Schlussbericht

zum Teilvorhaben der Robert Bosch GmbH

Hardwareunterstütztes Machine Learning für hochautomatisiertes Fahren

im Verbundprojekt PARIS:
Parallele Implementierungs-Strategien
für das hochautomatisierte Fahren

FKZ: 16ES0610

Laufzeit des Vorhabens: 01.04.2017 bis 30.04.2020

Autoren:
Christoph Schorn, Lydia Gauerhof, Christoph Kunze, Marc Luther,
Armin Runge, Sebastian Vogel

Robert Bosch GmbH
Zentralbereich Forschung und Vorausentwicklung
Renningen und Hildesheim

E-Mail: Christoph.Schorn@de.bosch.com

29. Oktober 2020

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Inhaltsverzeichnis

1 Kurzfassung

Die benötigte Rechenleistung für neuartige Algorithmen, insbesondere tiefe neuronale Netze, die vor allem im Rahmen der Perzeption von automatisierten Fahrzeugen eingesetzt werden, ist besonders hoch. Da die neuartigen Algorithmen klassische Verfahren deutlich in der Performanz übertreffen, sind sie beim automatisierten Fahren unentbehrlich.

Vor diesem Hintergrund wurde die Entwicklung eines dedizierten Hardware-Beschleunigers als IP-Core adressiert. Dabei wurden neue Detektionstechnologien für die Umfelderkennung in Form eines neuronalen Netzes für die Fußgängererkennung evaluiert und in effizienter Weise auf den entwickelten Hardware-Beschleuniger abgebildet.

Darüber hinaus wurden Methoden zur Komprimierung und Quantisierung von hochperformanten neuronalen Netzen für die Umfelderkennung entwickelt und evaluiert, welche deren Echtzeitfähigkeit und Energieverbrauch bei der Ausführung auf dedizierten Hardware-Beschleunigern verbessern. Ansatz und Ergebnisse wurden auf einer wissenschaftlichen Konferenz präsentiert und diskutiert [VSGA19].

Zudem wurden neuartige Verfahren für die Absicherung von neuronalen Netzen auf algorithmischer Ebene und auf Hardware-Ebene entwickelt und veröffentlicht [SGA18b], welche die Robustheit und Fehlertoleranz der Umfelderkennung bei geringem Mehraufwand hinsichtlich der Rechenoperationen erhöhen.

Schließlich wurden neue Anforderungen und Vorgehensweisen für die Validierung der Performanz von Machine Learning-basierten Funktionen im hochautomatisierten Fahren entwickelt und auf wissenschaftlichen Konferenzen veröffentlicht [GMB18, BGS+19].

Alles in allem konnten wichtige Ergebnisse zur Absicherbarkeit von neuronalen Netzen und der hierfür eingesetzte Hardware beim automatisierten Fahren unter Berücksichtigung der Rechenleistung erzielt werden. Die Validierung fand in Zusammenarbeit mit Konsortialpartnern in einem Versuchsfahrzeug statt.

2 Ausgangslage, Zielsetzung und Ablauf

2.1 Motivation des Verbundprojekts

Die kontinuierliche Weiterentwicklung der Mikroelektronik und insbesondere der Umfeldsensoren zur Situationserfassung, bietet die Fahrzeugautomatisierung ein großes Potential zur Steigerung der Verkehrssicherheit, gleichzeitig aber auch der Verkehrseffizienz und des Fahrkomforts. Das vollautomatisierte Fahren verlangt eine signifikante Weiterentwicklung der Elektroniksysteme im Fahrzeug. Energieeffiziente Sensordatenfusion und Echtzeitfähigkeit sind die beiden wesentlichen Anforderungen der Sensordatenverarbeitung für vollautomatisierte Elektrofahrzeuge im sich schnell veränderlichen urbanen Umfeld. Das System muss in der Lage sein, unbekannte Verkehrssituationen schnell (in Echtzeit) zu analysieren und sichere sowie energieeffiziente Entscheidungen treffen zu können. Aufgrund der Unvorhersehbarkeit der Ereignisse im Straßenverkehr muss das System in Zukunft anpassbar und lernfähig wie ein menschlicher Fahrer sein, der durch seine Erfahrungen lernt, das Fahrzeug sicher zu steuern und bestimmte Situationen frühzeitig zu erkennen. Neben dem Aspekt der Sicherheit kann vorausschauendes Fahren auch die Effizienz und somit die Reichweite von elektrisch betriebenen Fahrzeugen erhöhen. Ein Fahrzeug kann beispielsweise frühzeitig die Geschwindigkeit reduzieren, wenn eine Ampel auf rot schaltet. Insbesondere Machine-Learning-Fähigkeiten müssen hierfür in das Sensorik-System integriert werden, damit das System aus zuvor gemachten Erfahrungen lernen kann.

Das Machine-Learning umfasst hier mehrere Stufen: In einem ersten Schritt kann der Fahrer als Vorbild dienen und bestimmte Situationen im Verkehr bewerten. Das System kann von dessen Erfahrungen lernen und sich verbessern. In einem nächsten Schritt kann das System selbst erkennen, ob eine getroffene Entscheidung richtig war. Neben der Sicherheit und Effizienz ist auch die Akzeptanz ein wichtiger Aspekt. Insbesondere in Deutschland ist die Wahrung der Privatsphäre essentiell, damit eine Technologie akzeptiert wird. Cloud-basiertes Lernen, das

viele Information über einen Fahrer im Netz zusammenführt, ist daher nicht geeignet. Des Weiteren ist nicht flächendeckend eine Funkverbindung verfügbar und aufgrund des „Big-Data-Problems“ ist es nicht sinnvoll alle Rohdaten ohne entsprechende Vorverarbeitung in die Cloud zu übertragen. Aus diesem Grund müssen Machine-Learning-Algorithmen, nach einer initialen Offline Anlern-Phase, „lokal“ im Fahrzeug hinzu lernen. Diese Anforderung stellt eine wesentliche wissenschaftliche Herausforderung dar, denn insbesondere das Lernen ist im Moment noch mit einer hohen Komplexität verbunden. Machine-Learning Algorithmen werden daher mit hochperformanten Rechen-Clustern angelernt. Die Entwicklung einer spezialisierten und flexiblen Hardware-Architektur mit den dazugehörigen Software-Konzepten ist für die Erreichung des hochautomatisierten Fahrens unerlässlich.

