A Survey on Deep Learning Toolkits and Libraries for Intelligent User Interfaces

Jan Zacharias
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
jan.zacharias@dfki.de

Michael Barz
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
michael.barz@dfki.de

Daniel Sonntag
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
daniel.sonntag@dfki.de

arXiv:1803.04818v2 [cs.HC] 14 Mar 2018

ABSTRACT
This paper provides an overview of prominent deep learning toolkits and, in particular, reports on recent publications that contributed open source software for implementing tasks that are common in intelligent user interfaces (IUI). We provide a scientific reference for researchers and software engineers who plan to utilise deep learning techniques within their IUI research and development projects.

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation (e.g. HCI): User Interfaces

Author Keywords
Artificial intelligence; Machine learning; Deep learning; Interactive machine learning; Hyper-parameter tuning; Convolutional neural networks

INTRODUCTION

Intelligent user interfaces (IUIs) aim to incorporate intelligent automated capabilities in human-computer interaction (HCI), where the net impact is an interaction that improves performance or usability in critical ways. Deep learning techniques can be used in an IUI to implement artificial intelligence (AI) components that effectively leverage human skills and capabilities, so that human performance with an application excels [54].

Many IUIs, especially in the smartphone domain, use multiple input and output modalities for more efficient, flexible and robust user interaction [44]. They allow users to select a suitable input mode, or to shift among modalities as needed during the changing physical contexts and demands of continuous mobile use.

Deep learning has the potential to increase this flexibility with user and adaptation models that support speech, pen, (multi-) touch, gestures and gaze as modalities and can learn the appropriate alignment of them for mutual disambiguation. This ensures a higher precision in understanding the user input and to overcome the limitations of individual signal or interaction modalities [41].

Deep learning systems are able to process very complex real-world input data by using a nested hierarchy of concepts with increasing complexity for its representation [25]. Multimodal IUIs can greatly benefit from deep learning techniques because the complexity inherent in high-level event detection and multimodal signal processing can be modelled by likewise complex deep network structures and efficiently trained with nowadays available GPU-infrastructures. Especially recurrent model architectures are suitable for processing sequential multimodal signal streams, e.g., for natural dialogues and long-term autonomous agents [52].

This paper provides IUI researchers and practitioners with an overview of deep learning toolkits and libraries that are available under an open source license and describes how they can be applied for intelligent multimodal interaction (considering the modules of the architecture in figure 1 for classification). Related deep learning surveys are in the medical domain[21] or focus on the techniques [45, 4, 50].

A major challenge common to all AI systems is to move from closed to open world settings. Superhuman performance in one environment can lead to unexpected behaviour and dangerous situations in another:

> »The 'intelligence' of an artificial intelligence system can be deep but narrow.« [17]

An implication is the need for efficient learning methods and adaptive models for long-term autonomous systems. Promising techniques include active learning [48], reinforcement learning [27], interactive machine learning [3] and machine teaching [51]. We motivate the use of interactive training approaches enabling efficient and continuous updates to machine learning models. This is particularly useful for enhancing user interaction because it enables robust processing of versatile signal streams and joint analysis of data from multiple modalities and sensors to understand complex user behaviour.

TOOLKITS AND LIBRARIES

We briefly introduce the most popular frameworks used for implementing deep learning algorithms (summarised in table 2.1). We include information about licenses and supported programming languages for quick compatibility or preference

