Industrial Artificial Intelligence Technologies and Applications

RIVER PUBLISHERS SERIES IN COMMUNICATIONS AND NETWORKING

Series Editors

ABBAS JAMALIPOUR
The University of Sydney
Australia

MARINA RUGGIERI
University of Rome Tor Vergata
Italy

The "River Publishers Series in Communications and Networking" is a series of comprehensive academic and professional books which focus on communication and network systems. Topics range from the theory and use of systems involving all terminals, computers, and information processors to wired and wireless networks and network layouts, protocols, architectures, and implementations. Also covered are developments stemming from new market demands in systems, products, and technologies such as personal communications services, multimedia systems, enterprise networks, and optical communications.

The series includes research monographs, edited volumes, handbooks and textbooks, providing professionals, researchers, educators, and advanced students in the field with an invaluable insight into the latest research and developments.

Topics included in this series include:

- Communication theory
- Multimedia systems
- Network architecture
- Optical communications
- Personal communication services
- Telecoms networks
- Wi-Fi network protocols

For a list of other books in this series, visit www.riverpublishers.com

Industrial Artificial Intelligence Technologies and Applications

Editors

Ovidiu Vermesan

SINTEF, Norway

Franz Wotawa

TU Graz, Austria

Mario Diaz Nava

STMicroelectronics, France

Björn Debaillie

imec, Belgium

River Publishers

Published, sold and distributed by:
River Publishers
Alsbjergvej 10
9260 Gistrup
Denmark

www.riverpublishers.com

ISBN: 978-87-7022-791-9 (Hardback)
978-87-7022-790-2 (Ebook)

©The Editor(s) (if applicable) and The Author(s) 2022. This book is published open access.

Open Access
This book is distributed under the terms of the Creative Commons Attribution-Non-Commercial 4.0 International License, CC-BY-NC 4.0) (http://creativecommons.org/licenses/by/4.0/), which permits use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated. The images or other third party material in this book are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt, or reproduce the material.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper.

Dedication

"Without change there is no innovation, creativity, or incentive for improvement. Those who initiate change will have a better opportunity to manage the change that is inevitable."

- William Pollard

"The brain is like a muscle. When it is in use we feel very good. Understanding is joyous."

- Carl Sagan

"By far, the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."

- Eliezer Yudkowsky

Acknowledgement

The editors would like to thank all the contributors for their support in the planning and preparation of this book. The recommendations and opinions expressed in the book are those of the editors, authors, and contributors and do not necessarily represent those of any organizations, employers, or companies.

Ovidiu Vermesan
Franz Wotawa
Mario Diaz Nava
Björn Debaillie

Contents

Preface **xv**

List of Figures **xix**

List of Tables **xxv**

List of Contributors **xxix**

1 Benchmarking Neuromorphic Computing for Inference **1**
Simon Narduzzi, Loreto Mateu, Petar Jokic,
Erfan Azarkhish, and Andrea Dunbar
1.1 Introduction 2
1.2 State of the art in Benchmarking 3
1.2.1 Machine Learning 5
1.2.2 Hardware 7
1.3 Guidelines 9
1.3.1 Fair and Unfair Benchmarking 10
1.3.2 Combined KPIs and Approaches for Benchmarking 11
1.3.3 Outlook : Use-case Based Benchmarking 13
1.4 Conclusion 15
References 16

2 Benchmarking the Epiphany Processor as a Reference Neuromorphic Architecture **21**
Maarten Molendijk, Kanishkan Vadivel, Federico Corradi,
Gert-Jan van Schaik, Amirreza Yousefzadeh,
and Henk Corporaal
2.1 Introduction and Background 21

2.2 Comparison with a Few Well-Known Digital Neuromorphic Platforms . . . 24
2.3 Major Challenges in Neuromorphic Architectures . . . 26
2.3.1 Memory Allocation . . . 26
2.3.2 Efficient Communication . . . 28
2.3.3 Mapping SNN onto Hardware . . . 29
2.3.4 On-chip Learning . . . 29
2.3.5 Idle Power Consumption . . . 30
2.4 Measurements from Epiphany . . . 30
2.5 Conclusion . . . 32
References . . . 33