2.2 Motivation und Ziele des Teilvorhabens

Motivation sind Systeme im Straßenverkehr, die in Zukunft ein menschliches Maß an Überblick und Einschätzungsvermögen erreichen, um das Fahrzeug sicher zu steuern und kritische Situationen frühzeitig zu erkennen.

Die Kernziele des Teilvorhabens sind zum Einen die Entwicklung einer flexiblen und an Machine-Learning, Sensor-Signalverarbeitung und -Datenfusion angepassten Hardwareplattform für vollautomatisierte Fahrzeuge und zum Anderen die Bewertung und Auswahl geeigneter Algorithmen. Die Zielanwendung der Hardwareplattformen sind in erster Linie Algorithmen zur intelligenten Umfelderfassung des Fahrzeuges. Hardware-Beschleuniger, die unter anderem die Objekterkennung echtzeitfähig realisieren, werden unter den Gesichtspunkten der Safety entwickelt. Des Weiteren zielt Bosch auf die Entwicklung von flexiblen Architekturen ab, die Machine-Learning-Algorithmen auf Embedded-Systemen effizient ausführen. Und nicht zuletzt sind die Integration der Ergebnisse in das Gesamtsystem sowie der Austausch und die Zusammenarbeit mit den Projektpartnern wichtige Ziele.

2.3 Ausgangslage und Aufgabenstellung

Die stetige Weiterentwicklung der Sicherheit im Straßenverkehr, des automobilen Fahrkomforts und der Steigerung der Verkehrseffizienz wird von der Robert Bosch GmbH permanent getrieben. Dabei spielen immer mehr Teilbereiche in Lösungsansätze hinein. Während in der Vergangenheit vielfach mit mechanischen Mitteln unter Einbezug von elektronischen Systemen gearbeitet wurde, liegt der Fokus mittlerweile auf hochkomplexen Rechensystemen zur Weiterentwicklung des Automobils hin zu einem autonom agierenden Verkehrsmittel. Durch die Verbesserung der Umfeldsensoren und die Entwicklung insbesondere leistungsfähiger Elektroniksysteme zur Situationserfassung kommen wir dem Ziel eines sicheren, effizienten und komfortablen Fahrzeuges einige Schritte näher.

Die Systeme müssen in der Lage sein, unbekannte Verkehrssituationen schnell (in Echtzeit) zu analysieren und sichere sowie energieeffiziente Entscheidungen zu treffen. Aufgrund der unübersichtlichen und komplexen Szenarien im Straßenverkehr müssen Systeme in Zukunft ein menschliches Maß an Überblick und Einschätzungsvermögen erreichen, um das Fahrzeug sicher zu steuern und kritische Situationen frühzeitig zu erkennen. Dazu liefern Machine Learning Algorithmen vielversprechende Ergebnisse, welche insbesondere hinsichtlich der Integrierbarkeit in Fahrzeuge und in die Infrastruktur im Fokus dieses Projektes liegen. Während in der Presse bereits von ersten hoch automatisierten Fahrzeugen berichtet wird, gibt es noch großen Bedarf durch aktuelle Forschungen und Entwicklungen, zum einen die Systeme sicherer zu machen, zum anderen die rechenintensiven Algorithmen von kofferraumgroßen Computersystemen auf hoch integrierte, Performance-starke Embedded-Hardware zu bringen.

Zusammenfassen lassen sich die Anforderungen an zukünftige, auf Machine Learning basierende Systeme durch:

- Die Systeme müssen gleichzeitig energieeffizient und leistungsfähig sein, was nur durch spezialisierte Hardware erreicht werden kann. Sowohl die verfügbare Energie (insbesondere bei E-Fahrzeugen), als auch die Wärmeabgabe des Chips stellen hierbei Limits dar.
- Die Systeme müssen flexibel sein: Da sich Produktzyklen immer mehr verkürzen, kann nicht für jedes neue Produkt eine vollkommen neue Hardware-Architektur entwickelt werden.

- Der Entwicklungsaufwand muss reduziert werden: Kurze Produktzyklen erfordern Software-Konzepte, die eine schnelle Umsetzung neuer Anwendungen auf eine komplexe Hardware-Architektur erlauben. Dies umfasst auch die Simulation und die Validierung der Technologie.

2.4 Planung und Ablauf

Das Konsortium des Verbundprojektes PARIS (Parallele Implementierungs-Strategien für das hochautomatisierte Fahren) bestand aus folgenden Partnern:

- Institut für Mikroelektronische Systeme der Universität Hannover (IMS) (Projektkoordinator)
- Lehrstuhl Integrierte Systeme der Signalverarbeitung der RWTH Aachen (ISS)
- Lehrstuhl Software für Systeme auf Silizium der RWTH Aachen (SSS)
- Institut für Kraftfahrzeuge der RWTH Aachen (IKA)
- Computer Vision Group der RWTH Aachen (CVG)
- Lehrstuhl Adaptive Dynamische Systeme der Technischen Universität Dresden (TUD)
- Baselabs GmbH
- Robert Bosch GmbH
- NISYS GmbH
- Elektrobit Automotive GmbH
- Videantis GmbH
- Silexica GmbH

Das Gesamtprojekt war in neun Arbeitspakete (AP0 bis AP8) aufgeteilt, deren Interaktionen in Abbildung 1 dargestellt sind. Die Beiträge der Robert Bosch GmbH sind in die Arbeitspakete AP1 bis AP4 und AP6 bis AP8 eingeflossen. Nachfolgend werden die von Bosch zu erbringenden Arbeitsinhalte innerhalb der Arbeitspakete kurz zusammengefasst. Die Beschreibung wurde der zu Beginn des Projektes erstellten Teilvorhabenbeschreibung entnommen.

AP1: Anforderungserhebung und Szenario-Definition: Zusammen mit den Projektpartnern wird Bosch in diesem Arbeitspaket Anforderungen für die Algorithmen und die Zielplattform definieren. Darüber hinaus werden zwei oder drei relevante Use-Cases für eine reale Validierung in der Teststadt (CERMcity) im Aldenhoven Testing Center der RWTH Aachen konzipiert.