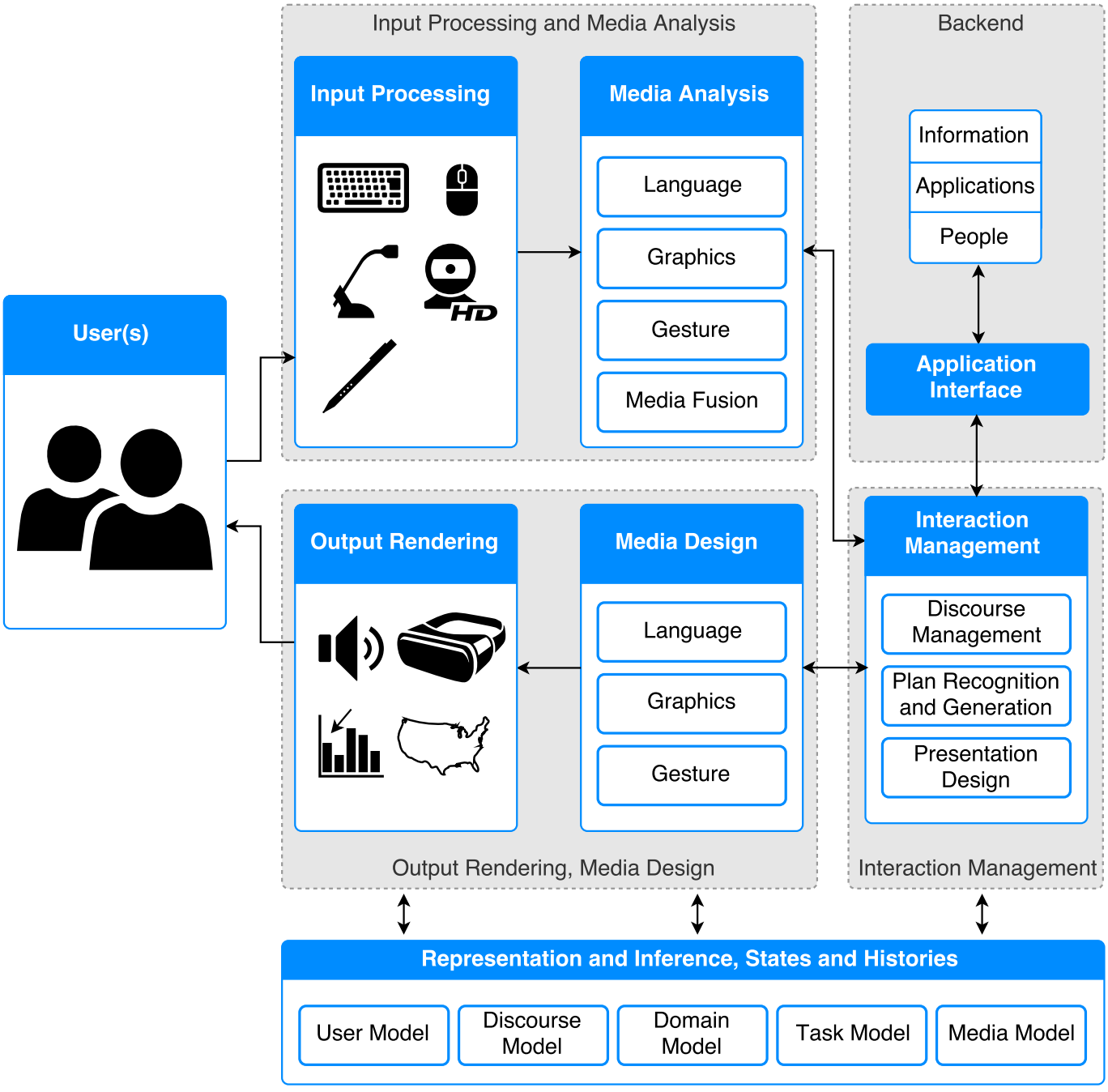

Figure 1. Categorization of intelligent user interface key components, based on the conceptual architecture [59] and DFKI's Smartweb system [55].

checks. Then, we present and qualitatively evaluate open source contributions that are based on these frameworks and that implement key elements of an IUI. Works are grouped into categories in alignment to an adapted version of the high-level IUI architecture as depicted in figure 1. Other ways of applying the described systems are certainly possible and appropriate depending on the use case at hand.

Deep Learning Frameworks

The core of most deep learning contributions are sophisticated machine learning frameworks like TensorFlow [1] and Caffe [30]. Most of these software packages are published as open source, allowing independent review, verification, and democratised usage in IUI projects. This does not imply that there are no errors in current implementations, however, the open character enables everybody to search for and eventually identify the root cause of wrong results or poor performance. Similarly, new features can be proposed and contributed. Table 2.1 lists toolkits and libraries that are a representative selection of available open source solutions. The table is sorted by a popularity rating proposed by François Chollet, the author of the Keras deep learning toolkit. GitHub metrics are used and weighted by coefficients such that the relative correlation of each metric reflects the number of users. To model the factors, Chollet informally took into account Google Analytics data for the Keras project website, PyPI download data, ArXiv mentions data as well as Google Trends data among other sources. The rating is calculated as the sum of the GitHub Contributions$\times$30, Issues$\times$20, Forks$\times$3 and the Stars, scaled to 100% defined by the top-scorer TensorFlow as a benchmark. While the exact numbers have been chosen manually, like in an ensemble model the relative order of magnitude of the coefficients matter beside the fact that multiple data sources are used.

Deep Learning in IUIs

One can find a multitude of deep learning software on the web and it is unclear whether these packages are useful and easy to setup in an IUI project at a first glance. With this work, we want to shed light on selected open source contributions by providing an overview and sharing our experiences in terms of utility and ease of use. We group related works based on the long acknowledged conceptual architecture by Wahlster and Maybury [59] and its reference implementation [55] which defines essential parts and modules of IUIs (see figure 1).

We consider the functional coherent elements *Input Processing & Media Analysis*; *Interaction Management, Output Rendering & Media Design* and *Backend* services as the main building blocks. Many works can be applied at different stages or implement multiple roles at once, particularly deep learning systems that are trained end-to-end. Inference mechanisms and the representation of the current "information state" and histories of meta models are shared across all IUI components.

Input Processing & Media Analysis

Components that implement this role help in analysing and understanding the user by including available modalities and fusing them, if appropriate. Examples are IUIs that require a natural language understanding (NLU) component to extract structured semantic information from unstructured text, e.g., for classifying the user's intent and extracting entities.