3 Temporal Delta Layer: Exploiting Temporal Sparsity in Deep Neural Networks for Time-Series Data . . . 35
Preetha Vijayan, Amirreza Yousefzadeh, Manolis Sifalakis, and Rene van Leuken
3.1 Introduction . . . 36
3.2 Related Works . . . 37
3.3 Methodology . . . 39
3.3.1 Delta Inference . . . 39
3.3.2 Sparsity Induction Using Activation Quantization . . . 40
3.3.2.1 Fixed Point Quantization . . . 41
3.3.2.2 Learned Step-Size Quantization . . . 42
3.3.3 Sparsity Penalty . . . 44
3.4 Experiments and Results . . . 45
3.4.1 Baseline . . . 45
3.4.2 Experiments . . . 45
3.4.3 Result Analysis . . . 46
3.5 Conclusion . . . 49
References . . . 50

4 An End-to-End AI-based Automated Process for Semiconductor Device Parameter Extraction . . . 53
Dinu Purice, Matthias Ludwig, and Claus Lenz
4.1 Introduction . . . 54
4.2 Semantic Segmentation . . . 56

4.2.1 Proof of Concept and Architecture Overview 56
4.2.2 Implementation Details and Result Overview 61
4.3 Parameter Extraction 64
4.4 Conclusion 68
4.5 Future Work 69
References 69

5 AI Machine Vision System for Wafer Defect Detection 73
Dmitry Morits, Marcelo Rizzo Piton, and Timo Laakko
5.1 Introduction and Background 73
5.2 Machine Vision-based System Description 75
5.3 Conclusion 78
References 79

6 Failure Detection in Silicon Package 81
Saad Al-Baddai and Jan Papadoudis
6.1 Introduction and Background 82
6.2 Dataset Description 83
6.2.1 Data Collection & Labelling 84
6.3 Development and Deployment 85
6.4 Transfer Learning and Scalability 86
6.5 Result and Discussion 87
6.6 Conclusion and Outlooks 89
References 89

7 S2ORC-SemiCause: Annotating and Analysing Causality in the Semiconductor Domain 91
Xing Lan Liu, Eileen Salhofer, Anna Safont Andreu, and Roman Kern
7.1 Introduction 92
7.2 Dataset Creation 93
7.2.1 Corpus 93
7.2.2 Annotation Guideline 93
7.2.3 Annotation Methodology 94
7.2.4 Dataset Statistics 95
7.2.5 Causal Cue Phrases 95
7.3 Baseline Performance 96
7.3.1 Train-Test Split 96

7.3.2 Causal Argument Extraction 97
7.3.3 Error Analysis 97
7.4 Conclusions 99
References 99

8 Feasibility of Wafer Exchange for European Edge AI Pilot Lines 103
Annika Franziska Wandesleben, Delphine Truffier-Boutry, Varvara Brackmann, Benjamin Lilienthal-Uhlig, Manoj Jaysnkar, Stephan Beckx, Ivan Madarevic, Audde Demarest, Bernd Hintze, Franck Hochschulz, Yannick Le Tiec, Alessio Spessot, and Fabrice Nemouchi
8.1 Introduction 104
8.2 Technical Details and Comparison 105
8.2.1 Comparison TXRF and VPD-ICPMS Equipment for Surface Analysis 105
8.2.2 VPD-ICPMS Analyses on Bevel 108
8.3 Cross-Contamination Check-Investigation 109
8.3.1 Example for the Comparison of the Institutes 109
8.4 Conclusiion 111
References 112

9 A Framework for Integrating Automated Diagnosis into Simulation 113
David Kaufmann and Franz Wotawa
9.1 Introduction 113
9.2 Model-based Diagnosis 115
9.3 Simulation and Diagnosis Framework 118
9.3.1 FMU Simulation Tool 118
9.3.2 ASP Diagnose Tool 120
9.4 Experiment 121
9.5 Conclusion 125
References 127

10 Deploying a Convolutional Neural Network on Edge MCU and Neuromorphic Hardware Platforms 129
Simon Narduzzi, Dorvan Favre, Nuria Pazos Escudero and L. Andrea Dunbar

10.1 Introduction 129
10.2 Related Work 130
10.3 Methods 131
10.3.1 Neural Network Deployment 131
10.3.1.1 Task and Model 132
10.3.1.2 Experimental Setup 132
10.3.1.3 Deployment 133
10.3.2 Measuring the Ease of Deployment 135
10.4 Results 136
10.4.1 Inference Results 136
10.4.2 Perceived Effort 137
10.5 Conclusion 137
References 138