AP2: Systemkonzeptionierung: Das Zusammenspiel von Multiprocessor System-on-Chip (MPSoC) Plattform, Software und externer Sensorik ist der zentrale Aspekt dieses Arbeitspakets. Ein Beitrag von Bosch wird hierbei die Untersuchung der relevanten Algorithmen für Computer Vision Anwendungen sein. Der zweite Beitrag von Bosch in diesem Arbeitspaket beinhaltet die Bewertung unterschiedlicher MPSoC-Lösungen im Hinblick auf die zuvor definierten Anforderungen. In die konzeptionellen Überlegungen werden auch die notwendigen Sensoren und Sensorschnittstellen, die das Sensordatenfusionskonzept ermöglichen sollen, mit einbezogen.

AP3: Hardware-unterstütztes Machine Learning: Dieses Arbeitspaket beschäftigt sich mit der Entwicklung neuartiger Machine Learning Ansätze (Next Generation Technology), insbesondere von Deep Learning Techniken mit Neuronalen Netzen. Bosch übernimmt in diesem Arbeitspaket die Aufgabe der Auswahl von Next Generation Alternativen für aktuell verwendete Computer Vision Algorithmen insbesondere für die semantische Szenensegmentierung. Neben einer Verbesserung der Klassifikationsperformanz steht auch die Optimierung für die Umsetzung auf Embedded-HW-Plattformen im Vordergrund. Zudem sollen die Anforderungen hinsichtlich der funktionalen Sicherheit in die Überlegungen mit einbezogen werden.

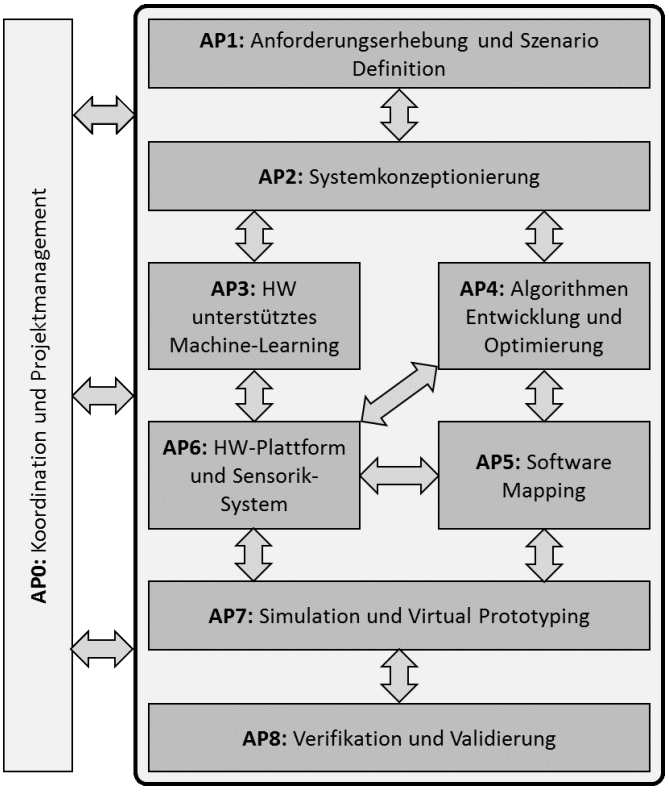

Abbildung 1: Arbeitspakete des gesamten Verbundprojektes PARIS und deren Interaktion.

AP4: Algorithmen-Entwicklung und Optimierung: Die verschiedenen Algorithmen für die Erfassung aller Zustände des Verkehrs, die hochautomatisiertes Fahren ermöglichen können, werden in diesem Arbeitspaket entwickelt. Bosch wird hierbei, gemeinsam mit den Projektpartnern, die im Fahrzeug benötigten Algorithmen betrachten. Dabei liegt der Fokus auf der Auslegung, Weiterentwicklung und Optimierung bestehender Echtzeit-Algorithmen für Embedded Hardware-Plattformen. Ausgehend von ausgewählten Algorithmen, die für eine Hardware-Implementierung in Frage kommen, wird eine Echtzeit-Referenzimplementierung realisiert, die als Basis für weitere Optimierungen und für das anschließende Hardware-Mapping dient. Die implementierten Algorithmen werden anhand von verschiedenen Anwendungen getestet.

AP6: Hardware-Plattform und Sensorik-System: Dieses Arbeitspaket widmet sich der Entwicklung neuer anwendungsspezifischer Prozessor-Cores (z.B. ASICs) für ein zukünftiges MPSoC, welches in der Lage ist, alle Algorithmen-Partitionen in Echtzeit und mit hoher Energieeffizienz auszuführen. Bosch wird im Rahmen dieses Arbeitspaketes verschiedene neue IP-Cores für die in AP3 und AP4 entwickelten Algorithmen entwickeln. Es handelt sich hierbei explizit um die Entwicklung neuer Hardware-Architekturen, welche an die entsprechenden Aufgaben angepasst sind.

AP7: Simulation und Virtual-Prototyping: Hauptziel von AP7 ist die Evaluation des in AP6 definierten Gesamtsystems als virtueller Prototyp. Dieser dient dazu, die neue Zielhardware, die Algorithmen und auch das Gesamtsystem zu simulieren, zu validieren und zu debuggen. Bosch wird zunächst die Integration der in AP6 entwickelten IP-Cores in die Simulationsumgebung unterstützen. Anschließend werden von Bosch mithilfe des virtuellen Prototyps Evaluationsmessungen durchgeführt, die sowohl auf die Bewertung des Systems hinsichtlich der funktionalen Sicherheit, als auch auf eine Performanz-Analyse der IP-Cores im Zusammenspiel mit den anderen Komponenten des Systems abzielen.