Hauswald et al. [28] implemented the Tonic Suite that offers different classification services as a service based on DjiNN, an infrastructure for deep neural networks (DNN). It can process visual stimuli for image classification based on AlexNet [36] and face recognition by replicating Facebook's DeepFace [57]. Natural language input is supported in terms of digit recognition based on MNIST [37] and automated speech recognition (ASR) based on the Kaldi[1] ASR toolkit [46] trained on the VoxForge[2] open-source large scale speech corpora. Further, several natural language processing techniques are available: part-of-speech tagging, named entity recognition and word chunking—all based on neural networks first introduced by NEC's SENNA[3] project. The Tonic suite relies on the Caffe framework.

The BSD 3-Clause licensed C++ source code[4] is however inconsistently annotated and TODO's to improve the documentation quality remain unresolved. Sufficient installation instructions for DjiNN and Tonic are separately hosted[5,6]. Both software packages have not been updated since 2015, rely on an outdated Caffe version and have many other legacy dependencies, thus extending installation time considerably.

Relation extraction is an important part of natural language processing (NLP): relational facts are extracted from plain text. For instance, this task is required in the automated population and extension of a knowledge base from user input. Lin et al. [40] released a neural relation extraction implementation[7] under the MIT license. The same repository hosts the re-implementation of the relation extractors of Zeng et al. [63, 62]. The C++ code is completely undocumented, the ReadMe file lists comparative results whereas required resources are not directly accessible and dependencies are not listed. More recently the authors published a better documented relation extractor [8] inspired by [40, 64] which uses TensorFlow and is written in Python—the code is annotated and dependencies are listed appropriately. Similarly, Nguyen and Grishman [42] proposed the combination of convolutional and recurrent neural networks in hybrid models via majority voting for relation extraction. The authors published their source code[9], which is based on the Theano framework, to allow others to verify and potentially improve on their method concerning the ACE 2005 Corpus[10]. The Python source does not contain licensing information and is not annotated. The ReadMe file outlines how the evaluation can be performed and states that the dataset needs to be procured separately, incurring a $4000 fee.

[1] https://github.com/kaldi-asr/kaldi
[2] http://www.voxforge.org
[3] http://ml.nec-labs.com/senna
[4] https://github.com/claritylab/djinn
[5] http://djinn.clarity-lab.org/djinn/
[6] http://djinn.clarity-lab.org/tonic-suite/
[7] https://github.com/thunlp/NRE
[8] https://github.com/thunlp/TensorFlow-NRE
[9] https://github.com/anoperson/DeepIE
[10] https://catalog.ldc.upenn.edu/ldc2006t06

Name	Website	GitHub URL	License	Language	APIs	Rating [%]
TensorFlow [1]	http://tensorflow.org	tensorflow/tensorflow	Apache-2.0	C++, Python	Python, C++, Java, Go	100
Keras [14]	http://keras.io/	fchollet/keras	MIT	Python	Python, R	46.1
Caffe [30]	http://caffe.berkeleyvision.org	BVLC/caffe	BSD	C++	Python, MATLAB	38.1
MXNet [13]	http://mxnet.io	apache/incubator-mxnet	Apache-2.0	C++	Python, Scala, R, JavaScript, Julia, MATLAB, Go, C++, Perl	34
Theano [2]	http://deeplearning.net/software/theano	Theano/Theano	BSD	Python	Python	19.3
CNTK [61]	https://docs.microsoft.com/en-us/cognitive-toolkit	Microsoft/CNTK	MIT	C++	Python, C++, C#, Java	18.4
DeepLearning4J [20]	https://deeplearning4j.org	deeplearning4j/deeplearning4j	Apache-2.0	Java, Scala	Java, Scala, Clojure, Kotlin	17.8
PaddlePaddle	http://www.paddlepaddle.org	baidu/paddle	Apache-2.0	C++	C++	16.3
PyTorch	http://pytorch.org	pytorch/pytorch	BSD	C++, Python	Python	14.3
Chainer [58]	https://chainer.org	pfnet/chainer	MIT	Python	Python	7.9
Torch7 [16]	http://torch.ch/	torch/torch7	BSD	C, Lua	C, Lua, LuaJIT	7.8
DIGITS [60]	https://developer.nvidia.com/digits	NVIDIA/DIGITS	BSD	Python	REST/Json	7.8
TFLearn [18]	http://tflearn.org	tflearn/tflearn	MIT	Python	Python	7.5
Caffe2	https://caffe2.ai	caffe2/caffe2	Apache-2.0	C++, Python	Python, C++	7.4
dlib [35]	http://dlib.net	davisking/dlib	Boost	C++	C++, Python	5.7

Table 1. Open source software overview with rating based on GitHub metrics

Chen and Manning proposed a dependency parser using neural networks that analyses the grammatical structure of sentences and tries to establish relationships between "head" words and words which modify those heads [12]. The software is part of the Stanford Parser[11] and CoreNLP,[12] a Java toolkit for NLP which allows the computer to analyse, understand, alter or generate natural language. Consequently, CoreNLP relates also to the media design element of an IUI. Building instructions are included in the ReadMe file and a well-written HTML documentation is available[13].