11 Efficient Edge Deployment Demonstrated on YOLOv5 and Coral Edge TPU 141
Ruben Prokscha, Mathias Schneider, and Alfred Höß
11.1 Introduction 141
11.2 Related Work 142
11.3 Experimental Setup 143
11.3.1 Google Coral Edge TPU 143
11.3.2 YOLOv5 145
11.4 Performance Considerations 145
11.4.1 Graph Optimization 145
11.4.1.1 Incompatible Operations 145
11.4.1.2 Tensor Transformations 146
11.4.2 Performance Evaluation 147
11.4.2.1 Speed-Accuracy Comparison 147
11.4.2.2 USB Speed Comparison 150
11.4.3 Deployment Pipeline 151
11.5 Conclusion and Future Work 152
References 152

12 Embedded Edge Intelligent Processing for End-To-End Predictive Maintenance in Industrial Applications 157
Ovidiu Vermesan and Marcello Coppola
12.1 Introduction and Background 158

12.2 Machine and Deep Learning for Embedded Edge Predictive Maintenance 159
12.3 Approaches for Predictive Maintenance 161
12.3.1 Hardware and Software Platforms 162
12.3.2 Motor Classification Use Case 163
12.4 Experimental Setup 163
12.4.1 Signal Data Acquisition and Pre-processing 164
12.4.2 Feature Extraction, ML/DL Model Selection and Training 165
12.4.3 Optimisation and Tuning Performance 167
12.4.4 Testing 169
12.4.5 Deployment 170
12.4.6 Inference 172
12.5 Discussion and Future Work 173
References 174

13 AI-Driven Strategies to Implement a Grapevine Downy Mildew Warning System 177
Luiz Angelo Steffenel, Axel Langlet, Lilian Hollard, Lucas Mohimont, Nathalie Gaveau, Marcello Copola, Clément Pierlot, and Marine Rondeau
13.1 Introduction 177
13.2 Research Material and Methodology 179
13.2.1 Datasets 179
13.2.2 Labelling Methodology 180
13.3 Machine Learning Models 180
13.4 Results 183
13.4.1 Primary Mildew Infection Alerts 183
13.4.2 Secondary Mildew Infection Alerts 184
13.5 Discussion 185
13.6 Conclusion 186
References 187

14 On the Verification of Diagnosis Models 189
Franz Wotawa and Oliver Tazl
14.1 Introduction 189
14.2 The Model Testing Challenge 192

14.3 Use Case 194
14.4 Open Issues and Challenges 198
14.5 Conclusion 201
References 201

Index **205**

About the Editors **207**

Preface

Industrial Artificial Intelligence Technologies and Applications

Digitalisation and Industry 5.0 are changing how manufacturing facilities operate by deploying many sensors/actuators, edge computing, and IIoT devices and forming intelligent networks of collaborative machines that are able to collect, aggregate, and intelligently process data at a network's edge.

Given the vast amount of data produced by IIoT devices, computing at the edge is required. In this context, edge computing plays an important role – the edge should provide computing resources for edge intelligence with dependability, data management, and aggregation provision in mind. Edge intelligence – for example, AI technologies with edge computing for training/learning, testing, or inference – is essential for IIoT applications to build models that can learn from a large amount of aggregated data.

Edge computing is a distributed computing paradigm that brings computation and data storage closer to a device's location. AI algorithms process the data created on a device with or without an internet connection. These new AI-based algorithms allow data to be processed within a few milliseconds, providing real-time feedback.

The AI models operate on the devices themselves without the need for a cloud connection and without the problems associated with data latency, which results in much faster data processing and support for use cases that require real-time inferencing.

Major challenges remain in achieving this potential due to the inherent complexity of designing and deploying energy-efficient edge AI algorithms and architectures, the intricacy of complex variations in neural network architectures, and the underlying limited processing capabilities of edge AI accelerators.

Industrial-edge AI can run on various hardware platforms, from ordinary microcontrollers (MCUs) to advanced neural processing devices. IIoT edge AI-connected devices use embedded algorithms to monitor device behaviour and collect and process device data. Devices make decisions, automatically correct problems, and predict future performance.

AI-based technologies are used across industries by introducing intelligent techniques, including machine and deep learning, cognitive computing, and computer vision. The application of the techniques and methods of AI in the industrial sector is a crucial reference source that provides vital research on implementing advanced technological techniques in this sector.

This book offers comprehensive coverage of the topics presented at the "International Workshop on Edge Artificial Intelligence for Industrial Applications (EAI4IA)" in Vienna, 25-26 July 2022. EAI4IA is co-located with the 31st International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence (IJCAI-ECAI 2022). It combines the ideas and concepts developed by researchers and practitioners working on providing edge AI methods, techniques, and tools for use in industrial applications.