AP8: Verifikation und Validierung: Zur Validierung der im Projekt umgesetzten Komponenten werden Demonstratoren aufgebaut, welche alle Ebenen der komplexen Systeme umfassen. Die in AP6 entwickelten IP-Cores sollen zunächst unter Laborbedingungen mittels einer FPGA-Implementierung validiert werden. Bosch wird

die Projektpartner bei der Einbindung der IP-Cores in ein eingebettetes Computer-Vision-System auf Basis eines für Videoverarbeitung geeigneten Prototypenboards unterstützen. Hierzu gehört auch die Auswahl und Bereitstellung geeigneter Hardwareschnittstellen für den Kameraanschluss und die physikalische Anbindung. In einem nächsten Schritt soll die Referenz-Plattform in ein Fahrzeug gebracht und in der Versuchsstadt auf der Forschungskreuzung der CERMcity-Initiative getestet werden. Bosch wird Unterstützung bei der Integration des FPGA-basierten Demonstrators in das Versuchsfahrzeug leisten.

Innerhalb des Verbundprojektes hat Bosch zudem die Leitung des AP6 (Hardware-Plattform und Sensorik-System) übernommen und die Beiträge der Konsortialpartner in diesem Arbeitspaket koordiniert.

2.5 Wissenschaftlich-technischer Stand zu Beginn des Vorhabens

Nachfolgend wird der Stand der Technik beschrieben, wie er sich zum Zeitpunkt der Projektantragsphase (2016) darstellte.

Als Bestandteile von Automobilen existieren bereits heute Fahrerassistenzsysteme, welche teilautomatisiertes Fahren erlauben. Für die Umsetzung solcher Systeme wird in der Regel zu bereits vorhandener Hardware gegriffen, wie beispielsweise Radar- und Kamerasysteme, sowie Grafikkarten (GPUs) und programmierbare logische Schaltungen (FPGAs). Im Bereich der Hardware-Plattformen werden programmierbare bzw. rekonfigurierbare Hardware-Lösungen gegenüber anwendungs-spezifischen, integrierten Schaltkreisen (ASICs) aufgrund des günstigeren Kosten- und Entwicklungsaufwandes bevorzugt. Chip-Hersteller wie z.B. NVIDIA bieten inzwischen erste einschlägige Produkte an [NVI15, Xil14]. Diese „off-the-shelf"-Bausteine sind allerdings nicht optimal an die Bedürfnisse des autonom fahrenden Autos von morgen angepasst. Diese Hardware-Plattformen sind entweder nicht energieeffizient genug (z.B. GPUs), oder sie sind für relevante Anwendungen in Machine Learning (z.B. für Sensordatenfusion, Objektdetektion und -erkennung) nur eingeschränkt einsetzbar. Im Bereich der Kamerasysteme konnten in den letzten Jahren deutliche Fortschritte bei der Leistungsfähigkeit erzielt werden. Auch in Zukunft wird hier mit einer weiteren Steigerung der Erfassungsqualität und -robustheit gerechnet. Insbesondere die Verknüpfung verschiedener Linsen mehrerer Kamerasensoren zu Stereo- oder Multifokalkameras eröffnet gegenüber herkömmlichen Monokameras neue Potentiale dieser Technologie. Es ist zu erwarten, dass die Stückkosten für Kamerasysteme in den nächsten Jahren im Vergleich zu Lidar- oder Radarsystemen deutlich stärker abnehmen. Kamerasysteme stellen somit sowohl heute als auch in Zukunft die kostengünstigsten Systeme zur Umfelderfassung dar.

Um die Sensorinformationen (z.B. Kamerabilder) in Echtzeit auszuwerten und eine geeignete Klassifizierung des Fahrzeugumfelds vorzunehmen, werden in Zukunft zunehmend – neben leistungsfähigeren und effizienteren Hardware-Plattformen – geeignete Machine Learning Algorithmen benötigt. Machine Learning ist in den letzten Jahren zu einer zentralen Komponente für viele Aufgaben im automobilen Umfeld geworden. Insbesondere die Analyse von Bild- und Videodaten kommt nicht mehr ohne Machine Learning aus. Neben klassischen Machine Learning Methoden wie Support Vector Maschinen [CV95] oder Boosting-Ansätzen [FS95], haben sich in den letzten Jahren Deep Learning Methoden basierend auf Neuronalen Netzen stark entwickelt. Dieser Wandel wurde durch bedeutende wissenschaftliche Durchbrüche bei der Konzeption von effektiven Netzwerkarchitekturen [KSH12, SZ15, SLJ+14] und Trainingsalgorithmen [SHK+14, GB10] ermöglicht. Für Computer Vision Anwendungen sind beispielsweise Convolutional Neural Networks [LBBH98] zum Standard geworden. Diese werden bereits sehr erfolgreich für Aufgaben wie Objektdetektion [GDDM14] oder semantische Szenensegmentierung [ZJRP+15, LSD15] eingesetzt. Für die Verarbeitung von Sequenzdaten kommen aktuell gehäuft Recurrent Neural Networks [PMB13] zum Einsatz, die zeitliche Abhängigkeiten lernen und über lange Zeiträume propagieren können. Auch hier gab es in den letzten Jahren wichtige Durchbrüche, insbesondere die Long Short Term Memory Methode [HS97].

2.6 Zusammenarbeit mit anderen Stellen

Für die Erreichung der Projektziele war eine kontinuierliche Zusammenarbeit mit den Projektpartnern mit Hilfe von Mailingliste, Telefonkonferenzen und persönlichen Treffen erforderlich. Ein persönlicher Austausch zwischen den Projektpartnern und Bosch hat bei den veranstalteten Projekttreffen, die von verschiedenen Projektpartnern organisiert wurden, stattgefunden. Bei allen Treffen hat Bosch mit mehreren Mitarbeitern teilgenommen.

Außerhalb der regelmäßigen Projekttreffen gab es weitere persönliche Treffen mit einzelnen Projektpartnern zum wissenschaftlichen Austausch. Hierzu hat ein Innovationsworkshop mit Bosch und dem Lehrstuhl für Integrierte Systeme der Signalverarbeitung (ISS) der RWTH Aachen stattgefunden. Es gab einen telefonischen Austausch bzgl. der Ergebnisse zur Fußgängererkennung mit NISYS. Außerdem gab es ein Arbeitstreffen im Rahmen des AP8 (Integrationstest der MPSoC Referenzplattform in das Versuchsfahrzeug) in Aachen zwischen dem Institut für Kraftfahrzeuge (IKA) der RWTH Aachen und Bosch.

