

ECSEL-Verbundvorhaben

„KI für neue Elektroniksysteme und Edge-Computing-Technologien - ANDANTE“

Teilvorhaben Technische Universität Dresden:

Speicheroptimiertes Lernen für eingebettete Systeme

Sachbericht zum Verwendungsnachweis

Zuwendungsempfänger:	Technische Universität Dresden
Förderkennzeichen:	16MEE0118S
Laufzeit des Vorhabens:	01.07.2020 - 29.02.2024
Projektleiter:	Prof. Dr.-Ing. habil. Christian Mayr, christian.mayr@tu-dresden.de
Telefon:	+49 351 463-42392
weitere Verfasser:	Bernhard Vogginger Chen Liu Amir Rostami
Adresse:	Technische Universität Dresden, Professur hochparallele VLSI-Systeme und Neuromikroelektronik 01062 Dresden

I. Kurzbericht (max 2 Seiten)

1. Aufgabenstellung und Stand der Wissenschaft und Technik

Das Gesamtziel von ANDANTE war die Entwicklung neuer Methoden und Hardware für künstliche neuronale Netze in eingebetteten Systemen. Ziel war die Erhöhung der Effizienz gegenüber bestehenden Lösungen, um so komplexere und leistungsfähigere KI-Anwendungen in eingebetteten Systemen realisieren zu können.

Das Vorhaben der TU Dresden widmete sich speicheroptimierten Lernverfahren für neuronale Netze. Diese sollten auf dem neuromorphen Multiprozessorsystem SpiNNaker2 realisiert und hinsichtlich ihrer Effizienz evaluiert werden. Ziel war es, durch verschiedene Ansätze, wie z.B. Redundanzen in den Eingangsdaten, struktureller Sparsity, starker Quantisierung der Gewichte, oder verbesserter Algorithmen, den Speicherbedarf und somit auch den Energieverbrauch konstant niedrig zu halten. Aus den Erkenntnissen bei der Implementierung und eines Benchmarkings der Lernverfahren sollten Modifikationen der SpiNNaker2-Architektur abgeleitet und evaluiert werden. Im Anwendungsfall 1.1 „Personenverfolgung und -zählung mit Indoor-Radarsystemen“ sollten die Konzepte zur Effizienzsteigerung bei der Echtzeitverarbeitung von Radarsignalen erprobt und evaluiert werden.

Schon bei Beginn des Projektes stellte Deep Learning den Stand der Technik für eine Vielzahl von Anwendungen dar. Die zugehörigen tiefen neuronalen Netze (DNN) wurden mit großen Datensätzen auf General-Purpose GPUs, oder sogar GPU-Clustern trainiert. Beim Training kam für gewöhnlich das Gradientenabstiegsverfahren mit Fehlerfortpflanzung (Error Backpropagation Algorithmus) zum Einsatz. Zu dem Zeitpunkt gab es schon viele Ansätze, um die Inferenz, also die Ausführung von neuronalen Netzen für Vorhersagen, effizienter auf Hardware auszuführen. Einerseits wurden spezielle Hardwarebeschleuniger entwickelt, andererseits wurden die Parameter von Fließkommazahlen in Ganzzahlen mit wenigen Bits gewandelt (quantisiert), um dadurch bei der Inferenz den Speicherbedarf und die Rechenkosten zu reduzieren. Das „Learning at the Edge“, das Trainieren von neuronalen Netzen in eingebetteten Systemen anstatt auf Servern, war damals noch ein recht neues Feld und gewann langsam an Bedeutung, z.B. um die Privatsphäre von KI zu gewährleisten. Eine große Herausforderung lag im hohen Speicherbedarf der beim Training benötigt wird: Einerseits für die Trainingsdaten und andererseits für die Zwischenwerte und Gradienten beim Lernen mittels Backpropagation, insbesondere beim Trainieren von rekurrenten neuronalen Netzen (RNN), welche zu dem Zeitpunkt Stand der Technik zur Klassifikation und Vorhersage von Zeitreihen waren. Es gab damals schon einige Ansätze um das Trainieren effizienter zu machen, z.B., durch Training von neuronalen Netzen mit dünnbesetzten Gewichtsmatrizen oder der Verwendung von neuro-inspirierten Lernverfahren die keine Backpropagation nutzen. Spikende neuronale Netze (SNN), die ähnlich wie im Gehirn funktionieren, hatten zu dem Zeitpunkt ein sehr großes Potenzial zur energieeffizienten Inferenz auf neuromorpher Hardware, hinkten bei der erreichbaren Genauigkeit den konventionellen DNN hinterher.

2. Ablauf des Vorhabens

Begonnen wurde mit einer Literaturstudie zu Algorithmen und Methoden für speicheroptimiertes Lernen, sowohl für SNN und RNN. Wir haben uns dann entschieden, uns auf das biologisch-inspirierte Lernverfahren E-Prop für rekurrente spikende neuronale Netze (RSNN) zu fokussieren, weil es ein hohes Potenzial zur Speicherreduzierung im Vergleich zum Standardtrainingsverfahren Backpropagation-Through-Time (BPTT) bot. E-Prop wurde in Software auf einem FPGA-Prototyp der neuromorphen Hardware SpiNNaker2 implementiert. Dabei wurde ein RSNN zur Schlüsselwörterkennung mit dem Google Speech Commands Datensatz parallel auf 12 Prozessorkernen von SpiNNaker2 trainiert. Die Ergebnisse gleichen denen einer Referenz-Implementierung auf einer GPU in TensorFlow. Eine Abschätzung ergab, dass das Training auf einem SpiNNaker2 Chip 10x weniger Energie benötigt als auf einer GPU. Für das RSNN wurden weitere Studien durchgeführt, z.B., wurde der Speicherbedarf mit BPTT verglichen, der Nutzen von struktureller Sparsity und alternativen Zahlenformaten evaluiert, sowie weitere neue Algorithmen aus der Literatur analysiert. Ebenso

fand eine Studie zu Verbesserungen der SpiNNaker2-Architektur für diesen Anwendungsfall statt.

Danach wurde ein Hardware-Beschleuniger entwickelt, der die Rechenoperationen, die für das Training mit speicheroptimierten Lernverfahren, wesentlich schneller als in einer CPU ausführt. Dazu wurde zuerst ein Profiling für drei repräsentative KI-Modelle durchgeführt, um zu verstehen, für welche Operationen eine Beschleunigung am Sinnvollsten ist. Der sogenannte GMAC-Beschleuniger wurde als Hardware-IP-Block entwickelt, charakterisiert und in einem Test-Chip eines anderen Projekts realisiert. Der Beschleuniger ist für typische Rechenoperationen 30- bis 80-mal schneller als eine CPU.

Für den Radar Use-Case wurde sich in den Radardatensatz von Infineon eingearbeitet und Software zur Vorverarbeitung der Daten entwickelt. Nach einer Literaturstudie wurde ein Baseline-Algorithmus zur Personenzählung und -Verfolgung mit Radar implementiert. Dann wurde RANet, eine neue DNN-Architektur, entworfen, welche direkt auf den Rohdaten der Radarsensoren arbeitet, anstatt auf den Daten nach den sonst üblichen Fouriertransformationen. RANet wurde für den Datensatz trainiert und optimiert, und erreichte zuletzt eine um weiten bessere Vorhersagequalität als der Baseline-Algorithmus. Zuletzt wurde RANet auf dem SpiNNaker2 Chip implementiert und mit einer Implementierung auf dem Edge-System NVIDIA Orin Nano hinsichtlich Laufzeit und Energie verglichen. SpiNNaker2 ist 10-mal schneller und benötigt fast 1000-mal weniger Energie als das NVIDIA-System.

Parallel dazu wurden SNN- und DNN-Architekturen zur Gestenerkennung mit einer eventbasierten Kamera verglichen und evaluiert.

3. Wesentliche Ergebnisse

Die wesentlichen Ergebnisse der TU Dresden im Projekt ANDANTE sind:

1. Software-Implementierung des E-Prop-Lernverfahrens zum Trainieren von rekurrenten spikenden neuronalen Netzen auf dem neuromorphen Hardware-System SpiNNaker2 mit 10-fach geringerem Energieverbrauch als NVIDIA Hardware.
2. Hardware-Beschleuniger für effizientes, speicheroptimiertes Lernen, welcher Operationen zwischen 30 und 80-fach gegenüber normalen Prozessoren beschleunigt.
3. Entwicklung des KI-Modells RANet zur Personenzählung und -verfolgung mit Indoor-Radarsystemen, welches einen neuen Stand der Technik bezüglich Genauigkeit darstellt. Eine um fast 1000-fache Reduktion der Energie bei der Inferenz des RANet Modells auf der SpiNNaker2 Hardware gegenüber einem NVIDIA-System.

4. Zusammenarbeit mit anderen Forschungseinrichtungen

Im Use-Case 1.1 "Indoor Radar People Tracking and Counting" wurde eng mit der Infineon AG (Neubiberg) zusammengearbeitet, welche den Trainingsdatensatz bereitstellte. Ebenso gab es hierzu Absprachen mit der eesy-innovation GmbH (Unterhaching). Auf europäischer Ebene gab es kontinuierlich Absprachen und Austausch mit Projektpartnern, einerseits bei der Arbeit an Deliverables und andererseits zum Austausch bezüglich standardisierter Ansätze zum Benchmarking von KI-Edge-Hardware.

II. Ausführlicher Bericht

1. Ergebnisse im Detail

Im Folgenden beschreiben wir die Ergebnisse der TU Dresden im Detail. Zum besseren Verständnis beginnen wir mit einer Einführung der SpiNNaker2 Hardware. Danach folgen die Ergebnisse aus den Workpackages 3 und 5.

Kurzeinführung SpiNNaker2 Hardware

Die Hardware-Architektur von SpiNNaker 2 ist in Abbildung 1 dargestellt. Sie besteht aus 36 Quad Processing Elements (QPEs), von denen jedes wiederum 4 Processing Elements (PEs) enthält. Die QPEs sind in einem 2D-Gitter über ein durchsatzstarkes Network-on-Chip (NoC) miteinander verbunden, wobei die Weiterleitung in vier Richtungen zu jedem Nachbarn erfolgt. Dieses 2D-Gitter ermöglicht einen schnellen gegenseitigen Zugriff auf die Speicher aller PEs. Jedes PE enthält einen ARM M4F-Prozessor, 128 KB SRAM und mehrere Beschleuniger, die auf die Beschleunigung von spikenden und künstlichen neuronalen Netzen (SNN und DNN) ausgerichtet sind. Der Beschleuniger für maschinelles Lernen (MLA) bietet ein 16x4-MAC-Array zur Beschleunigung der Matrixmultiplikation und 2D-Convolution für 8-Bit- oder 16-Bit-Ganzzahlen. Der Beschleuniger holt Daten aus dem lokalen SRAM und dem globalen NoC und schreibt die Ergebnisse zurück in den lokalen SRAM. Eine detailliertere Beschreibung des Datenflusses einer älteren Version des Beschleunigers ist in [55] zu finden. Im äußersten NoC-Ring bietet der Chip eine Vielzahl von Peripherie- und Zusatzschnittstellen sowie Zugang zu einem 2 GB LPDDR4-DRAM mit einem Gesamtdurchsatz von 6,4 GB/s. Sechs serielle Chip-zu-Chip-Verbindungen und der SpiNNaker-Router ermöglichen die paketbasierte Kommunikation mit anderen Chips (in diesem Projekt nicht verwendet). Weitere Einzelheiten über die Systemarchitektur sind in [56] zu finden.