By highlighting important topics, such as embedded AI for semiconductor manufacturing and trustworthy, dependable, and explainable AI for the digitising industry, verification, validation and benchmarking of AI systems and technologies, AI model development workflows and hardware target platforms deployment, the book explores the challenges faced by AI technologies deployed in various industrial application domains.

The book is ideally structured and designed for researchers, developers, managers, academics, analysts, post-graduate students, and practitioners seeking current research on the involvement of industrial-edge AI. It combines the latest methodologies, tools, and techniques related to AI and IIoT in a joint volume to build insight into their sustainable deployment in various industrial sectors.

The book is structured around four different topics:

1. **Verification, Validation and Benchmarking of AI Systems and Technologies.**
2. **Trustworthy, Dependable AI for Digitising Industry.**
3. **Embedded AI for semiconductor manufacturing.**
4. **AI model development workflow and HW target platforms deployment.**

In the following, the papers published in this book are briefly discussed.

S. Narduzzi, L. Mateu, P. Jokic, E. Azarkhish, and A. Dunbar: "Benchmarking Neuromorphic Computing for Inference" tackle the challenge of benchmarking aiming at providing a fair and user-friendly method. The authors introduce the challenge and finally come up with possible key performance indicators.

M. Molendijk, K. Vadivel, F. Corradi, G-J. van Schaik, A. Yousefzadeh, and H. Corporaal: "Benchmarking the Epiphany Processor as a Reference Neuromorphic Architecture" compare different implementations of neuromorphic processors and present suggestions for improvements.

P. Vijayan, A. Yousefzadeh, M. Sifalakis, and R. van Leuken: "Temporal Delta Layer: Exploiting Temporal Sparsity in Deep Neural Networks for Time-Series Data" deal with improving the learning of time-series data in the context of deep neural networks. In particular, the authors consider sparsity and show experimentally overall improvements.

D. Purice, M. Ludwig, and C. Lenz: "An End-to-End AI-based Automated Process for Semiconductor Device Parameter Extraction" present a validation pipeline aiming at gaining trust in semiconductor devices relying on authenticity checking. The authors further evaluate their approach by considering several artificial neural network architectures.

D. Morits, M. Rizzo Piton, and T. Laakko: "AI machine vision system for wafer defect detection" discuss the use of machine learning for fault detection based on images in the context of semiconductor manufacturing.

S. Al-Baddai and J. Papadoudis: "Failure detection in silicon package" discuss the use of machine learning techniques for wire-bonding inspection occurring during the packaging of semiconductors. The authors report on the accuracy of failure detection using machine learning in the complex industrial environment.

X. L. Liu, Eileen Salhofer, A. Safont Andreu, and R. Kern: "S2ORC-SemiCause: Annotating and analysing causality in the semiconductor domain" introduce a benchmark dataset to be used in the context of cause-effect reasoning for extracting causal relations.

A. Wandesleben, D. Truffier-Boutry, V. Brackmann, B. Lilienthal-Uhlig, M. Jaysnkar, S. Beckx, I. Madarevic, A. Demarest, B. Hintze, F. Hochschulz, Y. Le Tiec, A. Spessot, and F. Nemouchi: "Feasibility of wafer exchange for European Edge AI pilot lines" focus on contamination monitoring for allowing to exchange wafers among different facilities. In particular, the authors presented an analysis of whether such an exchange would be feasible considering three European research institutes.

D. Kaufmann and F. Wotawa: "A framework for integrating automated diagnosis into simulation" discuss a framework that allows the integration of model-based diagnosis algorithms in physical simulation. The framework can be used for verifying and validating diagnosis implementations for cyber-physical systems.

S. Narduzzi, D. Favre, N. Pazos Escudero, and A. Dunbar: “Deploying a Convolutional Neural Network on Edge MCU and Neuromorphic Hardware Platforms” discuss the deployment of neural networks for edge computing considering different platforms. The authors also report on the perceived effort of deployment for each of the platforms.

R. Prokscha, M. Schneider, and A. Höß: “Efficient Edge Deployment Demonstrated on YOLOv5 and Coral Edge TPU”consider the question of deployment of machine learning on the edge.

O. Vermesan and M. Coppola: ”Embedded Edge Intelligent Processing for End-To-End Predictive Maintenance in Industrial Applications” presented the use of machine learning for edge computing supporting predictive maintenance using different technologies, workflows, and datasets.

L. A. Steffenel, A. Langlet, L. Hollard, L. Mohimont, N. Gaveau, M. Copola, C. Pierlot, and M. Rondeau: ”AI-Driven Strategies to Implement a Grapevine Downy Mildew Warning System” outline the use of machine learning for identifying infections occurring in vineyards and present an experimental evaluation comparing different machine learning algorithms.