3 Arbeiten und Ergebnisse des Teilvorhabens

3.1 Anforderungserhebung und Szenario-Definition

Anforderungen an die Rechnerplattformen (HW-Plattform, Sensoren, Schnittstellen, etc.) und an die Funktionen zum Szenenverständnis wurden unter Berücksichtigung der gültigen Regularien und bereits vorhandener Einbauten im Versuchsfahrzeug des IKA abgeleitet und den Projektpartnern zur Verfügung gestellt.

Zusammen mit den Projektpartnern wurden zwei Szenarien definiert, die im weiteren Projektverlauf demonstriert werden sollen:

- Urbane Kreuzung mit kreuzenden Fußgängern (Szenario 1)
- Überholen eines vorausfahrenden Fahrzeugs (Szenario 2)

Für die selektierten Szenarien wurden die notwendigen Aktionen, der zeitliche Ablauf und die existierenden Abhängigkeiten identifiziert und den Projektpartnern zur Verfügung gestellt.

Für Szenario 1 wurden in Kooperation mit dem Institut für Kraftfahrzeuge Aachen (IKA) erste Testsequenzen auf dem Aldenhoven Testing Center eingefahren und den Projektpartnern zur Verfügung gestellt. Anhand dieser Testsequenzen wurden bei Bosch sowie bei der Computer Vision Group (CVG) der RWTH Aachen bereits existierende Algorithmen untersucht.

3.2 Systemkonzeptionierung

Machine Learning basierte Algorithmen wurden identifiziert und mit Hinblick auf den Einsatz in eingebetteten System bewertet. Bosch konzentriert sich im Projektverlauf unter anderem auf die videobasierte Fußgängererkennung (PDet) mittels Convolutional Neural Networks (CNNs), da die geplante semantische Segmentierung von dem Projektpartner CVG bereits bearbeitet wird und für das Szenario 1 eine Fußgängererkennung erforderlich ist.

Die Xilinx Zynq Ultrascale+ Plattform wurde als MPSoC Referenzplattform vorgeschlagen. Das Konsortium hat sich auf die vorgeschlagene Plattform geeinigt und Bosch hat das entsprechende Entwicklungs-Kit für das Projekt angeschafft.

Um eine effiziente Anbindung der definierten Sensoren zu ermöglichen, schlägt Bosch vor, sich an der im Versuchsfahrzeug existierenden Lösung auf Basis eines Car-PCs zu orientieren. Dieser Rechner soll im Sinne des Projektauftrags nur zur Ansteuerung der Sensoren und zur Verteilung der Sensordaten an die definierten MPSoC Plattformen eingesetzt werden.

Als Middleware zur Kommunikation zwischen den Rechnerplattformen schlägt Bosch den Einsatz des Robot Operating System (ROS) vor. Dieser Vorschlag wird vom Konsortium akzeptiert.

Im Konsortium wurde festgestellt, dass die Anbindung der vorhandenen IBEO-Sensoren mit einem hohen Entwicklungsrisiko verbunden ist. Bosch und das IKA haben dem Vorschlag des IMS zugestimmt, als alternativen 3D Sensor ein Stereokamerasystem im Versuchsträger zu verbauen. Bosch unterstützt das IKA bei der Ausstattung des Versuchsträgers durch die Konstruktion des benötigten Stereokamerahalters und der fahrzeugspezifischen Anpassung des Kamerahalters. Die erforderliche zweite Kamera wird dem Projekt von Bosch für den Projektzeitraum zur Verfügung gestellt.

3.3 Methodenentwicklung für Hardware-unterstütztes Machine Learning

Für Bosch sind die Entwicklungen zukünftiger Advanced Driver Assistance Systems (ADAS) von zentraler Bedeutung, wenn es darum geht, mehr Sicherheit im Straßenverkehr zu erreichen. Im Rahmen dieses Projektes fand zunächst ein Austausch mit den für zukünftige Algorithmen zuständigen Spezialisten statt. Es wurden verfügbare Referenzimplementierungen von CNN-basierten Algorithmen recherchiert und bewertet. Hierzu gehören folgende Ansätze:

- Feature Extractors: SqueezeNet [IHM+16], VGG [SZ15], MobileNet [SHZ+18], ShuffleNet [ZZLS18], DenseNet [HLMW17]
- Object Detection: RPN [RHGS15], SSD [LAE+16]
- Action Recognition: LSTM [HS97], GRU [CMBB14]
- Semantic Segmentation: FCN-8s [LSD15]

Basierend auf dem Austausch mit den Experten und den Anforderungen aus den definierten Szenarien (siehe Abschnitt 3.1), wurden für die Aufgabe der videobasierten Fußgängererkennung (PDet) das SqueezeNet + RPN (Region Proposal Network) ausgewählt (siehe auch Abschnitt 3.4). Eine erste Version des CNNs wurde von Bosch implementiert und evaluiert. Anschließend wurde das CNN den Projektpartnern zur Verfügung gestellt.

3.3.1 Identifikation geeigneter Trainings- und Testdatensätze

Gemeinsam mit dem Projektpartner Computer Vision Group at RWTH Aachen University (CVG) wurden geeignete Trainings- und Testdatensätze identifiziert. Um den Austausch der entwickelten Algorithmen zwischen den Projektpartner zu erleichtern, haben sich die Projektpartner auf Keras mit TensorFlow Backend als Framework geeinigt. Für das Training eines Semantic Segmentation oder PDet-Netzes wurden Cityscapes [COR+16] und Citypersons [ZBS17] als geeignete Datensätze identifiziert. Weitere Datensätze wie Open Images [KRA+20], Mapillary Vistas [NORBK17], BDD100K [YCW+20] und der KITTI [GLSU13] Datensatz können als Ergänzung oder Alternative verwendet werden:

Cityscapes: Cityscapes besteht aus einem großen, vielfältigen Satz von Stereo-Videosequenzen, die in Straßen in 50 verschiedenen Städten aufgenommen wurden. [COR+16]

CityPersons: CityPersons ist ein neuer Datensatz von Personenannotationen basierend auf Cityscapes. Die Vielfalt von CityPersonen ermöglicht es, ein einziges CNN-Modell zu trainieren, das sich gut über mehrere Benchmarks verallgemeinert. [ZBS17]