The open source contribution RASA_NLU[14] [8], written in Python and published under the Apache-2.0 license, performs natural language understanding with intent classification and entity extraction. Several machine learning backends can be employed: spaCy[15] which uses thinc[16], a deep learning capable library, MITIE[17] with dlib as backend and scikit-learn[18]. When using docker[19] the installation is noteworthy simple. A single command downloads all required components and dependencies and starts the container with the NLU service in minutes (depending on the internet connection). A complete installation and getting started guide is available as ReadMe. As the source code is consistently annotated, an automatically generated Sphinx[20] HTML documentation is accessible[21]. Short questions can be asked in a gitter chat[22].

In pervasive computing, understanding how the user interacts with the environment is essential. Bertasius et al. [6] designed a deep neural network model with Caffe[23] that processes the user's visual and tactile interactions for identifying the action object. The authors provide a pre-trained model and the Python code that predicts the area of the action object in an RGB(D) image, i.e., depth information is optional and can be used to improve the accuracy. Unfortunately, the annotated dataset that was created by the authors to train and test the model remains unpublished, thus preventing complete verification of the results and model enhancements by third parties. The published source is thoroughly annotated but lacks licensing information and detailed dependency information. In a follow-up publication, Bertasius et al. [7] demonstrated that the supervised creation of the training dataset can be omitted by using segmentation and recognition agents, implemented as cross-pathway architecture in a visual-spatial network (VSN).

[11] https://nlp.stanford.edu/software/lex-parser.html
[12] https://github.com/stanfordnlp/CoreNLP
[13] https://stanfordnlp.github.io/CoreNLP/
[14] https://github.com/RasaHQ/rasa_nlu
[15] https://github.com/explosion/spaCy
[16] https://github.com/explosion/thinc
[17] https://github.com/mit-nlp/MITIE
[18] https://github.com/scikit-learn/scikit-learn
[19] https://www.docker.com/
[20] http://www.sphinx-doc.org
[21] https://rasahq.github.io/rasa_nlu/master/
[22] https://gitter.im/RasaHQ/rasa_nlu
[23] https://github.com/gberta/EgoNet

Unsupervised learning is desirable, because the training of the action-object detection model requires pixelwise annotation of captured images by humans, which is time-intensive and costly. The implemented VSN learns to detect likely action-objects from unlabelled egocentrically captured image data. The GitHub repository[24] contains pre-trained models, the Python code to perform predictions and matlab sources for the multiscale combinatorial grouping that is required for the segmentation agent. Similarly to the first contribution, the code lacks a license while code annotations are sufficient. The ReadMe file contains general training instructions, the actual setup of the training toolchain is very cumbersome as merely pointers are given.

In [5] Barz and Sonntag used Caffe to combine gaze and egocentric camera data with GPU based object classification and attention detection for the construction of episodic memories of egocentric events in real-time. The code is not yet available, but will be published as a plug-in for the Pupil Labs eye tracking suite [34][25] in mid of 2018.

Interaction Management, Output Rendering & Media Design

User input is interpreted with regard to the current state of considered models, e.g., the discourse context, in order to identify and plan future actions and design appropriate IUI output. Components described here implement a central dialogue management functionality and generate outputs for the users in terms of multimodal natural language generation (NLG).

RASA_CORE[26] [8] is an open source discourse/dialogue manager framework which is written in Python and uses Keras' LSTM implementation in order to allow contextual, layered conversations. While no docker image is available, the documentation[27] quality is on par with RASA_NLU, a chat is analogously available[28]. Four example chatbots are provided[29] for getting started easily as well as a number of tutorials[30]. In an evaluation, Braun et al. [9] found the RASA ensemble to score second best against commercial closed source systems. Note that RASA_CORE also fits the media design module as it can generate responses and clarification questions in a conversational system on its own. The capabilities can be extended by using more sophisticated NLG techniques (see Gatt and Kramer [23] for a recent overview).

In [19, 56] a deep reinforcement learning system for optimising a visually grounded goal-directed dialogue system was implemented using TensorFlow. GuessWhat?![31], the corresponding open source contribution, is moderately annotated and contains a ReadMe file that allows verification of the published results by outlining required steps for their reproduction. Pre-trained models are available for download and basic installation instructions are contained as well. Unfortunately, the authors omitted correct Python dependency version information which leads to some trial-and-error during installation. This contribution uses the Apache-2.0 license.[32]

Backend

Visual scene understanding is an important property of an IUI which needs to process image or video input. From a technical perspective, this requires image classification and accurate region proposals so that a meaningful segmentation of a scene with multiple objects is possible. Sonntag et al. contributed the py-faster-rcnn-ft[33] Python library that allows convenient fine-tuning of deep learning models that offer this functionality. E.g., the VGG_CNN_M_1024 model [11] can be fine-tuned on specific categories of the MS COCO [39] image dataset, hence improving classification accuracy [24]. While the library can work entirely in the background, it also comes with an user interface allowing to select categories and inspect the results graphically. The software uses the Caffe framework and is GPL-3 licensed. It features an extensive ReadMe file containing an installation and getting started guide with example listings and optional pre-trained model resources for quick evaluation. This work is based on py-faster-rcnn[34] [47].