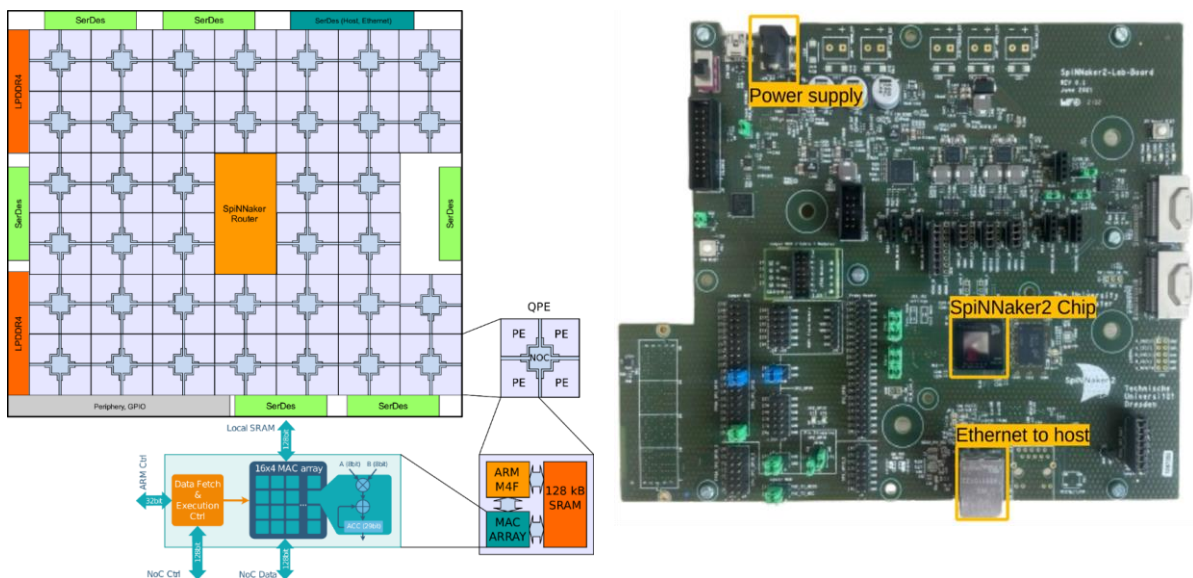


Abbildung 1: Übersicht SpiNNaker2 Chip und Testboard

Speicheroptimiertes Lernen auf SpiNNaker2 mit E-Prop

Das Hauptziel im Projekt war die Entwicklung und Umsetzung von speicheroptimiertem Lernen auf dem neuromorphen Hardware-System SpiNNaker2. Dazu standen zwei Paradigmen zur Auswahl, die beide von SpiNNaker2 unterstützt werden: Spikende Neuronale Netze (SNN) einerseits und künstliche tiefe Neuronale Netze (DNN) andererseits. Wir haben unterschiedliche Methoden zum speichereffizienten Lernen evaluiert und uns für das „E-Prop“-

Lernverfahren [14] entschieden. Mit E-Prop lassen sich rekurrente SNN zur Verarbeitung von sequentiellen Eingangsdaten (z.B. Audiosignale) trainieren, wobei eine ähnliche Genauigkeit wie mit dem Stand der Technik „Backpropagation-through-time“ (BPTT) erreicht wird. E-prop benötigt aber im Vergleich zu BPTT weniger Speicher und ist somit perfekt für das Lernen in eingebetteten Systemen geeignet.

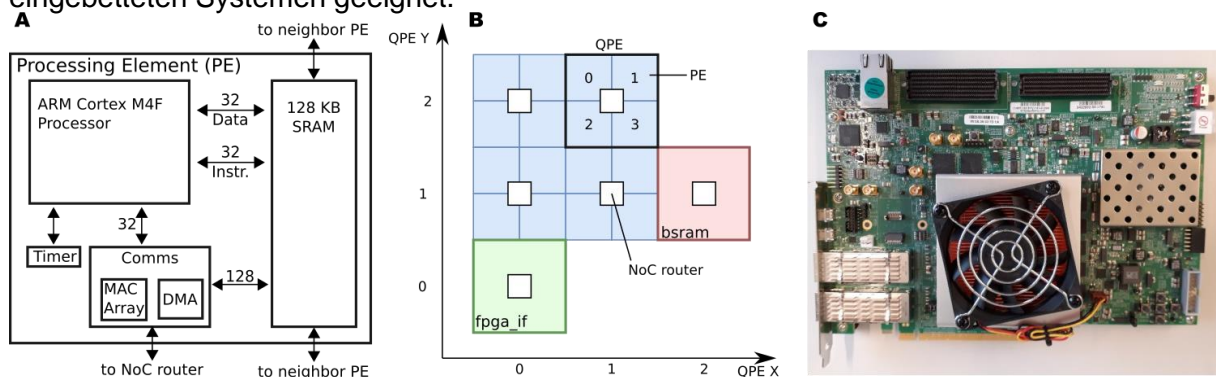


Abbildung 2: SpiNNaker 2 FPGA Prototyp. (A) Prozessorelement, (B) Reduzierte SpiNNaker2 Architektur mit 16 PEs auf 4 QPEs. (C) Xilinx Virtex UltraScale+ FPGA VCU118 Board. Aus [1].

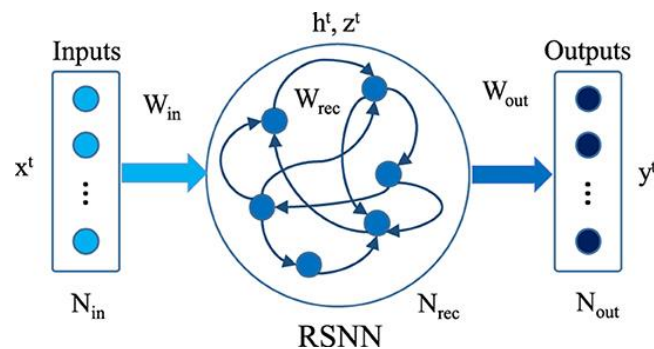


Abbildung 3: Architektur des rekurrenten spikenden neuronalen Netzes (RSNN). Aus [1].

SpiNNaker2 Implementierung

Verschiedene SNN wurden erfolgreich mit E-Prop auf dem SpiNNaker2 FPGA implementiert: Ausgangspunkt war der öffentlich verfügbare TensorFlow-Code von E-Prop. Zuerst wurde für ein kleines SNN das E-Prop Lernfahren in der Programmiersprache C auf einem x86 Computer implementiert und mit der TensorFlow-Variante verglichen. Dies war notwendig, um die korrekte Gradientenberechnung sicherzustellen, die in Tensorflow automatisch hinter den Kulissen geschieht und für den Nutzer schwer einsehbar ist. Danach wurde das C-Programm für den ARM-Core des SpiNNaker2 Systems angepasst und mit der Referenz verglichen.

Als echter Anwendungsfall wurde die Schlüsselwörtererkennung ausgewählt, wobei das Google Speech Command Dataset [15] als Benchmark dient. Dabei werden aus Audio-Streams zuerst die MFCC Koeffizienten berechnet und dann in das neuronale Netz zur Klassifikation gegeben. Das neuronale Netz besteht in unserem Fall aus einem Input Layer, einem rekurrenten Hidden Layer mit Adaptive Leaky-Integrate-and-Fire (ALIF) Neuronen sowie einem Output Layer zur Klassifikation, siehe Abbildung 3. Dann wurden Hyperparameterstudien in TensorFlow durchgeführt, um die optimale Netzwerkgröße und Trainingsparameter zu finden. Das gewählte Netzwerk hat 80 Inputs, 120 rekurrente Neuronen und 12 Output Neuronen. Die C-Implementierung für den ARM-Core wurde so angepasst, dass nun ALIF anstatt einfacher LIF Neuronen unterstützt werden. Das korrekte Training auf dem SpiNNaker2 Prototypen wurde zuerst für einen Core validiert.

Parallelisierung

Da das komplette Netz nicht auf einen Core passte, wurde die Anzahl der rekurrenten Neuronen zuerst auf 20 reduziert. Um das ganze Netzwerk zu implementieren und die Laufzeit zu verringern, wurde das Netzwerk auf 12 Cores parallelisiert. Abbildung 4 zeigt zwei

Möglichkeiten, wie das SNN auf mehrere Cores aufgeteilt werden kann. Es wurde sich für Variante (b) entschieden, weil dadurch die Rechenlast und Speichernutzung gleichmäßig auf alle PEs verteilt ist. Dabei berechnet jeder Core einen Teil der rekurrenten Neuronen und die zugehörigen eingehenden Synapsen. Es ist eine stetige Kommunikation und Synchronisation der Cores nach jedem Simulationszeitschritt notwendig.

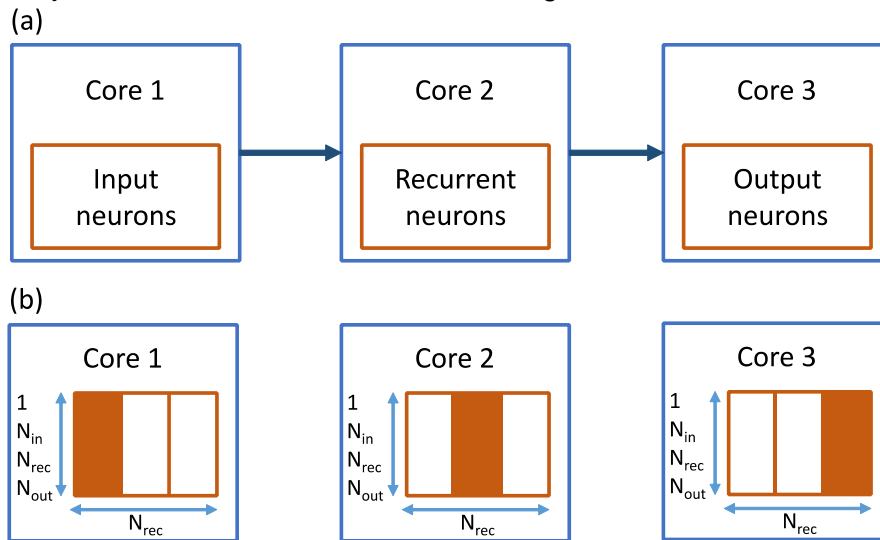


Abbildung 4: Parallelisierungsmöglichkeiten: (a) Jeder Layer wird auf einem Core realisiert, (b) Jeder Core berechnet einen Teil von allen Layern.