Open Images: Open Images ist ein Datensatz von ≈9 Millionen Bildern, die mit Labels und Objektrahmen versehen sind. Der Trainingssatz enthält 14,6M Bounding Boxen für 600 Objektklassen auf 1,74M Bildern und ist damit der größte vorhandene Datensatz mit Objektlokalisierungsannotationen. [KRA+20]

Mapillary Vistas: Der Mapillary Vistas Datensatz besteht aus Bilder von Straßenszenen aus vielen Teile der Welt. Ein Teil dieser Bilder wurde mit semantischen Segmentierungen annotiert. [NORBK17]

BDD100K: Der Berkly Deep Drive Datensatz besteht aus 100k Videosequenzen aufgenommen bei fahrten in verschiedenen Amerikanischen Städten. Teile des Datensatzes sind für verschiedene Aufgaben, wie z.B. semantische Segmentierung oder Spurerkennung annotiert. [YCW+20]

KITTI: Der KITTI Datensatz wurde bei Fahrten durch die Stadt Karlsruhe, in ländlichen Gebieten und auf Autobahnen erfasst. Bis zu 15 Autos und 30 Fußgänger sind pro Bild sichtbar. [GLSU13]

3.3.2 Definition von Evaluationskriterien

Generell zielen Objektdetektor-Bewertungsmaßnahmen darauf ab, das Verhältnis von true/false Positives und false Negatives bei unterschiedlichen Detektions-Schwellenwerten (z.B. Detektorarbeitspunkt) zusammenzufassen. Zu diesem Zweck wird ein Kriterium definiert, das erfüllt sein muss, um eine Bounding Box als true Positive Detektion zu betrachten. Die Standard-Messung basiert auf der Messung der Intersection over Union (IoU) zwischen jedem Paar von Ground Truth Box und erkannter Bounding Box, wobei ihr Intersection-Bereich durch ihren Union-Bereich geteilt wird. Wenn es ein Paar gibt, für das IoU $\geq 0,5$ ist, werden die Ground Truth Box und die Bounding Box als true Positive betrachtet. Wenn für eine Ground Truth Box keine solche Bounding Box gefunden wird, wird sie als false Negative gewertet, und wenn für eine Bounding Box kein Ground Truth Box vorhanden ist, wird sie als false Positive angesehen. Bei mehreren Bounding Boxen, die mit nur einer Ground Truth Box ausreichend überlappen, wird nur eine von ihnen als true Positive gezählt und die anderen als false Positives.

Basierend auf diesen Definitionen wird für jeden Detektorarbeitspunkt die Anzahl der erfassten Ground Truth Boxen und die Anzahl der false Positives berechnet. Durch Division der Anzahl der erfassten Ground Truth Boxen durch die Anzahl aller Ground Truth Boxen erhält man den sogenannten Recall. Dann wird die Fehlquote als $1 -$ Recall definiert. Durch die Verwendung von Paaren der false Positives pro Bild und der entsprechenden Missrate erhält man eine Kurve. Durch die Abtastung dieser Kurve auf logarithmisch verteilten false Positive Raten zwischen 10^{-2} und 10^{0} und die Mittelung der entsprechenden Missraten erhält man die sogenannte logarithmische durchschnittliche Missrate (log average miss rate), die von Dollar et al. [DWSP11] vorgeschlagen wurde und seitdem in vielen Publikationen zur Fußgängererkennung verwendet wird.

3.3.3 Komprimierung Neuronaler Netze

Um eine Umsetzung der entwickelten PDet-Algorithmen auf Hardware-Beschleunigern zu gewährleisten, haben wir zwei geeignete Kompressionsmethoden für die CNN-Algorithmen identifiziert:

- Filter Pruning komprimiert das Netz unabhängig von der Hardware Architektur.
- Weight Sharing kann mit geringem Hardware-Overhead eine Kompressionsrate von bis zu 85% erzeugen.

Beim Filter Pruning werden nicht nur Netzparameter reduziert, sondern auch Input und Output Feature Maps. Dabei entsteht ein Zielkonflikt zwischen dem Performance-Verlust und der Kompressionsrate. Unsere Untersuchung des RPN+VGG Netzes [RHGS15] zeigte, dass 16% der Filter mit 10% Verlust in der log average miss rate geprunt werden kann. Dadurch werden die Parameter zu 28% reduziert und 22% der Multiply-Accumulate (MAC) Operationen gespart.

Weight sharing ist eine Art von nicht-linearer Quantisierung, in der die Gewichte in Gruppen eingeteilt sind. Dabei wird für jede Gruppe ein Centroid bestimmt, so dass der durchschnittliche Abstand zwischen jedem Gewicht und dem Centroid minimiert wird. Danach werden die Centroids in einer Look-up Tabelle (LUT) gespeichert und die Gewichte mit dem Index der relevante Centroid in der LUT codiert. Weight sharing kann ein RPN+SqueezeNet [WIJK17] mit 7 Bit ohne Verlust in der log average miss rate quantisieren und dadurch eine Kompressionsrate von 78% gegenüber einer Float32-Datenrepräsentation erreichen.

3.3.4 Beurteilung hinsichtlich der funktionalen Sicherheit

Bei der Bewertung der funktionalen Sicherheit mit Blick auf die ISO 26262 haben wir uns auf den Aspekt der zufälligen Hardwarefehler, und hierbei insbesondere Bit-Flip Fehler in den Aktivierungen und Parametern der CNNs beschränkt. Wir haben dabei die folgenden Einflussfaktoren genauer untersucht:

Datenrepräsentation von Aktivierungen und Parametern: Das von Vogel et al. [VSGA19] eingeführte Verfahren für die schichtweise optimierte lineare Quantisierung vortrainierter Netze ist sowohl für die nominale Performanz als auch die Resilienz gegenüber Bit-Flip Fehlern von Vorteil [SEV+20]. Unsere Experimente legen nahe, dass die Bitbreite der Datenwerte dabei so weit verringert werden sollte, wie es ohne signifikante Verschlechterung der Klassifikationsgenauigkeit des CNNs möglich ist.