Nvidia's DIGITS[35] enables deep learning beginners to design and train models to solve image classification problems and put these models to use. The Python software features an intelligent web (HTML/JS) interface that allows highly interactive modification of the neural network by the user as well as data management and fine-tuning of existing models. The DIGITS interface displays performance statistics in real time so that the user can quickly identify and select the best performing model for deployment. On the other hand, DIGITS can also be used as a backend component of an IUI via its REST API to perform inference on trained models. The software is BSD-3 licensed and can use Caffe, Torch, and Tensorflow as deep learning framework. The installation via docker requires little effort and the accompanying ReadMe file links to multiple well-grounded howto guides. A graphically enriched documentation and introduction to deep learning is available from Nvidia[36].

OUTLOOK: INTERACTIVE MACHINE LEARNING

To develop the positive aspects of artificial intelligence, manage its risks and challenges, and ensure that everyone has the opportunity to help in building an AI-enhanced society and to participate in its benefits, we suggest using a methodology where human intelligences & machine learning take the center stage: *Interactive Machine Learning is the design and implementation of algorithms and intelligent user interface frameworks that facilitate machine learning with the help of human interaction.*

[24] https://github.com/gberta/Visual-Spatial-Network
[25] https://github.com/pupil-labs/pupil
[26] https://github.com/RasaHQ/rasa_core
[27] https://core.rasa.ai/
[28] https://gitter.im/RasaHQ/rasa_core
[29] https://github.com/RasaHQ/rasa_core/tree/master/examples
[30] https://core.rasa.ai/tutorial_basics.html
[31] https://github.com/GuessWhatGame/guesswhat/
[32] Update in future revisions: https://github.com/voicy-ai/DialogStateTracking
[33] https://github.com/DFKI-Interactive-Machine-Learning/py-faster-rcnn-ft
[34] https://github.com/rbgirshick/py-faster-rcnn
[35] https://github.com/NVIDIA/DIGITS
[36] https://devblogs.nvidia.com/parallelforall/digits-deep-learning-gpu-training-system/

This field of research explores the possibilities of helping AI systems to achieve their full potential, based on interaction with humans. The current misconception is that artificial intelligence is more likely to be performance oriented than learning oriented. By machine teaching [51] we can "assist" AI systems in becoming self-sustaining, "lifelong" learners [38, 54] as a domain expert trains complex models by encapsulating the required mechanics of machine learning. This resource oriented methodology contributes in closing the gap between the demand for machine learning models and relatively low amount of experts capable of creating them.

Interactive machine learning (IML) includes feedback in real-time, allowing fast-paced iterative model improvements through intelligent interactions with a user [3]. As shown by Sonntag, when using artificial intelligence (AI) to implement intelligent automated capabilities in an IUI, effects on HCI must be considered in order to prevent negative side-effects like diminished predictability and lost controllability which ultimately impact the usability of the IUI [53]. This adverse effect can be diminished by adopting the concept of the binocular view when building IUIs: both AI and HCI aspects are simultaneously addressed so that the systems intelligence, and how the user should be able to interact with it, are optimally synchronised [29].

Many principles used in active learning are adopted, for example, query strategies for selecting most influential samples that shall be labelled [22, 49] and semi-supervised learning for the automatic propagation of labels under confidence constraints [49]. IML benefits from IUIs that support the user in training/teaching a model and is, at the same time, essential for building and maintaining models for intelligent and multimodal interaction.

Recent works use deep learning in combination with active or passive user input to improve the model training or performance: Green et al. [26] applied the IML concept for a language translation task that benefits from human users and machine agents: the human in the loop can produce higher quality translations while the suggested machine translation is continuously improved as the human corrects its suggestions. Venkitasubramanian et al. [43] present a model that learns to recognise animals by watching documentaries. They implicitly involve humans by incorporating their gaze signal together with the subtitles of a movie as weak supervision signal. Their classifiers learns using image representations from pre-trained convolutional neural networks (CNN). Jiang et al. [32] present an algorithm for learning new object classes and corresponding relations in a human-robot dialogue using CNN-based features. The relations are used to reason about future scenarios where known faces and objects are recognised. Cognolato et al. [15] use human gaze and hand movements to sample images of objects that get manipulated and fine-tune a CNN with that data. In the context of active learning, Käding et al. [33] investigate the trade-off between model quality and the computational effort of fine-tuning for continuously changing models. Jiang et al. [31] present a GPU-accelerated framework for interactive machine learning that allows easy model adoption and provides several result visualisations to support users.

In [31] techniques for customised and interactive model optimisation are proposed. Jiang and Canny use the BIDMach framework [10] which builds upon Caffe to provide a machine learning architecture which is modular and supports primary and secondary loss functions. The users of this system are able to directly manipulate deep learning model parameters during training. The user can perform model optimisations with the help of interactive visualisation tools and controls via a web interface. It however remains challenging to transfer the concepts of interactive machine learning to deep learning algorithms and to investigate their impact on model performance and usability, particularly in the context of IUIs and multimodal settings.