Das SNN konnte erfolgreich auf dem SpiNNaker2 FPGA trainiert werden und erreicht eine Genauigkeit von 91.2 %. Das Ergebnis ist zwar etwas schlechter als andere SNN, hat aber einen viel geringeren Speicher- und Rechenbedarf, wie Tabelle 1 zeigt:

Tabelle 1: Vergleich der E-Prop Implementierung mit anderen Ansätzen (vorläufiges Ergebnis)

Paper	Architecture, Training	# recurrent neurons	# params	Memory	MAC	Accuracy (%)
This work	ALIF, E-Prop	120	25K	680KB	50M	91.2
Yin et al.	ALIF, BPTT	256	167K	2.7MB	200M	92.1
Pellegrini et al.	NLIF, 3 layer Conv2D	-	130K	30MB	320M	94.5
Salaj et al.	LSNN with SFA, BPTT	2048	4M	120MB	9.3G	91.2

Die parallele Implementierung des rekurrenten SNN zur Schlüsselworterkennung (Google Speech Commands Dataset) auf dem SpiNNaker2 FPGA Prototypen wurde weiter optimiert. Mithilfe des „E-Prop“-Lernverfahrens wird das SNN auf der Hardware mit Echtzeitinput trainiert

(Kein Batch-learning) und erreicht die gleiche Genauigkeit wie die Referenzimplementierung in Software mit einer GPU, wie das folgende Bild zeigt.

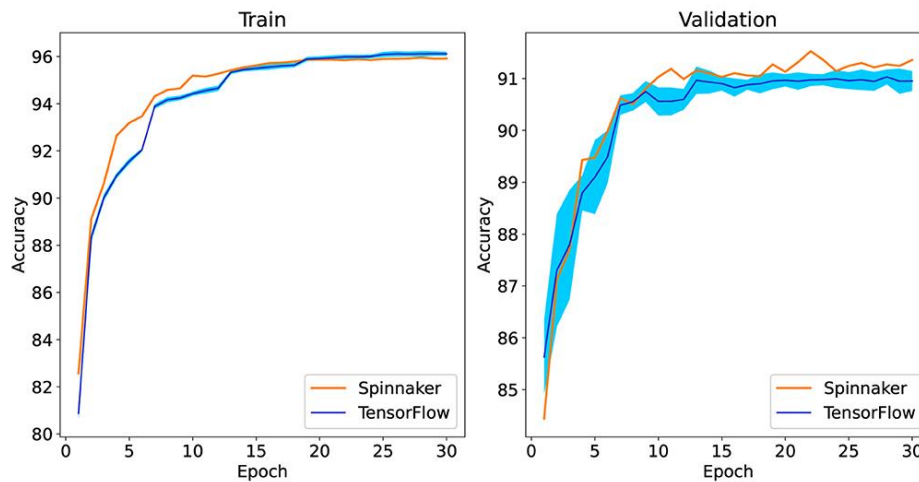


Abbildung 5: Vergleich der Train & Validation Accuracy für TensorFlow und SpiNNaker2. Aus [1].

Die Ergebnisse wurden Ende November 2022 im Fachjournal „Frontiers in Neuromorphic Engineering“ [1] veröffentlicht. Neben Details zur Implementierung enthält der Artikel auch die Ergebnisse der Exploration optimierter Hardwarearchitekturen, welche das Training mit E-Prop noch weiter beschleunigt. Es wurde zudem mit der Portierung von E-Prop auf den SpiNNaker2 Chip (mit 153 ARM Kernen anstatt 16 Kernen des FPGA) begonnen und eine erste Abschätzung des Energieverbrauchs für das Trainieren des SNN zur Schlüsselworterkennung durchgeführt: Demnach benötigt SpiNNaker2 zehn Mal weniger Energie als eine NVIDIA V100 GPU. Der TensorFlow Code für das Trainieren des SNN wurde als Open-Source Software auf Gitlab veröffentlicht: <https://gitlab.com/tud-hpsn/public/E-Prop-on-gsc>.

Hardware-Beschleuniger für speicheroptimiertes Lernen

Ein weiteres Ziel in Workpackage 3 war die prototypische Entwicklung eines Hardware-Beschleunigers für speicheroptimiertes Lernen. Die folgenden Abschnitte geben einen Überblick über die Anforderungen, die Architektur und die Ergebnisse der Charakterisierung.

Profiling

Um einen allgemeineren Hardwarebeschleuniger zu entwerfen und die dominanten Operationen herauszufinden, haben wir ein Profiling für das Training verschiedener neuronaler Netze durchgeführt. Dafür wurden die Algorithmen GRU (Gated Recurrent Unit), MLP (Multi-Layer-Perzeptron) und E-Prop ausgewählt.

GRU ist ein rekurrentes neuronales Netz, das für sequentielle Daten verwendet wird, MLP ist das Rückgrat der berühmten Transformer und E-Prop ist eine biologisch plausible Lernregel. Wir haben die Operationen in vier Hauptgruppen eingeteilt: Vektor-Matrix-Multiplikation, elementweise Operationen, Äußeres Produkt und andere für den Algorithmus spezifische Funktionen. Die Ergebnisse des Profiling sind in der folgenden Abbildung dargestellt:

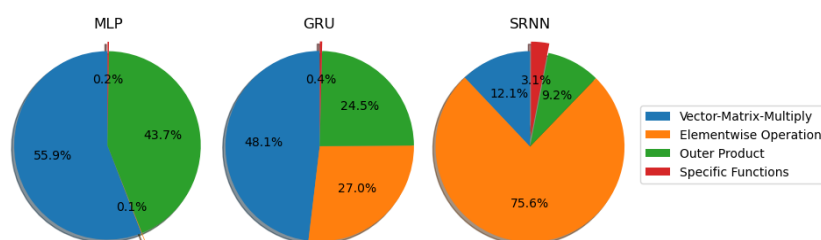


Abbildung 6: Profiling-Ergebnisse

Die Beschleuniger-Mikroarchitektur

Unsere Analyse führte zur Entwicklung eines neuartigen Beschleunigers, der als GMAC (General Multiplication and Addition Computation) bezeichnet wird. Dieser Name spiegelt seine Fähigkeit wider, verschiedene Operationen mit Skalaren, Vektoren und Matrizen zu bewältigen.

Die folgende Abbildung veranschaulicht die Mikroarchitektur des GMAC. Die Host-CPU speist Anweisungen in den dedizierten Speicher ein und triggert den Controller über ein bestimmtes Register, um Operationen zu initiieren. Der Controller interpretiert Anweisungen, entweder aus dem Speicher oder aus einer Registerdatei, mit Hilfe des Decoders. Durch die Extraktion von Details wie Vorgangstyp, Ein-/Ausgabeadressen und Datentypen koordiniert der Controller alle Einheiten, um die erforderlichen Berechnungen durchzuführen.

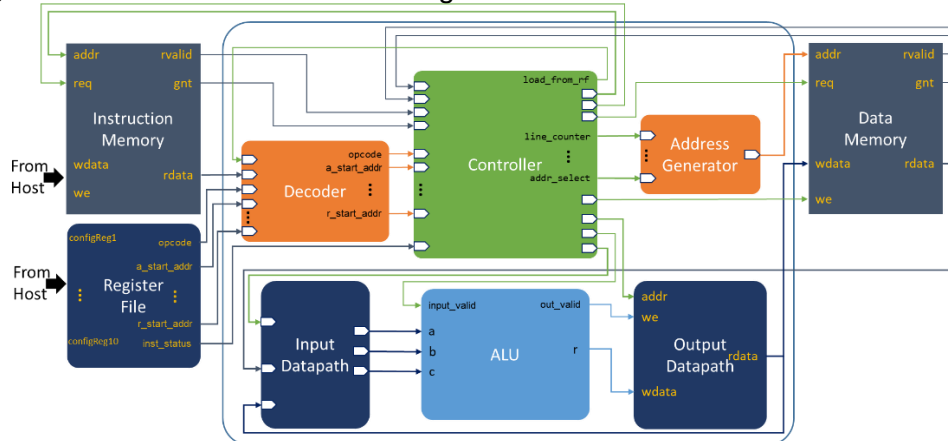


Abbildung 7: GMAC-Mikroarchitektur

Die ALU (Arithmetic Logic Unit) besteht aus acht bfloat16-Einheiten (16-Bit-Brain-Floating-Point), welche die Fused Multiply and Add (FMA) Operation berechnen kann. Diese Einheiten verarbeiten Additions-, Subtraktions-, Multiplikations- und Multiplikations-Akkumulationsoperationen (MAC) und unterstützen sowohl den Nächste-Nachbar- als auch den stochastischen Rundungsmodus. Dieses Projekt umfasste insbesondere die Entwicklung einer bfloat16 FMA-Einheit, die speziell auf das Deep-Learning-Training zugeschnitten ist.

