204002. [CrossRef]
114. Zhang, L.; Li, J.; Li, P.; Lu, X.; Gong, M.; Shen, P.; Zhu, G.; Shah, S.A.; Bennamoun, M.; Qian, K.; et al. MEDAS: An open-source platform as a service to help break the walls between medicine and informatics. *Neural Comput. Appl.* **2022**, *34*, 6547–6567. [CrossRef]
115. Cardoso, M.J.; Li, W.; Brown, R.; Ma, N.; Kerfoot, E.; Wang, Y.; Murrey, B.; Myronenko, A.; Zhao, C.; Yang, D.; et al. MONAI: An open-source framework for deep learning in healthcare. *arXiv* **2022**, arXiv:2211.02701.
116. Abe, T.; Kinsella, I.; Saxena, S.; Buchanan, E.K.; Couto, J.; Briggs, J.; Kitt, S.L.; Glassman, R.; Zhou, J.; Paninski, L.; et al. Neuroscience Cloud Analysis As a Service: An open-source platform for scalable, reproducible data analysis. *Neuron* **2022**, *110*, 2771–2789. [CrossRef]
117. Molano-Mazon, M.; Barbosa, J.; Pastor-Ciurana, J.; Fradera, M.; Zhang, R.Y.; Forest, J.; del Pozo Lerida, J.; Ji-An, L.; Cueva, C.J.; de la Rocha, J.; et al. NeuroGym: An open resource for developing and sharing neuroscience tasks. *PsyArXiv* **2022**. [CrossRef]
118. Reina, G.A.; Gruzdev, A.; Foley, P.; Perepelkina, O.; Sharma, M.; Davidyuk, I.; Trushkin, I.; Radionov, M.; Mokrov, A.; Agapov, D.; et al. OpenFL: An open-source framework for Federated Learning. *Phys. Med. Biol.* **2022**, *67*, 214001.

119. Jungo, A.; Scheidegger, O.; Reyes, M.; Balsiger, F. pymia: A Python package for data handling and evaluation in deep learning-based medical image analysis. *Comput. Methods Programs Biomed.* **2021**, *198*, 105796. [CrossRef] [PubMed]
120. Pérez-García, F.; Sparks, R.; Ourselin, S. TorchIO: A Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. *Comput. Methods Programs Biomed.* **2021**, *208*, 106236. [CrossRef] [PubMed]
121. Bisong, E. Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*; Apress: Berkeley, CA, USA, 2019; pp. 59–64. [CrossRef]
122. Basyal, G.P.; Rimal, B.P.; Zeng, D. A Systematic Review of Natural Language Processing for Knowledge Management in Healthcare. *arXiv* **2020**, arXiv:2007.09134.
123. Locke, S.; Bashall, A.; Al-Adely, S.; Moore, J.; Wilson, A.; Kitchen, G.B. Natural language processing in medicine: A review. *TRends Anaesth. Crit. Care* **2021**, *38*, 4–9. [CrossRef]
124. Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; Sun, M. Graph Neural Networks: A Review of Methods and Applications. *arXiv* **2018**, arXiv:1812.08434.
125. Zhang, X.M.; Liang, L.; Liu, L.; Tang, M.J. Graph Neural Networks and Their Current Applications in Bioinformatics. *Front. Genet.* **2021**, *12*, 690049. [CrossRef]
126. Li, M.M.; Huang, K.; Zitnik, M. Graph Representation Learning in Biomedicine. *arXiv* **2022**, arXiv:2104.04883.
127. Faes, L.; Wagner, S.K.; Fu, D.J.; Liu, X.; Korot, E.; Ledsam, J.R.; Back, T.; Chopra, R.; Pontikos, N.; Kern, C.; et al. Automated deep learning design for medical image classification by health-care professionals with no coding experience: A feasibility study. *Lancet Digit. Health* **2019**, *1*, e232–e242. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.