Neuronen pro Schicht: Die mittlere erwartete Fehler-Resilienz einer Schicht des CNNs ist proportional zur Anzahl der Neuronen in dieser Schicht [SGA18a]. Dies sollte bei der Network Compression beachtet werden.

Resilienz-Verteilung der Features innerhalb einer Schicht: Einzelne Features einer Schicht haben eine unterschiedliche Fehler-Resilienz. Wenig resiliente Features können entweder durch Redundanz abgesichert werden, oder die Resilienz kann mit einem Regularisierungsverfahren homogenisiert werden [SGA19]. Die Absicherung durch Hinzufügen von Redundanz wurde im späteren Projektverlauf näher betrachtet (siehe Abschnitt 3.6).

3.4 Algorithmen-Entwicklung im Fahrzeug

3.4.1 Herleitung der benötigten Bildauflösung

Eine wesentliche Frage, die für die kamerabasierte Erkennung von Fußgängern eine bedeutende Rolle spielt, ist die Frage nach einer geeigneten Sensorauflösung aus zwei Gesichtspunkten: Die Auflösung sollte so groß wie für die zuverlässige Erkennung von Fußgängern nötig sein, aber aufgrund von Echtzeitgrenzen bei der Verarbeitung der Algorithmen so klein wie möglich ausfallen. Die Größe der Eingangsbildauflösung skaliert mit der Anzahl der Pixel pro Fußgänger, d.h. je größer die Auflösung, desto mehr Verarbeitungszeit wird benötigt. Die Pixel-Höhe für einen 1,80m großen Fußgänger in 30m Entfernung ist 63. Die untere Grenze der Detektions-Performance liegt bei 50px. D.h. wir können die maximale Bildauflösung der verwendeten Kamera (Allied Vision Manta G235) von 1936×1216, um ca. 20% reduzieren, auf zum Beispiel 1552×976. Außerdem ist zu beachten, dass der relevante Bildbereich in voller x-Richtung liegt und in y-Richtung oben und unten (Himmel und Motorhaube) nicht relevante Bereiche eher weggelassen werden können.

Zum Erreichen der Echtzeitfähigkeit der embedded Hardware war es nötig die Auflösung weiter auf 1472×736 zu reduzieren. Allerdings wurden hierbei die in Abschnitt 3.3.3 identifizierten Kompressionsmethoden zwar ausgewählt und evaluiert, aber nicht auf den Algorithmus in der embedded Hardware angewendet. Durch Komprimierung des neuronalen Netzes ließen sich höhere Bildauflösungen bei gleicher Verarbeitungsgeschwindigkeit erreichen.

3.4.2 Entwicklung und Optimierung eines Convolutional Neural Networks für die Fußgängererkennung

Für die Aufgabe der CNN-basierten Objektdetektion von Fußgängern wurde ein SqueezeNet + RPN (Region Proposal Network) Ansatz gewählt. Die Vorteile dieser Kombination, die auch als SqueezeDet bekannt ist, wurden von Wu et al. [WIJK17] beschrieben. SqueezeDet ist ein vollständig gefaltetes neuronales Netzwerk (FCN) zur Objekterkennung und erfüllt folgende Anforderungen:

1. Hohe Genauigkeit für hohe Sicherheit,
2. Echtzeitfähigkeit für autonomes Fahren (zeitnahe Fahrzeugkontrolle),
3. kleine Modellgröße und Energieeffizienz für optimale Nutzung auf einem Embedded-System.

Das SqueezeDet Netz nutzt nicht nur Convolutional Layers, um Feature Maps zu erzeugen, sondern auch im Output Layer (RPN bzw. ConvDet), um die Bounding Boxen und Klassen-Wahrscheinlichkeiten zu berechnen. Damit ist die Detektions-Pipeline sehr schnell, weil die Daten das Netz nur ein einziges Mal durchlaufen müssen.

Das CNN extrahiert aus dem Eingangsbild zuerst mehrere Feature Maps, welche die Präsenz und räumliche Position bestimmter komplexer Muster, die sich aus dem Training des CNNs ergeben, im aktuellen Eingangsbild wiederspiegeln. Die Feature Maps werden dann dem ConvDet Layer für die Berechnung der Bounding Boxen zugeführt. Jede berechnete Bounding Box hat eine Klassen-Wahrscheinlichkeit (in unserem Fall sind das nur zwei Klassen: Fußgänger und Hintergrund) und einen Confidence Score. Die Höhe dieser Werte bestimmt, ob die Bounding Box dem Post-Processing zugeführt wird. Dort werden die ausgewählten Bounding Boxen noch gefiltert mit der None-Maximum Suppression (NMS) Methode. Danach stehen die finalen Detektionen mit Ankerpunkt, Breite und Höhe als Ergebnisse zur Verfügung.

Die Detektionsgenauigkeit und der Recall sind die wesentlichen Werte für die Sicherheit von autonomen Fahrzeugen. Die Analyse dieser Werte anhand von durchgeführten Tests zeigt, dass das SqueezeDet Netz eine sehr hohe Genauigkeit erreicht. Tabelle 1 vergleicht SqueezeDet mit anderen Netz-Modellen bezüglich Modellgröße, Geschwindigkeit, Energieeffizienz und weiterer Aspekte. Zusammenfassend kann man sagen, dass SqueezeDet ein kleines, schnelles, energieeffizientes und genaues Netz ist, welches in all diesen Punkten state-of-the-art entspricht.

Tabelle 1: Vergleich verschiedener Parameter des SqueezeDet Modells mit Referenzmethoden (Daten aus [WIJK16])

Modell	Modell-größe (MB)	Rechen-operationen (FLOPs $\times 10^9$)	Aktivie-rungs-daten (MB)	Durchschn. GPU Leistung (W)	Geschwin-digkeit (FPS)	Energie-effizienz (J/Frame)	Mean Average Precision (%)
SqueezeDet	7,9	9,7	117,0	80,9	57,2	1,4	76,7
VGG16 +ConvDet	57,4	288,4	540,4	153,9	16,6	9,3	79,1
ResNet50 +ConvDet	35,1	61,3	369,0	95,4	22,5	4,2	76,1
Faster-RCNN +VGG16	485	–	–	200,1	1,7	117,7	–
Faster-RCNN +AlexNet	240	–	–	143,1	2,9	49,3	–
YOLO	753	–	–	187,3	25,8	7,3	–

Die Standard Bildauflösung ist in diesem Fall 1242×375 (die scale-up Variante ist H und W jeweils $\times 1.5$, und scale-down $\times 0,75$). Weitere Erläuterungen sind in [WIJK17] beschrieben. Abbildung 2 zeigt die Netzwerkarchitektur des von uns verwendeten Netzes (SqueezeNet + RPN) und deren Building Blocks im Detail.