System on Chip (SoC)

Der GMAC wurde in ein dediziertes System-on-Chip (SoC) aus einem anderen Projekt integriert, um die Fähigkeiten des entwickelten GMAC bewerten zu können. Dieses System enthält einen RISC-V-Kern, den CV32E40P, der mit einer eingebauten Gleitkommaeinheit (fpnew) und 128 KB On-Chip-Speicher ausgestattet ist. Der RISC-V-Core fungiert als Datenlieferant für den Beschleuniger und übernimmt Operationen, die nicht in den Aufgabenbereich des Beschleunigers fallen. Ein APB-Bus dient als Kommunikationskanal zwischen dem RISC-V-Core und dem Beschleuniger. Der SoC ist in der folgenden Abbildung dargestellt:

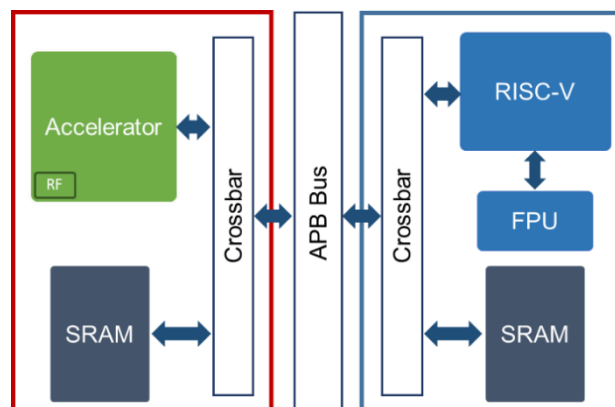


Figure 1: SoC Architektur

Wir haben den Beschleuniger in synthetisierbarem SystemVerilog HDL implementiert, alle Operationen in der Netzlistensimulation verifiziert und den ASIC-Flow bis zum Place-and-Route (PnR) mit GlobalFoundries 22nm FDSOI-Technologie ausgeführt. Die folgende Abbildung zeigt das Post-PnR-Layout des SoC und die Flächenaufteilung:

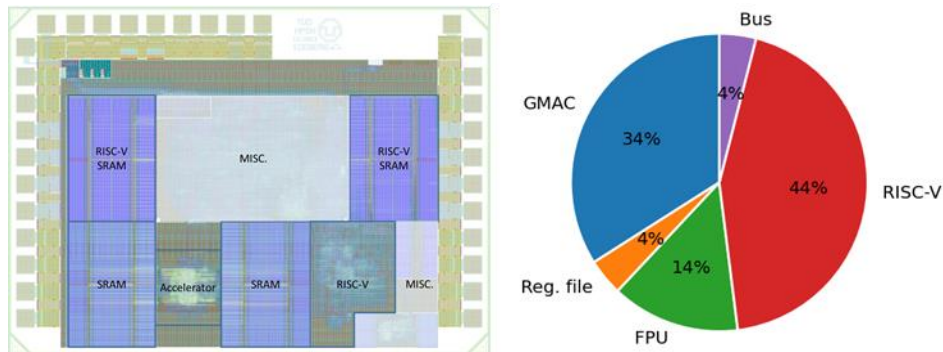


Figure 2: Chip-Layout (links) und Flächenaufteilung (rechts)

Performance

Wir haben den Beschleuniger mit einer Software-Implementierung auf einer RISC-V-CPU mit einem FPU-Beschleuniger verglichen. Für RISC-V wählten wir das Gleitkommaformat mit einfacher Genauigkeit, da es das schnellste Gleitkomma-Datenformat ist, das von der RISC-V-FPU und den GCC-Compilern unterstützt wird. Wir haben die Anzahl der Taktzyklen gemessen, die für die Ausführung der Operation erforderlich waren. Für RISC-V zählten wir also den RISC-V-Taktzyklus und für den Beschleuniger die Beschleuniger-Taktzyklen. Die folgende Abbildung zeigt die Beschleunigung des Beschleunigers gegenüber dem RISC-V für einzelne Operationen.

Operation	RISC-V (CLK)	GMAC (CLK)	Acceleration	FMA utilization
Scalar-Vector Elementwise Add/Sub/Mul	4058	127	32X	27%
Scalar-Matrix Elementwise Add/Sub/Mul	60722	2026	30X	42%
Vector-Vector Elementwise Add/Sub/Mul	4178	136	31X	27%
Vector-Matrix Elementwise Add/Sub/Mul	84086	2107	40X	42%
Outer Product	86765	1783	49X	50%
Matrix-Matrix Elementwise Add/Sub/Mul	76071	2674	28X	27%
Vector-Matrix Multiplication	91853	2101	44X	74%
Transpose Vector-Matrix Multiplication	91853	1129	81X	90%
ReLu	74253	987	75X	-
Step	68927	987	70X	-

Die folgende Abbildung zeigt den Geschwindigkeitszuwachs für verschiedene Trainingsarten des neuronalen Netzes (Vorwärts- und Rückwärtsdurchläufe):

Operation	RISC-V (CLK)	RISC-V + GMAC (CLK)	Acceleration
MLP	162150	8329	19X
GRU	559852	22963	24X
E-PROP	188770	14592	13X

Insgesamt ergibt sich durch den GMAC eine Beschleunigung zwischen 13x und 24x. Sobald der Chip samt Test-Leiterplatte im Sommer 2024 verfügbar ist, sollen zusätzlich noch Energiemessungen durchgeführt werden. Danach ist noch eine Journal-Veröffentlichung geplant [9].

Zusätzlich zum beschriebenen Beschleuniger wurden in einer studentischen Abschlussarbeit [13] weitere Architekturen für effizientes Lernen auf Edge-Systemen untersucht.

Radar Use-Case

In Workpackage 5 arbeitete die TU Dresden (TUD) am Use-Case 1.1 “Indoor Radar People Tracking and Counting” mit. Der Datensatz für den Use-Case wurde vom Projektpartner Infineon AG (IFAG) bereitgestellt. Ziel der TU Dresden war es, neuronale Netze für den Use-Case zu trainieren und auf dem SpiNNaker2 System zu implementieren und in einem Echtzeitsetup zu demonstrieren. Die Ergebnisse sollten hinsichtlich Genauigkeit, Ausführungszeit und Energie mit Algorithmen und Hardware-System des Stands der Technik verglichen werden.

Ziele

Die Aufgabe der Personenzählung und -verfolgung wird seit langem von einigen herkömmlichen Radaralgorithmen gelöst, aber die Leistung und Robustheit dieser Ansätze sind nicht zufriedenstellend. Daher wenden wir einen DNN-basierten Ansatz an, um die genaue Anzahl und Position von Zielen für die Lösung von Zähl- und Tracking-Aufgaben in Innenräumen vorherzusagen. Unser Ansatz war auf die folgenden Ziele ausgerichtet:

- Vorhersage mit Super-Resolution: Die vorhergesagte Range-Angle-Map soll eine räumliche Auflösung haben, die höher ist als die Auflösung, die mit gewöhnlichen Radarsignalverarbeitungsschritten (z. B. FFT¹) erreicht wird.
- Einsparung der Vorverarbeitung von Radarsignalen: Das Modell verwendet Radarrohdaten als Eingaben, ohne dass eine Vorverarbeitung erforderlich ist.
- Hohe Energieeffizienz: Geringer Stromverbrauch und geringe Latenzzeiten sollen durch die Nutzung der Machine-Learning-Beschleuniger im neuromorphen Chip SpiNNaker2 erreicht werden.

Demonstrator

Wir haben einen Deep Neural Network (DNN)-Ansatz gewählt, um die Aufgabe der Personenzählung und -verfolgung in Innenräumen mithilfe von Radarsensoren zu bewältigen. Die Trainingsdaten werden von der IFAG erfasst und zur Verfügung gestellt. Das DNN-Modell wird in einem GPU-Cluster der Universität trainiert und dann für die Demonstration mit der neuromorphen Multiprozessorplattform SpiNNaker2 bereitgestellt.

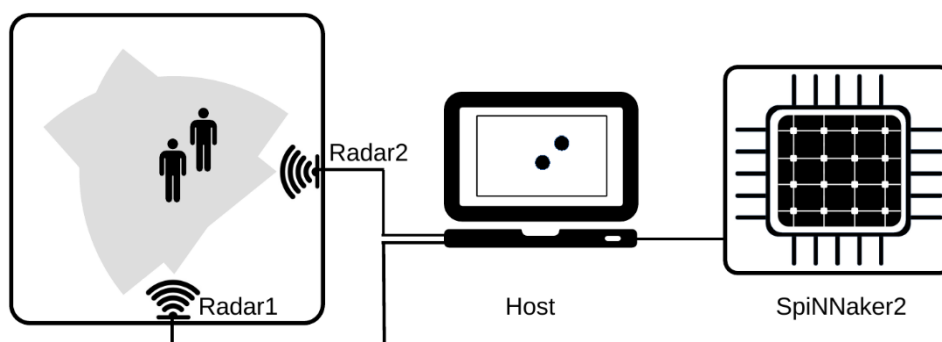


Abbildung 8: Übersicht Demonstrator

¹ Fast Fourier Transformation

Abbildung 8 zeigt einen Überblick über den Demonstrator. Die Testumgebung wird von 2 Infineon-Radaren BGT60TR13C überwacht, die jeweils über 1 Sendeantenne (Tx) und 3 Empfängerantennen (Rx) verfügen. Die Live-Daten werden an einen Laptop gesendet. Der Laptop spielt die Host-Rolle, um (1) die Radardaten aufgrund einer Schnittstellendiskrepanz (USB am Radarsensor, während Ethernet am SpiNNaker2) an die SpiNNaker2-Platine weiterzuleiten, (2) das Zähl- und Tracking-Ergebnis von SpiNNaker2 zu visualisieren. Das Demo-Backend, SpiNNaker2, führt das entwickelte DNN-Modell für die Vorhersage von Range-Angle-Maps aus und führt basierend auf dem vorhergesagten Bild CFAR-Erkennungen (Constant False Alarm Rate) und einen eigens entwickelten Tracking-Algorithmus durch.

Unser Ansatz stützt sich nicht auf eine klassische Vorverarbeitung von Radarsignalen, wie z.B. FFT in Reichweiten-, Doppler- oder Winkeldimensionen, sondern zerlegt die Aufgabe der Personenzählung und -verfolgung in Innenräumen in eine vorgelagerte Aufgabe (DNN-basierte Vorhersage der Range-Angle-Map) und zwei nachgelagerte Aufgaben (Zählung und Verfolgung). Wir verwenden ein DNN-Modell, um die ADC²-Rohdaten von Radaren zu interpretieren und qualitativ hochwertige Range-Angle-Maps vorherzusagen, die eine höhere Winkelauflösung aufweisen als herkömmliche FFT-basierte Methoden. Qualitativ hochwertige Vorhersagen stellen reduzierte Anforderungen an nachgelagerte Aufgaben, so dass durch den Einsatz einfacher und robuster Zähl- und Tracking-Algorithmen eine hohe Genauigkeit erreicht werden kann. Im Allgemeinen weist unser Ansatz folgende Vorteile auf:

- Direkte Verarbeitung von Radarrohdaten mit einem DNN-Modell.
- Kombinieren von Daten mehrerer Frames von mehreren Sensoren, um die Vorhersagefähigkeit zu verbessern.
- Erleichterung nachgelagerter Aufgaben, z. B. Zählen und Verfolgen, durch genaue Vorhersagen.
- Echtzeit-Sensordatenverarbeitung mit effizienter Implementierung des DNN-Modells auf neuromorpher Hardware.