REFERENCES

1. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. (2015). `https://www.tensorflow.org/`

2. Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, Alexander Belopolsky, Yoshua Bengio, Arnaud Bergeron, James Bergstra, Valentin Bisson, Josh Bleecher Snyder, Nicolas Bouchard, Nicolas Boulanger-Lewandowski, Xavier Bouthillier, Alexandre de Brébisson, Olivier Breuleux, Pierre-Luc Carrier, Kyunghyun Cho, Jan Chorowski, Paul Christiano, Tim Cooijmans, Marc-Alexandre Côté, Myriam Côté, Aaron Courville, Yann N. Dauphin, Olivier Delalleau, Julien Demouth, Guillaume Desjardins, Sander Dieleman, Laurent Dinh, Mélanie Ducoffe, Vincent Dumoulin, Samira Ebrahimi Kahou, Dumitru Erhan, Ziye Fan, Orhan Firat, Mathieu Germain, Xavier Glorot, Ian Goodfellow, Matt Graham, Caglar Gulcehre, Philippe Hamel, Iban Harlouchet, Jean-Philippe Heng, Balázs Hidasi, Sina Honari, Arjun Jain, Sébastien Jean, Kai Jia, Mikhail Korobov, Vivek Kulkarni, Alex Lamb, Pascal Lamblin, Eric Larsen, César Laurent, Sean Lee, Simon Lefrancois, Simon Lemieux, Nicholas Léonard, Zhouhan Lin, Jesse A. Livezey, Cory Lorenz, Jeremiah Lowin, Qianli Ma, Pierre-Antoine Manzagol, Olivier Mastropietro, Robert T. McGibbon, Roland Memisevic, Bart van Merriënboer, Vincent Michalski, Mehdi Mirza, Alberto Orlandi, Christopher Pal, Razvan Pascanu, Mohammad Pezeshki, Colin Raffel,

Daniel Renshaw, Matthew Rocklin, Adriana Romero, Markus Roth, Peter Sadowski, John Salvatier, François Savard, Jan Schlüter, John Schulman, Gabriel Schwartz, Iulian Vlad Serban, Dmitriy Serdyuk, Samira Shabanian, Étienne Simon, Sigurd Spieckermann, S. Ramana Subramanyam, Jakub Sygnowski, Jérémie Tanguay, Gijs van Tulder, Joseph Turian, Sebastian Urban, Pascal Vincent, Francesco Visin, Harm de Vries, David Warde-Farley, Dustin J. Webb, Matthew Willson, Kelvin Xu, Lijun Xue, Li Yao, Saizheng Zhang, and Ying Zhang. 2016. Theano: A Python framework for fast computation of mathematical expressions. (2016), 1–19. http://arxiv.org/abs/1605.02688

3. Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine* 35, 4 (dec 2014), 105. DOI: http://dx.doi.org/10.1609/aimag.v35i4.2513

4. Soheil Bahrampour, Naveen Ramakrishnan, Lukas Schott, and Mohak Shah. 2016. Comparative Study of Caffe, Neon, Theano, and Torch for Deep Learning. (2016), 1–11.

5. Michael Barz and Daniel Sonntag. 2016. Gaze-guided object classification using deep neural networks for attention-based computing. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct - UbiComp '16*. ACM Press, New York, New York, USA, 253–256. DOI: http://dx.doi.org/10.1145/2968219.2971389

6. Gedas Bertasius, Hyun Soo Park, Stella X. Yu, and Jianbo Shi. 2017a. First Person Action-Object Detection with EgoNet. In *Proceedings of Robotics: Science and Systems*. http://arxiv.org/abs/1603.04908

7. Gedas Bertasius, Hyun Soo Park, Stella X. Yu, and Jianbo Shi. 2017b. Unsupervised Learning of Important Objects from First-Person Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1956–1964. DOI: http://dx.doi.org/10.1109/ICCV.2017.216

8. Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open Source Language Understanding and Dialogue Management. (dec 2017). http://arxiv.org/abs/1712.05181

9. Daniel Braun and Manfred Langen. 2017. Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. August (2017), 174–185.

10. John Canny and Huasha Zhao. 2013. Big Data Analytics with Small Footprint : Squaring the Cloud. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM* (2013), 95–103. DOI: http://dx.doi.org/10.1145/2487575.2487677

11. Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. *British Machine Vision Conference* (2014). DOI: http://dx.doi.org/10.5244/C.28.6

12. Danqi Chen and Christopher Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* i (2014), 740–750. DOI: http://dx.doi.org/10.3115/v1/D14-1082

13. Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. (2015), 1–6. DOI: http://dx.doi.org/10.1145/2532637

14. François Chollet and Others. 2015. Keras. (2015). https://github.com/fchollet/keras

15. Matteo Cognolato, Mara Graziani, Francesca Giordaniello, Gianluca Saetta, Franco Bassetto, Peter Brugger, Barbara Caputo, Henning Müller, and Manfredo Atzori. 2017. Semi-automatic training of an object recognition system in scene camera data using gaze tracking and accelerometers. In *International Conference on Computer Vision Systems (ICVS)*.

16. Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. 2011. Torch7: A matlab-like environment for machine learning. *BigLearn, NIPS Workshop* (2011), 1–6. http://infoscience.epfl.ch/record/192376/files/Collobert

17. Committee on Technology. 2016. *Preparing for the Future of Artificial Intelligence*. Technical Report. Executive Office of the President of the United States, National Science and Technology Council, Committee on Technology. https://obamawhitehouse.archives.gov/sites/default/files/whitehouse

18. Aymeric Damien and Others. 2016. TFLearn. (2016). https://github.com/tflearn/tflearn

19. Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. GuessWhat?! Visual object discovery through multi-modal dialogue. *Conference on Computer Vision and Pattern Recognition* (2017). DOI: http://dx.doi.org/10.1109/CVPR.2017.475

20. Deeplearning4j Development Team. 2018. Deeplearning4j: Open-source distributed deep learning for the JVM. (2018). http://deeplearning4j.org

21. Bradley J. Erickson, Panagiotis Korfiatis, Zeynettin Akkus, Timothy Kline, and Kenneth Philbrick. 2017. Toolkits and Libraries for Deep Learning. *Journal of Digital Imaging* 30, 4 (2017), 400–405. DOI: http://dx.doi.org/10.1007/s10278-017-9965-6

22. James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. 2008. CueFlik: Interactive Concept Learning in Image Search. In *Proceeding of the twenty-sixth annual*

CHI conference on Human factors in computing systems - CHI '08. ACM Press, New York, New York, USA, 29. DOI:http://dx.doi.org/10.1145/1357054.1357061

23. Albert Gatt and Emiel Krahmer. 2018. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61 (mar 2018), 65–170. http://dx.doi.org/10.1613/jair.5477http://arxiv.org/abs/1703.09902

24. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), 580–587. DOI: http://dx.doi.org/10.1109/CVPR.2014.81

25. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

26. Spence Green, Jeffrey Heer, and Christopher D. Manning. 2015. Natural Language Translation at the Intersection of AI and HCI. *Queue* 13, 6 (2015), 1–13. DOI: http://dx.doi.org/10.1145/2791301.2798086

27. Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. 2016. Cooperative Inverse Reinforcement Learning. In *Advances in Neural Information Processing Systems 29*, D D Lee, M Sugiyama, U V Luxburg, I Guyon, and R Garnett (Eds.). Curran Associates, Inc., 3909–3917. http://papers.nips.cc/paper/6420-cooperative-inverse-reinforcement-learning.pdf

28. Johann Hauswald, Yiping Kang, Michael A. Laurenzano, Quan Chen, Cheng Li, Trevor Mudge, Ronald G. Dreslinski, Jason Mars, and Lingjia Tang. 2015. DjiNN and Tonic: DNN as a service and its implications for future warehouse scale computers. In *Proceedings of the 42nd Annual International Symposium on Computer Architecture - ISCA '15*. ACM, 27–40. DOI: http://dx.doi.org/10.1145/2749469.2749472

29. Anthony Jameson, Aaron Spaulding, and Neil Yorke-Smith. 2009. Introduction to the Special Issue on "Usable AI". *AI Magazine* 30, 4 (2009), 11–15.

30. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. (2014). DOI:http://dx.doi.org/10.1145/2647868.2654889

31. Biye Jiang and John Canny. 2017. Interactive Machine Learning via a GPU-accelerated Toolkit. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces - IUI '17*. ACM Press, New York, New York, USA, 535–546. DOI: http://dx.doi.org/10.1145/3025171.3025172

32. Shuqiang Jiang, Weiqing Min, Xue Li, Huayang Wang, Jian Sun, and Jiaqi Zhou. 2017. Dual Track Multimodal Automatic Learning through Human-Robot Interaction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 4485–4491. DOI: http://dx.doi.org/10.24963/ijcai.2017/626

33. Christoph Käding, Erik Rodner, Alexander Freytag, and Joachim Denzler. 2017. Fine-Tuning Deep Neural Networks in Continuous Learning Scenarios. In *Computer Vision – ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III*, Chu-Song Chen, Jiwen Lu, and Kai-Kuang Ma (Eds.). Springer International Publishing, Cham, 588–605. DOI: http://dx.doi.org/10.1007/978-3-319-54526-4_43

34. Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. In *Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 1151–1160. DOI: http://dx.doi.org/10.1145/2638728.2641695

35. Davis. E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10 (2009), 1755–1758. DOI: http://dx.doi.org/10.1145/1577069.1755843

36. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems* (2012), 1–9. DOI: http://dx.doi.org/10.1016/j.protcy.2014.09.007

37. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

38. Henry Lieberman, Bonnie A Nardi, and David J Wright. 2001. Chapter 12 - Training Agents to Recognize Text by Example. In *Your Wish is My Command*, Henry Lieberman (Ed.). Morgan Kaufmann, San Francisco, 227 – XII. DOI:http://dx.doi.org/https://doi.org/10.1016/B978-155860688-3/50013-0

39. Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. *European conference on computer vision* (2014), 740–755. DOI: http://dx.doi.org/10.1007/978-3-319-10602-1_48

40. Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural Relation Extraction with Selective Attention over Instances. *Proceedings of ACL* (2016), 2124–2133. DOI: http://dx.doi.org/10.18653/v1/P16-1200

41. Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal Deep Learning. In *Proceedings of the 28th International*

Conference on International Conference on Machine Learning (ICML'11). Omnipress, USA, 689–696. http://dl.acm.org/citation.cfm?id=3104482.3104569

42. Thien Huu Nguyen and Ralph Grishman. 2015. Combining Neural Networks and Log-linear Models to Improve Relation Extraction. *arXiv preprint arXiv:1511.05926* (2015). http://arxiv.org/abs/1511.05926

43. Aparna Nurani Venkitasubramanian, Tinne Tuytelaars, and Marie-Francine Moens. 2017. Learning to recognize animals by watching documentaries: using subtitles as weak supervision. In *Proceedings of the 6th Workshop on Vision and Language (VL'17) at EACL 2016*.