F. Wotawa and O. Tazl: ”On the Verification of Diagnosis Models” focus on challenges of verification and in particular testing applied to logic-based diagnosis. The authors consider testing system models and use a running example for demonstrating how such models can be tested and come up with open research questions.

List of Figures

Figure 1.1 Benchmarking fairness. (a) Unfair benchmarking: the KPIs are comparable, but the benchmarked hardware platforms are not exploited to their full potential. (b) Fair benchmarking: the hardware platforms are exploited to their full potential, but the resulting combined KPIs (KPI_{CB}) are not comparable. 12
Figure 1.2 Combined KPIs for fair benchmarking 12
Figure 1.3 Benchmarking pipeline based on use-cases. An automated search finds the best possible model exploiting the performance offered by each target hardware platforms. The resulting combined KPIs are comparable. 14
Figure 2.1 Overall scalable architecture of Epiphany-III. . . . 22
Figure 2.2 Adapteva launched an $99 Epiphany-III based single board computer as their first product. 23
Figure 2.3 Flow chart of processing a LIF neuron with processing time measured in Epiphany. 31
Figure 3.1 (a) Standard DNN, and (b) DNN with proposed temporal delta layer 37
Figure 3.2 Sparsity in activation (Δx) drastically reduce the memory fetches and multiplications between Δx and columns of weight matrix, W, that correspond to zero. 38
Figure 3.3 Demonstration of two temporally consecutive activation maps leading to near zero values (rather than absolute zeroes) after delta operation. 41
Figure 3.4 Importance of step size in quantization: on the right side, in all three cases, the data is quantized to five bins with different uniform step sizes. However, without optimum step size value, the quantization

	can detrimentally alter the range and resolution of the original data.	43
Figure 3.5	Evolution of quantization step size from initialization to convergence in LSQ. As step-size is a learnable parameter, it gets re-adjusted during training to cause minimum information loss in each layer.	48
Figure 4.1	Overview of the architecture.	56
Figure 4.2	Examples showcasing different semiconductor technologies	57
Figure 4.3	Examples of labelled data showcasing the different ROIs: green – VIA; yellow – metal; teal – lateral isolation; red – poly; blue – deep trench isolation	57
Figure 4.4	Histograms of the investigated data grouped by label of interest	58
Figure 4.5	Overview of the U-net architecture	58
Figure 4.6	Overview of the FPN architecture	59
Figure 4.7	Overview of the GSCNN architecture	59
Figure 4.8	Overview of the PSPNet architecture	60
Figure 4.9	Overview of the Siamese network architecture	60
Figure 4.10	Average Dice Scores (blue) and spread (green) per investigated network architecture, along with the final chosen architecture (red)	61
Figure 4.11	An overview of the U-net cascade architecture, consisting of a 2D U-net (top) and a 3D U-net (bottom) which takes as input the high resolution input image stacked with the output segmentation of the first stage	63
Figure 4.12	Utilised cluster evaluation techniques.	65
Figure 4.13	Example cross-section image with annotated metal and contact/VIA features	66
Figure 4.14	Example cross-section image (upper left). The polygonised VIA objects are shown (lower left). A dendrogram is shown for the relative distances of the y-coordinates of the single objects (upper right). Finally, the results of the utilised cluster evaluation techniques are presented (lower right).	67
Figure 5.1	Examples of microscopic images of various superconductor and semiconductor devices with surface defects	74

Figure 5.2 General architecture of the developed machine vision system 76
Figure 5.3 A scheme of the image dataset preparation, including labelling, cropping and data augmentation . . . 76
Figure 5.4 Example of binary classification of wafer defects: defect vs background 78
Figure 6.1 *Left*: Curve with abnormal minimum position (red) in comparison to normal ones (white) of recorded sensor data during wirebonding process. *Right*: shows an example of abnormal OOI image with shown crack on the surface. 84
Figure 6.2 Flow chart of development and deployment life cycle for AI solution at IFX. In development phase data scientists could use different programming language as the final model can be converted to ONNX. In deployment phase, the vision frame can simply access to ONNX and run during inference time. 85
Figure 6.3 Process flow integration of the developed AD solution into an existing IFX infrastructure. 86
Figure 6.4 show the flow processes during silicon package, the backside blue arrow shows the position of transfer learning from OOI backwards to taken images after molding process 87
Figure 6.5 shows an example of the OOI image on left side (This image is taken before shopping and after electrical test) and example of image after molding process on right side. 87
Figure 7.1 Causal cue phrases ranked by frequency for all sentences in S2ORC-SemiCause dataset. 96
Figure 8.1 Comparison of TXRF LLDs of CEA LETI / IMEC 106
Figure 8.2 Comparison of VPD-ICPMS LLDs of CEA LETI / IMEC / FhG 106
Figure 8.3 Schematic of the VPD bevel collection at (a) IMEC, (b) CEA-LETI and (c) FhG IPMS 108
Figure 8.4 Comparison LLDs CEA LETI / IMEC for VPD-ICPMS Bevel 109
Figure 8.5 Comparison TXRF results of CEA LETI / IMEC for IMEC inspection tool 110