Hardware- und Softwarekomponenten

In diesem Demonstrator wird der SpiNNaker2 Chip verwendet, welches auf einer Testleiterplatte montiert wurde, siehe Abbildung 1. Da unser DNN-Modell als kleines, ressourceneffizientes Modell konzipiert ist, werden in diesem Demonstrator nur 3 QPEs verwendet, in denen 2 QPEs das DNN-Modell ausführen und 1 QPE das Zählen und Nachverfolgen durchführt. Dies entspricht 8% aller Rechenkerne.

DNN-Model RANet

Unser Vorhersagemodell für Range-Angle-Maps, genannt „RANet“ (Range-Angle Net), ist ein räumlich-zeitliches Modell, das 4 Datenkanäle von 2 orthogonal positionierten Radarsensoren zusammenführt. Darüber hinaus beobachten wir, dass die Vorhersagequalität weiter verbessert werden kann, indem extrahierte Merkmale von mehr als einem Frame akkumuliert werden. Das RANet besteht aus einem 4-köpfigem Frontend und einem verketteten Backend, siehe Abbildung 9. Jeder Kopf des Modells, der ein CNN³ und ein RNN mischt, die sowohl räumliche Samples als auch temporale Radar-Chirps berücksichtigen, wird von einem PE verarbeitet, daher kann das gesamte Frontend in einem QPE (Quad Processing Element) realisiert werden. Die einzelnen Ausgänge der GRU-Schichten der Modell-Köpfe werden zum Eingangstensor für das Backend verschmolzen. Um die Speicherkosten für große Fully-Connected-Layer zu sparen, verwenden wir mehrere Convolution-Layer, um die Feature Maps der Kanäle im Backend zu mischen. Somit kann das gesamte Backend auf einem einzigen QPE implementiert werden. Am Ende prognostiziert das RANet eine 64x64 Range-Angle-Map unter Verwendung von 4 Frames von Radar-ADC-Daten.

² Analog-Digital-Wandler

³ Convolutional Neural Network

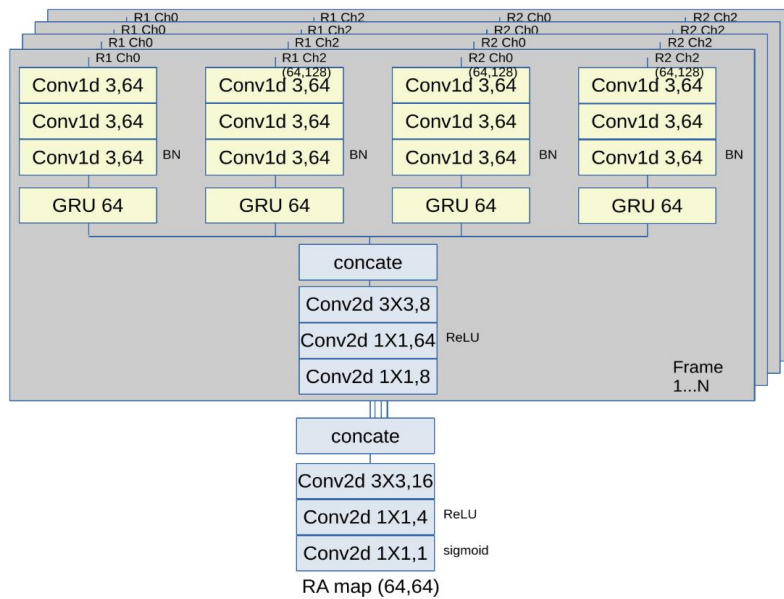


Abbildung 9: Architektur des RANet Modells

Zur Veranschaulichung zeigt Abbildung 10 drei Beispiel-Range-Angle-Maps, aufgeteilt in 3 Spalten. Die erste Spalte enthält die synthetischen Ground-Truth-Bilder (GT). Die Position jedes Ziels wird basierend auf der Kameraerkennung und den zugehörigen Labels erzeugt. In der mittleren Spalte werden die Vorhersagen des RANet Modells angezeigt. In der rechten Spalte sind schließlich Tracking-Felder für die erkannten Ziele eingezeichnet.

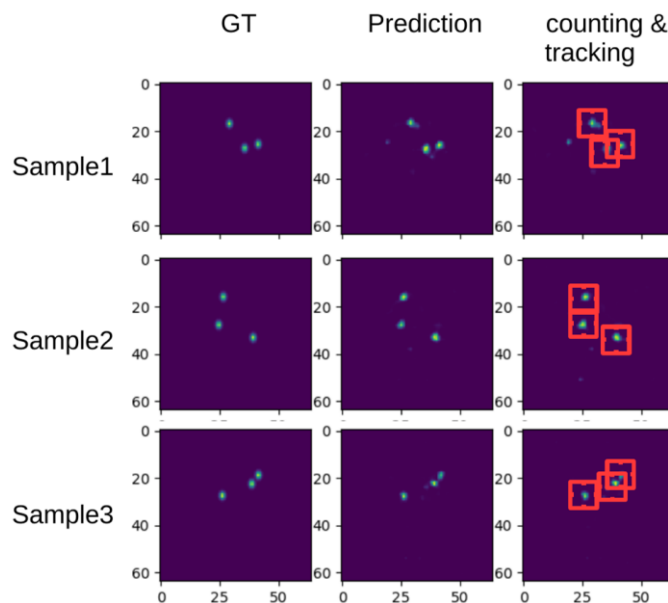


Abbildung 10: Visualisierung der Ground-Truth (GT, links)- und vorhergesagten Range-Angle-Map (Mitte) und des Zähl-/Tracking-Ergebnisses (rechts).

Da das RANet in der Lage ist, genaue Positionen von Zielen vorherzusagen, verwenden wir einen einfachen 2D-CFAR-Detektor für jede Zelle der Range-Angle-Map und gruppieren die erkannten Punkte innerhalb eines definierten Gruppierungsradius, um Ziele zu rekonstruieren, deren Nummer als Zählnummer gemeldet wird, und wir verwenden die Zählgewissheit, um die Zählleistung zu bewerten.

Ein robustes Tracking erfordert eine Aktualisierung der Erkennungen in jedem Frame, daher definieren wir eine Metrik für die Tracking-Konfidenz, die die Anzahl der aufeinanderfolgenden verfolgten Frames darstellt. Der Konfidenzwert kann durch fortgesetzte Erkennungen erhöht

oder verringert werden. Ein Tracker-Status enthält eine Tracking-ID zusammen mit ihrer Tracking-Konfidenz und der Tracking-Koordinate. Durch den Vergleich der Koordinate der Ground-Truth-Ziele und der Koordinate der Tracker wandeln wir die Auswertung des Trackings in eine binäre Klassifizierung um und verwenden den F1-Score, um die Tracking-Leistung zu bewerten. Der gesamte Arbeitsablauf ist in Abbildung 11 dargestellt.

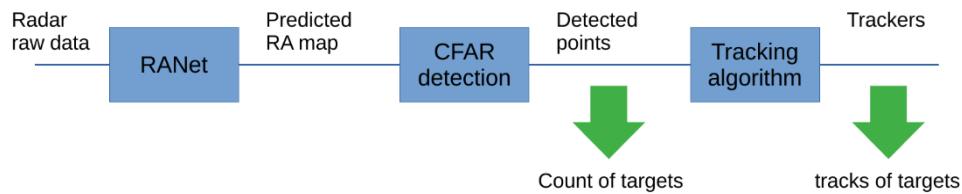


Abbildung 11: Arbeitsablauf der Zählung und Verfolgung

Evaluierungs- und Benchmarking-Ergebnisse

Wir vergleichen unseren Ansatz in Verbindung mit einem Baseline-Ansatz, der Rohsignale mit 3 Runden FFT vorverarbeitet, um die Range-Angle-Karte zu generieren. Die Metriken sind in der folgenden Tabelle aufgeführt:

Task	KPI Name	Accuracy/F1	Model Size [kB]	Inference Time [ms]	Inference Energy [mJ]
Counting	FFT baseline (PC)	35.0%	-	51	NA
Counting	RANet (SpiNNaker2)	93.0%	300	11	0.418
Counting	RANet (Orin Nano)	93.0%	1200	105	314
Tracking	FFT baseline (PC)	59.43%	-	58	NA
Tracking	RANet (SpiNNaker2)	97.73%	300	12	0.453
Tracking	RANet (Orin Nano)	97.73%	1200	111	356

Der RANet-basierte Ansatz übertrifft die herkömmliche auf Radarsignalverarbeitung basierende Methode deutlich (93 % vs. 35 % für die Zählung und 97 % vs. 60 % für die Positionserkennung) bei einer relativ kleinen Modellgröße. Mit einer Größe von 300 Kilobyte lässt sich das Modell problemlos in einem Embedded-Edge-Gerät umsetzen. Zum Vergleich wurde das Modell auch auf dem Nvidia Orin Nano implementiert und die Inferenzzeit und die Leistungsaufnahme gemessen. Für die Inferenz desselben Modells erreicht SpiNNaker2 eine 10-fache Beschleunigung und eine 100-fach reduzierte Leistungsaufnahme als mit dem Orin Nano, was zu einer ca. 1000-fachen Energiereduzierung pro Inferenz führt.

Im Allgemeinen wurden alle 3 Ziele für den Radar Use-Case erfüllt. Die vorhergesagte Range-Angle-Map stellt die Position von Zielen im Gegensatz zu herkömmlichen FFT-basierten Ansätzen klarer und genauer dar. Wir zeigen, dass die konventionelle Radarsignalverarbeitung bestehend aus vielen Schritten durch ein einziges DNN-Modell ersetzt werden kann. Die Benchmark-Ergebnisse zeigen, dass KI dazu beitragen kann, die Grenzen der herkömmlichen Radarsignalverarbeitung zu überwinden und Erkennungsaufgaben mit hervorragender Leistung zu lösen, während sie durch die Kombination von Software-Hardware-Co-Design eine Spitzenleistung in Bezug auf Latenz, Speicher- und Energieverbrauch erreicht.

Die TUD hat im Oktober 2023 auf der EdgeAI-Konferenz in Athen ein Hardware-Beschleunigungsverfahren zur CFAR-Erkennung [4] und das RANet-Modell [5] öffentlich vorgestellt. Erstere Arbeit wurde außerdem bei der MECO-Konferenz 2024 präsentiert [8]. Eine Journal-Veröffentlichung der RANet Implementierung auf SpiNNaker2 ist noch in Arbeit [10].