44. Sharon Oviatt, Björn Schuller, Philip R Cohen, Daniel Sonntag, Gerasimos Potamianos, and Antonio Krüger. 2017. The Handbook of Multimodal-Multisensor Interfaces. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA, Chapter Intro, 1–15. DOI:http://dx.doi.org/10.1145/3015783.3015785

45. Aniruddha Parvat, Jai Chavan, Siddhesh Kadam, Souradeep Dev, and Vidhi Pathak. 2017. A Survey of Deep-learning Frameworks. *International Conference on Inventive Systems and Control (ICISC)* (2017), 7. DOI: http://dx.doi.org/10.1109/ICISC.2017.8068684

46. Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society.

47. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems (NIPS)*.

48. Burr Settles. 2010. Active learning literature survey. *University of Wisconsin, Madison* 52, 55-66 (2010), 11.

49. Burr Settles. 2011. Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1467–1478. http://dl.acm.org/citation.cfm?id=2145588

50. Shaohuai Shi, Qiang Wang, Pengfei Xu, and Xiaowen Chu. 2016. Benchmarking State-of-the-Art Deep Learning Software Tools. (2016). DOI: http://dx.doi.org/10.1109/CCBD.2016.029

51. Patrice Simard, Saleema Amershi, Max Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Chris Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. 2017. Machine Teaching: A New Paradigm for Building Machine Learning Systems. (2017). https://arxiv.org/abs/1707.06742

52. Daniel Sonntag. 2009. Introspection and Adaptable Model Integration for Dialogue-based Question Answering. In *IJCAI*. 1549–1554.

53. Daniel Sonntag. 2012. Collaborative Multimodality. *KI - Künstliche Intelligenz* 26, May 2012 (2012), 161–168. DOI:http://dx.doi.org/10.1007/s13218-012-0169-4

54. Daniel Sonntag. 2017. Intelligent User Interfaces - A Tutorial. *CoRR* abs/1702.0 (2017). http://arxiv.org/abs/1702.05250

55. Daniel Sonntag, Ralf Engel, Gerd Herzog, Alexander Pfalzgraf, Norbert Pfleger, Massimo Romanelli, and Norbert Reithinger. 2007. SmartWeb Handheld - Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services. In *Artifical Intelligence for Human Computing, ICMI 2006 and IJCAI 2007 International Workshops, Banff, Canada, November 3, 2006, Hyderabad, India, January 6, 2007, Revised Seleced and Invited Papers (Lecture Notes in Computer Science)*, Thomas S. Huang, Anton Nijholt, Maja Pantic, and Alex Pentland (Eds.), Vol. 4451. Springer, 272–295. DOI: http://dx.doi.org/10.1007/978-3-540-72348-6_14

56. Florian Strub, Harm de Vries, Jérémie Mary, Bilal Piot, Aaron Courville, and Olivier Pietquin. 2017. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2765–2771. DOI: http://dx.doi.org/10.24963/ijcai.2017/385

57. Yaniv Taigman, Marc Aurelio Ranzato, Tel Aviv, and Menlo Park. 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *Cvpr* (2014). DOI:http://dx.doi.org/10.1109/CVPR.2014.220

58. Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a Next-Generation Open Source Framework for Deep Learning. *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)* (2015), 1–6.

59. Wolfgang Wahlster and Mark Maybury. 1998. Intelligent User Interfaces: An Introduction. *RUIU* (1998), 1–13.

60. Luke Yeager. 2015. DIGITS : the Deep learning GPU Training System. *ICML AutoML Workshop* (2015).

61. Dong Yu, Adam Eversole, Mike Seltzer, Kaisheng Yao, Zhiheng Huang, Brian Guenter, Oleksii Kuchaiev, Yu Zhang, Frank Seide, Huaming Wang, Jasha Droppo, Geoffrey Zweig, Chris Rossbach, Jon Currey, Jie Gao, Avner May, Baolin Peng, Andreas Stolcke, and Malcolm Slaney. 2015. An Introduction to Computational Networks and the Computational Network Toolkit. *Microsoft Technical Report* 112, MSR-TR-2014-112 (2015).

62. Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant Supervision for Relation Extraction via

Piecewise Convolutional Neural Networks. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* September (2015), 1753–1762. DOI:http://dx.doi.org/10.18653/v1/D15-1203

63. Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. *Coling* 2011 (2014), 2335–2344. http://www.nlpr.ia.ac.cn/cip/liukang.files/camera

64. Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2016), 207–212. http://anthology.aclweb.org/P16-2034