Figure 8.6 Comparison VPD-ICPMS results of CEA LETI / IMEC /FhG for IMEC inspection tool 111
Figure 8.7 Comparison VPD-ICPMS bevel results of CEA LETI / IMEC for IMEC inspection tool 111
Figure 9.1 A simple electric circuit comprising bulbs, a switch and a battery. 115
Figure 9.2 Illustration of the simulation and diagnose environment as well as the overall operating principles. The framework of the FMU Simulation Tool provides an interface to enable the integration of a diagnose tool and/or other methods. The models can be substituted by any others in the provided framework. 119
Figure 9.3 Simulation showing the measured signal output of the two bulbs, switch and the battery. For this example a fault injection (`broken`) in bulb 1 after 0.2 seconds (red indicator) and a fault injection (`broken`) to the switch after 0.3 seconds (orange indicator) is initiated. 124
Figure 9.4 Simulation and diagnose output results based on the electrical two-lamps circuit with a broken bulb after 0.2 seconds and a broken switch at 0.3 seconds. The upper tables illustrate the simulation input/output signals, which are used as observation for the diagnose (lower tables) part. Based on the given observations for the three selected time steps, different diagnose results are obtained. 125
Figure 10.1 Illustration of LeNet-5 architecture. 132
Figure 10.2 Deployment pipelines for all platforms. From left to right: STM32L4R9, Kendryte K210 and DynapCNN. For DynapCNN, the pipeline is contained in a single Python script, while the other relay on external languages and tools. 134
Figure 11.1 Raspberry Pi 4 with Google Coral edge TPU USB accelerator. 144
Figure 11.2 Quantized edge TPU Models. 146
Figure 11.3 USB3 speed-accuracy comparison of different model types and configurations for edge TPU deployment. 149

Figure 11.4 YOLOv5s inference speed comparison between USB2 and USB3 150
Figure 11.5 Micro software stack for fast and lightweight edge deployment. 151
Figure 12.1 Industrial motor components 160
Figure 12.2 Micro-edge AI processing flow 162
Figure 12.3 Visualisation of two selected classes signals in both temporal and frequency domain with NEAI 165
Figure 12.4 Benchmarking with NEAI 166
Figure 12.5 Snapshots of Feature Explorer in EI based on the pre-processing block early in the process. 167
Figure 12.6 Confusion Matrix and Data Explorer based on full training set: Correctly Classified (Green) and Misclassified (Red). 168
Figure 12.7 A comparison between int8 quantized and unoptimized versions of the same model, showing the difference in performance and results. 169
Figure 12.8 Evaluation of trained model using NEAI Emulator with live streaming. 170
Figure 12.9 EI model testing with test datasets. 171
Figure 12.10 Live classification streaming with detected state and confidence (with Tera Term) 172
Figure 13.1 Algorithm for primary infection alarms 181
Figure 13.2 Algorithm for secondary infection alarms 182
Figure 14.1 A simple electric circuit comprising bulbs, a switch and a battery. 191
Figure 14.2 The model-based diagnosis principle and information needed for testing. 192
Figure 14.3 A model for diagnosis of the two lamp example from Figure 14.1 comprising the behavior of the components (lines 1-7) and connections (lines 8-10), and the structure of the circuit (lines 11-18). 195
Figure 14.4 Another simple electric circuit comprising bulbs, switches and a battery. This circuit is an extended version of the circuit from Figure 14.1. On the right, we have the structural model of this circuit in Prolog notation. 198

List of Tables

Table 1.1 Relevant KPIs for tasks, models and hardware domains. We also mention some combined KPIs to illustrate the inter-dependency of the domains. 4

Table 1.2 Accuracy (Acc) for different object detection settings on COCO test-dev. 5