Darüber hinaus fanden in studentischen Arbeiten weitere Untersuchungen zur effizienten Sensordatenverarbeitung mit neuronalen Netzen statt:

- Komplexwertige neuronale Netze wurden zur Auflösung von Doppler-Ambiguitäten bei der Radarverarbeitungen entwickelt [12]. Hierzu wurden ein Datensatz [2] und ein Konferenzbeitrag [3] publiziert.
- Zur Gestenerkennung mittels event-basierter Kameras wurden unterschiedliche Methoden zum Trainieren von spikenden neuronalen Netzen untersucht [11]. Die Arbeit wird in 2024 im EU-Projekt PRIMI fortgesetzt.

Abweichungen von der Vorhabensbeschreibung

Die Aufgaben aus der Vorhabensbeschreibung in den Arbeitspaketen wurden erfolgreich abgeschlossen. Die Integration des GMAC Hardware-Beschleunigers in einen Test-Chip geht sogar über die Planung hinaus.

2. Zusammenfassung des zahlenmäßigen Nachweises

Im Projekt wurden verausgabt:

- Mittel für Personal in Höhe von 337.244,60 €. Dies entspricht einem Umfang von 54,77 Personenmonaten.
- Mittel für Verbrauchsmaterial in Höhe von 3.718,26€.
- Mittel für Gegenstände lt. Geräteliste in Höhe von 13.032,63€.

Das Personal, Verbrauchsmaterial und die Geräte wurden zur Ausführung der oben dargestellten Arbeiten verwendet.

3. Verwertungsplan

a) Erfindungen/Schutzrechtsanmeldungen und erteilte Schutzrechte

Keine.

b) Wirtschaftliche Erfolgsaussichten nach Projektende

Im Jahr 2021 wurde die Firma SpiNNcloud Systems GmbH ausgegründet, deren Geschäftsbereich die kommerzielle Verwertung der SpiNNaker2 Hardware ist. Dazu gab es Transfer von IP von der TU Dresden an die Firma (Hardware-Design und Software entwickelt in anderen Projekten). Unter anderem will SpiNNcloud Systems zukünftig viele SpiNNaker2-Cloud-Systeme verkaufen und als Plattform zur Verfügung stellen. Auch die Entwicklung von kleineren Edge-Systemen ist geplant. Die Firma ist gerade durch Förderprojekte (z.B. EIC Transition über 2,5 Millionen €) und Aufträge in der Wachstumsphase. Dies steigert auch die Aussichten zur wirtschaftlichen Verwertung der ANDANTE Ergebnisse, weil dies zu einer größeren Verbreitung von SpiNNaker2-Systemen weltweit führt und dies ebenfalls den Reifegrad der Software erhöht. Großes Potenzial bieten dazu die Ergebnisse für speicheroptimiertes Lernen auf SpiNNaker2, die für kleine KI-Modelle ein 10-fach höhere Energieeffizienz aufweisen konnten als Standard NVIDIA GPUs.

Neben einer möglichen wirtschaftlichen Verwertung der ANDANTE Ergebnisse durch SpiNNcloud Systems durch Weiterentwicklung der Algorithmen und Bereitstellung von Software für Kunden, stellen sowohl das E-Prop-Training als auch die effiziente Radarverarbeitung mit SpiNNaker2 „Success-Stories“ dar, die das Potenzial der SpiNNaker2 Hardware demonstrieren.

c) **Wissenschaftliche und/oder technische Erfolgsaussichten nach Projektende**

Die TU Dresden weist in ANDANTE die folgenden wissenschaftlich-technischen Ergebnisse auf:

- Lernen von rekurrenten SNN mit dem E-Prop-Lernverfahren auf der SpiNNaker2-Hardware: Die Publikation Rostami et al 2022 [1] findet in der akademischen Landschaft viel Beachtung (16 Zitate, Stand August 2024). Auch wenn rekurrente neuronale Netze (RNN) in der Zwischenzeit für viele Tasks von Transformer-basierten Neuronalen Netzen überboten wurden, erleben RNN durch sogenannte State-Space-Modelle (SSM) eine Wiederbelebung. Die TU Dresden kann hierzu Erfahrung im Training von RNN nutzen und auf SSM anwenden, um diese effizient auf neuromorpher Hardware wie SpiNNaker2 zu trainieren und auszuführen.
- Hardware-Beschleuniger für speicheroptimiertes Lernen: Hierzu soll es eine Veröffentlichung mit den Messergebnissen des Test-Chips geben [9]. Das wissenschaftliche und technische Knowhow fließt zudem in die Lehre in den Kurs „Deep Neural Network Hardware“ ein und bildet die Grundlage für weitere Forschung im Bereich von Hardware-Beschleunigern für Lernen.
- Radarsignalverarbeitung mit DNN zur Personenzählung und -verfolgung und Umsetzung auf SpiNNaker2: Die Ergebnisse zeigen sowohl bezüglich der Genauigkeit des Algorithmus als auch hinsichtlich der Effizienz (bis zu 1000x weniger Energie) eine wesentliche Verbesserung des Stands der Technik. Die Ergebnisse hierzu sollen in einem wissenschaftlichen Journal veröffentlicht werden [10]. Wir gehen davon aus, dass insbesondere die verwendete Modell-Architektur Nachahmung in der akademischen Welt finden wird. Die Entwicklung eines dauerhaften Demonstrators zur Radarverarbeitung mit SpiNNaker2 wird zurzeit erwogen. Dieser könnte z.B. für die Lehre und für Öffentlichkeitsarbeit Verwendung finden.

Die Ergebnisse aus ANDANTE leisten einen wesentlichen Beitrag zum Fortschritt des Stands der Technik im Bereich von KI-Edge-Anwendungen. Dadurch werden einerseits neue Möglichkeiten geschaffen (bessere Qualität von KI als auch bessere Effizienz), andererseits wurde und wird damit auch die Sichtbarkeit der Professur als Expertin und Technologietreiberin in diesem Bereich erhöht.

d) **Wissenschaftliche und wirtschaftliche Anschlussfähigkeit**

Das gewonnene Knowhow und die Ergebnisse aus ANDANTE bilden die Grundlage für weitere Forschungsprojekte. Namentlich zu nennen sind dabei:

- BMWK Projekt ESCADE „Energy-Efficient Large-Scale Artificial Intelligence for Sustainable Data Centers“, gestartet im Mai 2023. Die Ergebnisse zum speicheroptimierten Lernen bilden hierzu eine wichtige Grundlage. In dem Projekt soll der Einsatz von neuromorpher Hardware wie SpiNNaker2 in KI-Rechenzentren für maximaler Ressourceneffizienz untersucht werden. Unter anderem soll dabei das effiziente Trainieren von großen KI-Modellen für die natürliche Sprachverarbeitung auf SpiNNaker2-Hardware umgesetzt werden, wobei zum Teil auf den ANDANTE-Erkenntnissen aufgebaut werden kann.
- EU-Projekt PRIMI „Performance in Robot Interaction via Mental Imagery“, gestartet im Januar 2024. SpiNNaker2 Chips sollen im Projekt mit humanoiden Robotern integriert werden und dort Teile der Signalverarbeitung und des Lernens übernehmen. Hier kann zum einen auf dem Wissen zu speicheroptimierten Algorithmen für On-Chip-Lernen aufgebaut werden. Zum anderen kann die Erfahrung zur Anbindung von eventbasierten Kameras und deren Verarbeitung wiederverwendet werden.

Darüber hinaus plant die TU Dresden die Entwicklung der nächsten Generation von SpiNNaker Systemen. „Spinnaker3“ soll den Fokus auf effiziente, neuroinspirierte allgemeine Künstliche Intelligenz (AGI) legen. Der Hardwarebeschleuniger aus

ANDANTE soll dafür weiterentwickelt werden. Hierzu wird zurzeit eine Projektskizze entworfen.

Mit ANDANTE hat die Professur ihre Expertise und Sichtbarkeit im Bereich der Anwendung der SpiNNaker2 Hardware zur effizienten Radarsignalverarbeitung und zum speichereffizienten On-Chip-Lernen vergrößert. Dies bildet die beste Grundlage für weitere Forschung in den betroffenen Bereichen neuromorphe Hardware, On-Chip-Lernen und effiziente Sensordatenverarbeitung.

4. Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Algorithmen für speichereffizientes on-device Lernen

In den letzten Jahren wurden erhebliche Fortschritte bei Online-Lernalgorithmen erzielt, die speziell für neuromorphe Systeme entwickelt wurden. Diese Algorithmen sind von biologischen Mechanismen inspiriert und eignen sich für das Training von Spiking Neural Networks (SNNs). Zenke und Neftci (2021) [16] schlugen ein Framework vor, das von rekurrenten neuronalen Netzen (RNNs) für das Online-Lernen auf neuromorpher Hardware inspiriert ist. Sie erreichten dies, indem sie die Jacobi-Matrix in implizite (spärliche) und explizite (dichte) Komponenten trennten und den rechenintensiven expliziten Term eliminierten. Kaiser et al. (2020) [17] stellten DECOLLE vor, das schichtweise lokale Readouts auf Basis von SuperSpike [18] nutzt, um Gradienten effizient innerhalb der Hardware selbst zu berechnen. Bohnstingl et al. (2022) [19] stellten OSTL vor, einen Ansatz, der den Gradientenfluss in räumliche, zeitliche und gemischte Komponenten unterteilt. Durch das Ignorieren der gemischten Komponente demonstrierten sie die Möglichkeit des Online-Trainings für Deep Spiking Recurrent Networks (SRNNs). Wunderlich et al. (2021) [20] entwickelten EventProp, eine Methode zur Backpropagation-Berechnung in ereignisbasierten SNNs. Diese Methode verwendet die adjungierte Methode und beinhaltet partielle Ableitungssprünge, um eine Backpropagation durch diskrete Spike-Ereignisse ohne Approximationen zu ermöglichen. Pehle et al. (2023) [21] haben EventProp erfolgreich auf der neuromorphen Hardwareplattform BrainScaleS-2 implementiert [22]. Summe et al. (2023) [23] schlugen ein Online-Training mit postsynaptischen Schätzungen (OTPE) für das Training von Feed-Forward-SNNs vor. Dieser Ansatz approximiert Real-Time Recurrent Learning (RTRL), indem er die zeitliche Dynamik innerhalb des Netzwerks berücksichtigt.

Hardware für speichereffizientes on-device Lernen

Bei neuromorpher Hardware, die speziell für effizientes Lernen auf dem Gerät entwickelt wurde, wurden mehrere Fortschritte erzielt. Frenkel und Indiveri (2022) [24] stellten den ReckOn-Chip vor, der eine vereinfachte Variante von E-Prop verwendet. Dieses Design nutzt Platz, Zeit, Lokalität und Sparsamkeit bei Gewichtsaktualisierungen, um den Speicherbedarf zu reduzieren. Tang et al. (2023) [25] schlugen SENECA vor, einen vollständig digitalen neuromorphen Prozessor, der auf der RISC-V-Architektur aufbaut und über einen Ereignisgenerator verfügt. Sie berichteten von Verbesserungen der Energieeffizienz beim Training eines einfachen SRNN-Netzwerks mit dem E-Prop-Algorithmus.