Table 1.3 Representation of resource-constrained KPIs. 6

Table 1.4 Typical display of performance comparison of neuromorphic hardware platforms. 9

Table 1.5 Recent display of performance comparison of neuromorphic hardware platforms. 9

Table 2.1 Memory fragmentations in some digital large-scale neuromorphic chips 27

Table 2.2 Mapping LeNet-5 neural network (with binary weights) in different neuromorphic architectures . . . 28

Table 3.1 Spatial stream - comparison of accuracy and activation sparsity obtained through the proposed scenarios against the baseline. In the case of fixed point quantization, the reported results are for a bitwidth of 6 bits. 46

Table 3.2 Temporal stream - comparison of accuracy and activation sparsity obtained through the proposed scenarios against the benchmark. In the case of fixed point quantization, the reported results are for a bitwidth of 7 bits. 46

Table 3.3 Result of decreasing activation bitwidth in fixed point quantization method. For spatial stream, decreasing below 6 bits caused the accuracy to drop considerably. For temporal stream, the same happened below 7 bits. 47

Table 3.4 Final results from two-stream network after average fusing the spatial and temporal stream weights. With 5% accuracy loss, the proposed method almost doubles the activation sparsity available in comparison to the baseline. 48

Table 4.1 Obtained Dice Scores for each showcased network architecture 61

Table 4.2 Averaged Dice Scores for each label of interest 64

Table 4.3 Utilised cluster evaluation techniques. Notation: n: number of objects in data-set; c: centre of data-set; NC: number of clusters; C_i: the i-th cluster; n_i: number of objects in C_i; c_i: centre of C_i; W_k: the within-cluster sum of squared distances from cluster mean; $W*_k$ appropriate null reference; B reference data-sets 65

Table 6.1 Show the confusion matrix and metrics of the CNN model on productive data for BOT and TOP of OOI images. 89

Table 6.2 Show the confusion matrix and metrics of the CNN model on productive data for BOT and TOP of the new process. 89

Table 7.1 Inter-annotator agreement for the first two iterations. *Arg1* (cause) refers to the span of the arguments that lead to *Arg2* (effect) for the respective relation type. . 94

Table 7.2 Comparison of labels generated by both annotators for Iteration 2. Examples and total counts (in number of arguments) for each type also given. 94

Table 7.3 Descriptive statistics of benchmark datasets. Overview of CoNLL-2003 (training split) and BC5CDR (training split) for named entity recognition, as well as causality dataset BioCause (full dataset), and S2ORC-SemiCause (training split). 95

Table 7.4 Descriptive statistics of *S2ORC-SemiCause* dataset. *#-sent*: total number of annotated sentences, *#-sent no relations*: number of sentences without causality, *Argument*: total amount and mean length (token span) of all annotated argument, *Consequence/Purpose*: amount and mean length of cause and effect arguments for the respective relation types. 97

Table 7.5 Baseline performance using BERT with a token classification head. Both the F_1 scores and the standard derivation over 7 different runs are shown. Despite the small sample size, the standard deviation remain low, similar to previous work. 97
Table 7.6 Comparison of predicted and annotated argument spans for the test split. Examples and total counts (in number of arguments) for correct prediction and for each error source are also given. 98
Table 8.1 Contamination monitoring techniques LETI / IMEC / FhG 105
Table 8.2 Overview VPD-ICPMS LLD determination and technical details for LETI / IMEC / FhG 107
Table 9.1 CPS Model component state description for the light bulb, switch and battery. All used states, including fault states of the components are shown. 123
Table 10.1 Relevant technical specifications of the devices (from constructor websites). 134
Table 10.2 Results on MNIST dataset for all platforms. For the DynapCNN, we report the accuracy and latency for the first spike prediction and over the entire simulation. . 136
Table 10.3 Perceived effort for each stage of the inference. 1: small, 5: large. 137
Table 11.1 Comparison of YOLOv5s model before and after optimizations. 147
Table 11.2 Model comparison in regards of input size, file size, operation 148
Table 12.1 Frameworks and inference engines for integrating AI mechanisms within MCUs 161
Table 13.1 Accuracy of 2019 Primary Infection Models 182
Table 13.2 Accuracy of 2020 Primary Infection Models 182
Table 13.3 Accuracy of 2021 Primary Infection Models 182
Table 13.4 Accuracy of 2021 Primary Infection Models 183
Table 14.1 All eight test cases used to verify the 2-bulb example comprising the used observations and the expected diagnoses. The **P/F** column indicates whether the original model passes ($\surd$) or fails ($\times$) the test. 196

Table 14.2 Running 7 model mutations Mi, where we removed line i in the original model of Figure 14.3, using the 8 test cases from Table 14.1. 197

Table 14.3 Test cases for the extended two-bulb example from Figure 14.4 and their test execution results. In gray we indicate tests that check the expected (fault-free) behavior of the circuit. 199

List of Contributors

Al-Baddai, Saad, *Infineon Technologies AG, Germany*

Andreu, Anna Safont, *University of Klagenfurt, Austria, Infineon Technologies Austria*

Azarkhish, Erfan, *CSEM, Switzerland*

Beckx, Stephan, *imec, Belgium*

Brackmann, Varvara, *Fraunhofer IPMS CNT, Germany*

Coppola, Marcello, *STMicroelectronics, France*

Corporaal, Henk, *Technical University of Eindhoven, Netherlands*

Corradi, Federico, *imec, Netherlands*

Demarest, Audde, *Université Grenoble Alpes, CEA-Leti, France*

Dunbar, Andrea, *CSEM, Switzerland*

Escudero, Nuria Pazos, *HE-Arc, Switzerland*

Favre, Dorvan, *CSEM, Switzerland, HE-Arc, Switzerland*

Gaveau, Nathalie, *Université de Reims Champagne Ardenne, France*

Höß, Alfred, *Ostbayerische Technische Hochschule Amberg-Weiden, Germany*

Hintze, Bernd, *FMD, Germany*

Hochschulz, Franck, *Fraunhofer IMS, Germany*

Hollard, Lilian, *Université de Reims Champagne Ardenne, France*

Jaysnkar, Manoj, *imec, Belgium*

Jokic, Petar, *CSEM, Switzerland*

Kaufmann, David, *Graz University of Technology, Austria*

Kern, Roman, *Graz University of Technology, Austria*

Laakko, Timo, *VTT Technical Research Centre of Finland Ltd, Finland*

Langlet, Axel, *Université de Reims Champagne Ardenne, France*

Le Tiec, Yannick, *Université Grenoble Alpes, CEA, LETI, France*

Lenz, Claus, *Cognition Factory GmbH, Germany*

Leuken, Rene van, *TU Delft, Netherlands*

Lilienthal-Uhlig, Benjamin, *Fraunhofer IPMS CNT, Germany*

Liu, Xing Lan, *Know-Center GmbH, Austria*

Ludwig, Matthias, *Infineon Technologies AG, Germany*

Madarevic, Ivan, *imec, Belgium*

Mateu, Loreto, *Fraunhofer IIS, Germany*

Mohimont, Lucas, *Université de Reims Champagne Ardenne, France*

Molendijk, Maarten, *imec, Netherlands, Technical University of Eindhoven, Netherlands*

Morits, Dmitry, *VTT Technical Research Centre of Finland Ltd, Finland*

Narduzzi, Simon, *CSEM, Switzerland*

Nemouchi, Fabrice, *Université Grenoble Alpes, CEA, LETI, France*

Papadoudis, Jan, *Infineon Technologies AG, Germany*

Pierlot, Clément, *Vranken-Pommery Monopole, France*

Piton, Marcelo Rizzo, *VTT Technical Research Centre of Finland Ltd, Finland*

Prokscha, Ruben, *Ostbayerische Technische Hochschule Amberg-Weiden, Germany*

Purice, Dinu, *Cognition Factory GmbH, Germany*

Rondeau, Marine, *Vranken-Pommery Monopole, Reims, France*

Salhofer, Eileen, *Know-Center GmbH, Austria, Graz University of Technology, Austria*

Schneider, Mathias, *Ostbayerische Technische Hochschule Amberg-Weiden, Germany*

Sifalakis, Manolis, *imec, Netherlands*

Spessot, Alessio, *imec, Belgium*

Steffenel, Luiz Angelo, *Université de Reims Champagne Ardenne, France*

Tazl, Oliver, *Graz University of Technology, Austria*

Truffier-Boutry, Delphine, *Université Grenoble Alpes, CEA, LETI, France*

Vadivel, Kanishkan, *Technical University of Eindhoven, Netherlands*

van Schaik, Gert-Jan, *imec, Netherlands*

Vermesan, Ovidiu, *SINTEF AS, Norway*

Vijayan, Preetha, *TU Delft, Netherlands, imec, Netherlands*

Wandesleben, Annika Franziska, *Fraunhofer IPMS CNT, Germany*

Wotawa, Franz, *Graz University of Technology, Austria*

Yousefzadeh, Amirreza, *imec, Netherlands*