Auch einige neuartige KI-Hardware wurde von Technologieunternehmen eingeführt. Lee et al. (2022) [26] stellten den KI-Chip der dritten Generation von IBM vor, der aus zwei Corelets mit jeweils vier Kernen besteht, die durch einen Ringbus mit hoher Bandbreite miteinander verbunden sind. Cerebras [27] entwickelte den größten Chip auf einem einzigen Wafer mit 850.000 KI-Kernen. Dieser Chip funktioniert im Wesentlichen als riesiges Matrix-Multiplikationsarray. Insbesondere behebt der Cerebras WSE-2 einen großen Engpass beim Deep-Learning-Training, indem er verteilten internen Speicher (SRAM) mit deutlich schnelleren Zugriffszeiten im Vergleich zu herkömmlichem externem Speicher (DRAM) verwendet. SambaNova stellte die Cardinal SN10 (Prabhakar et al., 2022) [28] vor, eine rekonfigurierbare Dataflow Unit (RDU), die mehrere Kacheln enthält, die jeweils mit Verarbeitungs- und Speichereinheiten ausgestattet sind. Tenstorrent (Vasiljevic et al., 2021)

[29] hat die Wormhole-Architektur entwickelt, die über mehrere Tensix-Kerne verfügt. Jeder Tensix Core besteht aus verschiedenen Verarbeitungselementen und Speichereinheiten, die effiziente Datentypen wie float16 und bfloat16 unterstützen.

Da große Batch-Größen in Deep-Learning-Modellen zu einem erheblichen Rechenaufwand für die Allgemeine Matrixmultiplikation (GEMM) führen, haben sich mehrere Forschungsbemühungen auf die Beschleunigung dieses Vorgangs konzentriert. Qin et al. (2020) [30] schlugen SIGMA (Sparse and Irregular GEMM Accelerator) vor, eine spezialisierte Hardware-Einheit, die für die effiziente Multiplikation von unregelmäßigen dünnbesetzten Matrizen entwickelt wurde, die für bestimmte Algorithmen des maschinellen Lernens von entscheidender Bedeutung ist. Samajdar et al. (2022) [31] stellten SARA (Self-Adaptive Reconfigurable Arrays) vor, einen flexiblen systolischen Array-Beschleuniger. Dieses Design ermöglicht es der Hardware, ihre Konfiguration dynamisch an die unterschiedlichen Rechenanforderungen verschiedener Aufgaben des maschinellen Lernens anzupassen. Darüber hinaus wurden mehrere Forschungsarbeiten vorgestellt, die auf einem ähnlichen Ansatz basieren, wie z.B. VEGETA [32], GAMMA [33] und RASA [34].

Personenzählung und -verfolgung mittels Radarsensoren

In den letzten Jahren wurde die Erkennung von Personen in Innenräumen, die mit Zähl- und Verfolgungsaufgaben verbunden sind, mit verschiedenen Lösungen angegangen. In der Frühphase können Radardaten als Mikro-Doppler-Bewegungen, Makro-Doppler-Bewegungen oder vitale Doppler [35] für Detektionen charakterisiert werden, worauf dann eine klassische Detektionspipeline aus einem CFAR-Detektor und einem DBSCAN-basierten Clustering folgt [36]. Auf der anderen Seite verbessern Yamada et al. [37] die menschliche Erkennungsrate aus der räumlichen Perspektive, indem sie die Khatri-Rao-Matrixprodukttransformation verwenden, um eine hochauflösende Range-Angle-Map zu erstellen. Konventionelle Techniken des maschinellen Lernens wie die Principle Component Analysis (PCA) und die Support Vector Machine (SVM) wurden in [38] ebenfalls für die Insassenerkennung eingesetzt. Mit der bemerkenswerten Leistung von Deep Learning nutzten Stephan et al. [39] ein tiefes residuales U-Net für Smart-Home-Anwendungen, und Mauro et al. [40] adressierten die Personenzählung mittels Few-Shot-Learning.

Als natürliche Erweiterung der Detektionstechnologie werden Tracking-Techniken immer auch ausgiebig erforscht. Eine einfache Metrik zur Messung der Tracking-Leistung ist die Verwendung des euklidischen Abstands zwischen dem Ziel und der Tracker-Koordinate [41][42]. Auf dieser Grundlage ist das filterbasierte adaptive Tracking der am weitesten verbreitete Ansatz, einschließlich der alpha-beta-filterbasierten Methode [43] und des Kalman-filterbasierten Trackings [44][45][46][47][48]. Einige andere Forscher formulieren das konsekutive Tracking-Problem in ein gewichtetes bipartites Graphen-Matching-Problem um und schlugen den KM-Algorithmus [49] und die Markov-Ketten-Monte-Carlo-Datenassoziation [50] vor, um es zu lösen.

Die oben genannten Arbeiten zielen ausschließlich auf die Millimeterwellen-Radarerfassung ab. Wenn heterogene Sensoren beteiligt sind, wird von Ayudogdu [51] ein multimodales Cross-Learning vorgeschlagen, das Radar und Kamera kombiniert. Stephan et al. verarbeiten Radardaten mit Kamerawissensdestillation für die Personenzählung [52], während in [53][54] Kameradaten mit Radardaten für die Personenverfolgung fusioniert werden.

5. Veröffentlichungen

Journals/Konferenzen/Preprints

1. Rostami, A., Vogginger, B., Yan, Y., & Mayr, C. G. (2022). E-prop on SpiNNaker 2: Exploring online learning in spiking RNNs on neuromorphic hardware. *Frontiers in Neuroscience*, 16, 1018006. <https://www.frontiersin.org/articles/10.3389/fnins.2022.1018006/full>
2. Jiawei Li, Chen Liu. (2022). Doppler Ambiguity Dataset. IEEE Dataport. <https://dx.doi.org/10.21227/wet7-gc70>
3. Liu, C., Li, J., Gonzalez, H. A., Vogginger, B., & Mayr, C. (2023, May). Complex-Valued Neural Networks for Doppler Disambiguation in FMCW Radars. In *2023 24th International Radar Symposium (IRS)* (pp. 1-10). IEEE. <https://ieeexplore.ieee.org/document/10172424>
4. Chen Liu (2023). CA-CFAR is Convolution: Fast Target Detection With Machine Learning Accelerator. European Conference on EDGE AI Technologies and Applications (EEAI). <https://edge-ai-tech.eu/wp-content/uploads/2023/11/2023-10-18 Presentation E02 2 Conference EDGE-AI Athens EEAI 2023.pdf>
5. Chen Liu (2023). RANet: An Autoencoder for High-Resolution Range-Angle Maps in Multistatic Radar. European Conference on EDGE AI Technologies and Applications (EEAI). <https://edge-ai-tech.eu/wp-content/uploads/2023/11/2023-10-19 Presentation B03 3 Conference EDGE-AI Athens EEAI 2023.pdf>
6. Yik, J., Ahmed, S. H., Ahmed, Z., Anderson, B., Andreou, A. G., Bartolozzi, C., ..., Mayr, C., ..., Vogginger, B., ... & Reddi, V. J. (2023). NeuroBench: Advancing neuromorphic computing through collaborative, fair and representative benchmarking. *arXiv preprint arXiv:2304.04640*. <https://doi.org/10.48550/arXiv.2304.04640>, submitted to *Nature Communications*
7. Bernhard Vogginger (2024). Advancing Neuromorphic Computing with NeuroBench. Workshop "Driving Next-Gen Edge AI Technologies" at HiPEAC 2024. <https://edge-ai-tech.eu/wp-content/uploads/2024/02/2024-01-17 Presentation S2 P1 Workshop Next Gen EAI HiPEAC 2024 Bernhard Vogginger.pdf>
8. Liu, C., Kelber, F., Vogginger, B., & Mayr, C. (2024, June). CA-CFAR is Convolution: Fast Target Detection with Machine Learning Accelerator. In *2024 13th Mediterranean Conference on Embedded Computing (MECO)* (pp. 1-6). IEEE. <https://doi.org/10.1109/MECO62516.2024.10577789>
9. Rostami, A., Vogginger, B., Guo, L., Kelber, F., Dixius, A., Scholze, S., & Mayr, C. G., A Flexible online Learning Hardware Accelerator for Spiking RNNs. *In Vorbereitung*
10. Liu, C., Arfa, S., Vogginger, B., & Mayr, C.. RANet: An Autoencoder for High-Resolution Range-Angle Maps in Multistatic Radar. *In Vorbereitung*

Studentische Arbeiten

11. Sirine Arfa (2023). Real-time sensor processing using the SpiNNaker2 neuromorphic multi-processor system, Master Thesis, Sup'Com Tunis
12. Jiawei Li (2023). Komplexes neuronales Netz zur Bestimmung der eindeutigen Geschwindigkeit in Fahrzeugradarsystemen, Diplomarbeit, TU Dresden
13. Fabian Stoppok (2023). Speedup of neural network training algorithms with a floating point accelerator for a 32bit RISC-V CPU, Diplomarbeit, TU Dresden

Weitere Referenzen

14. Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., & Maass, W. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1), 3625.
15. Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*.
16. F. Zenke and E. O. Neftci, "Brain-Inspired Learning on Neuromorphic Substrates," *Proc. IEEE*, vol. 109, no. 5, pp. 935–950, May 2021, doi: 10.1109/JPROC.2020.3045625.
17. J. Kaiser, H. Mostafa, and E. Neftci, "Synaptic Plasticity Dynamics for Deep Continuous Local Learning (DECOLLE)," *Front. Neurosci.*, vol. 14, p. 424, 2020, doi: 10.3389/fnins.2020.00424.
18. F. Zenke and S. Ganguli, "SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks," *Neural Comput.*, vol. 30, no. 6, pp. 1514–1541, Jun. 2018, doi: 10.1162/neco_a_01086.
19. T. Bohnstingl, S. Woźniak, A. Pantazi, and E. Eleftheriou, "Online Spatio-Temporal Learning in Deep Neural Networks," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2022, doi: 10.1109/TNNLS.2022.3153985.
20. T. C. Wunderlich and C. Pehle, "Event-based backpropagation can compute exact gradients for spiking neural networks," *Sci. Rep.*, vol. 11, no. 1, p. 12829, Jun. 2021, doi: 10.1038/s41598-021-91786-z.
21. C. Pehle, L. Blessing, E. Arnold, E. Müller, and J. Schemmel, "Event-based Backpropagation for Analog Neuromorphic Hardware." arXiv, Feb. 13, 2023. doi: 10.48550/arXiv.2302.07141.
22. C. Pehle *et al.*, "The BrainScaleS-2 Accelerated Neuromorphic System With Hybrid Plasticity," *Front. Neurosci.*, vol. 16, 2022, Accessed: Jan. 16, 2024. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fnins.2022.795876>
23. T. Summe, C. J. Schaefer, and S. Joshi, "Estimating Post-Synaptic Effects for Online Training of Feed-Forward SNNs." arXiv, Nov. 07, 2023. doi: 10.48550/arXiv.2311.16151.
24. C. Frenkel and G. Indiveri, "ReckOn: A 28nm Sub-mm² Task-Agnostic Spiking Recurrent Neural Network Processor Enabling On-Chip Learning over Second-Long Timescales," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb. 2022, pp. 1–3. doi: 10.1109/ISSCC42614.2022.9731734.
25. G. Tang *et al.*, "SENECA: building a fully digital neuromorphic processor, design trade-offs and challenges," *Front. Neurosci.*, vol. 17, 2023, Accessed: Dec. 12, 2023. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fnins.2023.1187252>
26. S. K. Lee *et al.*, "A 7-nm Four-Core Mixed-Precision AI Chip With 26.2-TFLOPS Hybrid-FP8 Training, 104.9-TOPS INT4 Inference, and Workload-Aware Throttling," *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 182–197, Jan. 2022, doi: 10.1109/JSSC.2021.3120113.
27. G. Lauterbach, "The Path to Successful Wafer-Scale Integration: The Cerebras Story," *IEEE Micro*, vol. 41, no. 6, pp. 52–57, Nov. 2021, doi: 10.1109/MM.2021.3112025.
28. R. Prabhakar, S. Jairath, and J. L. Shin, "SambaNova SN10 RDU: A 7nm Dataflow Architecture to Accelerate Software 2.0," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb. 2022, pp. 350–352. doi: 10.1109/ISSCC42614.2022.9731612.
29. J. Vasiljevic *et al.*, "Compute Substrate for Software 2.0," *IEEE Micro*, vol. 41, no. 2, pp. 50–55, Mar. 2021, doi: 10.1109/MM.2021.3061912.
30. E. Qin *et al.*, "SIGMA: A Sparse and Irregular GEMM Accelerator with Flexible Interconnects for DNN Training," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb. 2020, pp. 58–70. doi: 10.1109/HPCA47549.2020.00015.
31. A. Samajdar, E. Qin, M. Pellauer, and T. Krishna, "Self adaptive reconfigurable arrays (SARA): learning flexible GEMM accelerator configuration and mapping-space using ML," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, in DAC '22. New York, NY, USA: Association for Computing Machinery, Aug. 2022, pp. 583–588. doi: 10.1145/3489517.3530506.
32. G. Jeong *et al.*, "VEGETA: Vertically-Integrated Extensions for Sparse/Dense GEMM Tile Acceleration on CPUs," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, Feb. 2023, pp. 259–272. doi: 10.1109/HPCA56546.2023.10071058.
33. G. Zhang, N. Attaluri, J. S. Emer, and D. Sanchez, "Gamma: leveraging Gustavson's algorithm to accelerate sparse matrix multiplication," in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, in ASPLOS '21. New York, NY, USA: Association for Computing Machinery, Apr. 2021, pp. 687–701. doi: 10.1145/3445814.3446702.

34. G. Jeong *et al.*, "RASA: Efficient Register-Aware Systolic Array Matrix Engine for CPU," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA: IEEE Press, Dec. 2021, pp. 253–258. doi: 10.1109/DAC18074.2021.9586257.
35. A. Santra, R. V. Ulaganathan, and T. Finke, "Short-Range Millimetric-Wave Radar System for Occupancy Sensing Application," *IEEE Sens. Lett.*, vol. 2, no. 3, pp. 1–4, Sep. 2018, doi: 10.1109/LESENS.2018.2852263.
36. J. Weiss, R. Perez, and E. Biebl, "Improved People Counting Algorithm for Indoor Environments using 60 GHz FMCW Radar," in *2020 IEEE Radar Conference (RadarConf20)*, Florence, Italy: IEEE, Sep. 2020, pp. 1–6. doi: 10.1109/RadarConf2043947.2020.9266607.
37. H. Yamada and T. Horiuchi, "High-resolution Indoor Human detection by Using Millimeter-Wave MIMO Radar," in *2020 International Workshop on Electromagnetics: Applications and Student Innovation Competition (iWEM)*, Makung, Penghu, Taiwan: IEEE, Aug. 2020, pp. 1–2. doi: 10.1109/iWEM49354.2020.9237397.
38. M. Alizadeh, H. Abedi, and G. Shaker, "Low-cost low-power in-vehicle occupant detection with mm-wave FMCW radar," in *2019 IEEE SENSORS*, Montreal, QC, Canada: IEEE, Oct. 2019, pp. 1–4. doi: 10.1109/SENSORS43011.2019.8956880.
39. M. Stephan and A. Santra, "Radar-Based Human Target Detection using Deep Residual U-Net for Smart Home Applications," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Boca Raton, FL, USA: IEEE, Dec. 2019, pp. 175–182. doi: 10.1109/ICMLA.2019.00035.
40. G. Mauro *et al.*, "Context-adaptable radar-based people counting via few-shot learning," *Appl Intell*, vol. 53, no. 21, pp. 25359–25387, Nov. 2023, doi: 10.1007/s10489-023-04778-z.
41. J. Wu, H. Cui, and N. Dahnoun, "A Novel High Performance Human detection, Tracking and Alarm System Based on millimeter-wave Radar," in *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro: IEEE, Jun. 2021, pp. 1–4. doi: 10.1109/MECO52532.2021.9460150.
42. H. Cui and N. Dahnoun, "High Precision Human Detection and Tracking Using Millimeter-Wave Radars," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 36, no. 1, pp. 22–32, Jan. 2021, doi: 10.1109/MAES.2020.3021322.
43. C. Will, P. Vaishnav, A. Chakraborty, and A. Santra, "Human Target Detection, Tracking, and Classification Using 24-GHz FMCW Radar," *IEEE Sensors J.*, vol. 19, no. 17, pp. 7283–7299, Sep. 2019, doi: 10.1109/JSEN.2019.2914365.
44. J. Pegoraro, F. Meneghello, and M. Rossi, "Multiperson Continuous Tracking and Identification From mm-Wave Micro-Doppler Signatures," *IEEE Trans. Geosci. Remote Sensing*, vol. 59, no. 4, pp. 2994–3009, Apr. 2021, doi: 10.1109/TGRS.2020.3019915.
45. P. Zhao *et al.*, "mID: Tracking and Identifying People with Millimeter Wave Radar," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Santorini Island, Greece: IEEE, May 2019, pp. 33–40. doi: 10.1109/DCOSS.2019.00028.
46. P. Zhao *et al.*, "Human tracking and identification through a millimeter wave radar," *Ad Hoc Networks*, vol. 116, p. 102475, May 2021, doi: 10.1016/j.adhoc.2021.102475.
47. J. Pegoraro and M. Rossi, "Real-Time People Tracking and Identification From Sparse mm-Wave Radar Point-Clouds," *IEEE Access*, vol. 9, pp. 78504–78520, 2021, doi: 10.1109/ACCESS.2021.3083980.
48. A. Ninos, J. Hasch, M. Heizmann, and T. Zwick, "Radar-Based Robust People Tracking and Consumer Applications," *IEEE Sensors J.*, vol. 22, no. 4, pp. 3726–3735, Feb. 2022, doi: 10.1109/JSEN.2022.3141202.
49. C. Wu, F. Zhang, B. Wang, and K. J. Ray Liu, "mmTrack: Passive Multi-Person Localization Using Commodity Millimeter Wave Radio," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Toronto, ON, Canada: IEEE, Jul. 2020, pp. 2400–2409. doi: 10.1109/INFOCOM41043.2020.9155293.
50. N. Knudde *et al.*, "Indoor tracking of multiple persons with a 77 GHz MIMO FMCW radar," in *2017 European Radar Conference (EURAD)*, Nuremberg: IEEE, Oct. 2017, pp. 61–64. doi: 10.23919/EURAD.2017.8249147.

51. C. Y. Aydogdu, S. Hazra, A. Santra, and R. Weigel, "Multi-Modal Cross Learning for Improved People Counting using Short-Range FMCW Radar," in *2020 IEEE International Radar Conference (RADAR)*, Washington, DC, USA: IEEE, Apr. 2020, pp. 250–255. doi: 10.1109/RADAR42522.2020.9114871.
52. M. Stephan, S. Hazra, A. Santra, R. Weigel, and G. Fischer, "People Counting Solution Using an FMCW Radar with Knowledge Distillation From Camera Data," in *2021 IEEE Sensors*, Sydney, Australia: IEEE, Oct. 2021, pp. 1–4. doi: 10.1109/SENSORS47087.2021.9639798.
53. M. Dimitrievski, L. Jacobs, P. Veelaert, and W. Philips, "People Tracking by Cooperative Fusion of RADAR and Camera Sensors," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand: IEEE, Oct. 2019, pp. 509–514. doi: 10.1109/ITSC.2019.8917238.
54. N. S. Zewge, Y. Kim, J. Kim, and J.-H. Kim, "Millimeter-Wave Radar and RGB-D Camera Sensor Fusion for Real-Time People Detection and Tracking," in *2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA)*, Daejeon, Korea (South): IEEE, Nov. 2019, pp. 93–98. doi: 10.1109/RITAPP.2019.8932892.
55. S. M. A. Zeinolabedin, F. M. Schüffny, R. George, F. Kelber, H. Bauer, S. Scholze, S. Hänzsche, M. Stolba, A. Dixius, G. Ellguth, D. Walter, S. Höppner, and C. Mayr, "A 16-channel fully configurable neural SoC with 1.52 μ W/Ch signal acquisition, 2.79 μ W/Ch real-time spike classifier, and 1.79 TOPS/W deep neural network accelerator in 22 nm FDSOI," *IEEE Transactions on Biomedical Circuits and Systems*, pp. 1–1, 2022.
56. Höppner, S., Yan, Y., Dixius, A., Scholze, S., Partzsch, J., Stolba, M., ... & Mayr, C. (2021). The SpiNNaker 2 processing element architecture for hybrid digital neuromorphic computing. arXiv preprint arXiv:2103.08392. <https://doi.org/10.48550/arXiv.2103.08392>