Die wesentlichen Building Blocks sind 3×3 Conv2D, 1×1 Conv2D, MaxPooling2D und Concatenate. Bei einem Eingangsbild mit der Auflösung 2048×1024 beträgt die Anzahl der Parameter $0,75 \times 10^6$. Die Anzahl der MAC Berechnungen pro Frame ist $12,3 \times 10^9$ (22% davon sind 1×1 Conv2D und 78% sind 3×3 Conv2D). Insgesamt sind 292 MByte an Daten im 8-bit Datenformat pro Frame zu übertragen, wobei jeder Feature Map Pixel nur einmal

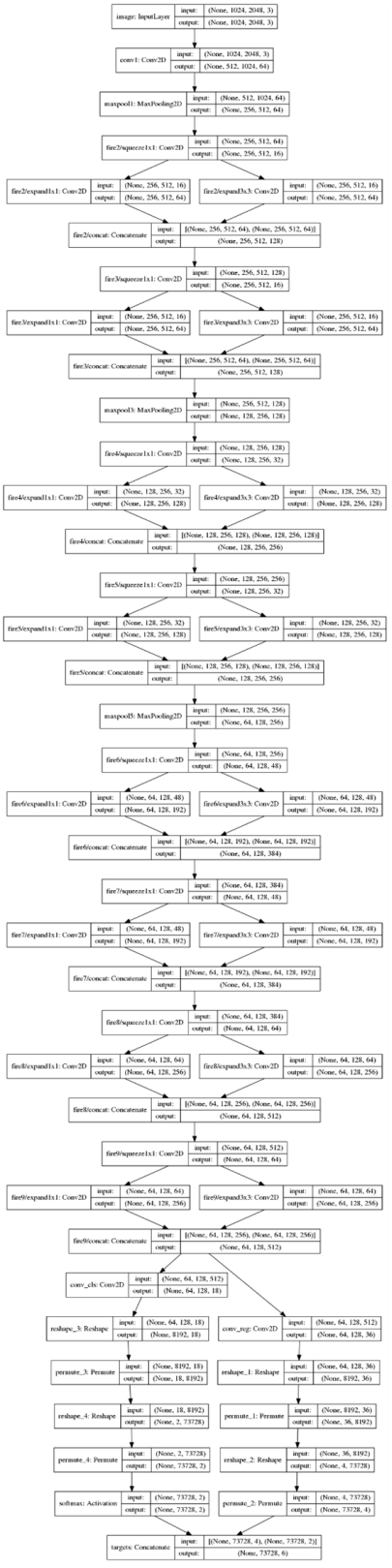

Abbildung 2: Building Blocks des verwendeten neuronalen Netzes (SqueezeNet + RPN).

gelesen/geschrieben wird und jeder Parameter nur einmal gelesen. Diese Zahlen sind natürlich abhängig von der Auflösung und werden sich noch ändern, entsprechend der noch festzulegenden Zielauflösung. Wir erwarten aber ähnlich gute Werte für die Anzahl der Berechnungen (FLOPs bzw. MACs), Verarbeitungsgeschwindigkeit (Frames pro Sekunde, FPS) und Energieeffizienz (J/frame) wie in Tabelle 1 aufgeführt.

Tabelle 2 zeigt die erste CNN Implementierung im Vergleich zu einer optimierten Version. Hierbei wurden die Building Blocks des Netzes optimiert und die Bildauflösung gemäß der Überlegungen in Abschnitt 3.4.1 reduziert.

Tabelle 2: Kennzahlen des neuronalen Netzes vor und nach der Optimierung

	Initiales Netz, volle Bildauflösung	Optimiertes Netz, reduzierte Auflösung
Parameters	$0,750 \times 10^6$	$0,748 \times 10^6$
MAC operations / frame	$12,3 \times 10^9$	$6,0 \times 10^9$
Data transfers / frame	292 MB	132 MB
Costly 3×3 pooling layers	3	0

3.5 Hardware-Plattformen und Sensorik System

3.5.1 IP-Core Entwicklung

Aufgrund der gewählten Ultrascale+ Plattform (siehe Abschnitt 3.2) wurde die Xilinx Toolchain für die Implementierung unseres IP Cores zur Beschleunigung Neuronaler Netze ausgewählt. Nach intensiver Literaturrecherche konnten zwei wesentliche und vielversprechende Hardware Architekturen identifiziert werden (im Folgenden als $\mathrm{HW_A}$ und $\mathrm{HW_B}$ bezeichnet), die sich in erster Linie durch ihre Effizienz und ihre Ressourcenanforderungen unterscheiden. Die Effizienz bezieht sich hierbei auf die durchschnittliche Performanz bei der Berechnung der zu beschleunigenden Neuronalen Netze. Die Ressourcenanforderungen beziehen sich dagegen auf die nötigen Hardwareressourcen, wie z.B. Block-RAMs oder DSPs. Da beide Architekturen als Pareto-optimale Lösungen angesehen werden können, wurden beide Architekturen in Vivado High-level Synthesis (HLS) implementiert und bewertet. Nach einigen Optimierungen konnte mit beiden Architekturen eine ähnliche Performanz erreicht werden, allerdings wurden unterschiedliche Vor- und Nachteile bei den nötigen Hardware Ressourcen identifiziert. Diese Vor- und Nachteile sind in Tabelle 3 zusammengefasst.

Tabelle 3: Vergleich der beiden Hardware-Architekturvarianten

	$\mathrm{HW_A}$	$\mathrm{HW_B}$
Latenz pro Frame bei 1024 Rechenelementen	≈200ms / Frame	≈200ms / Frame
Vorteile	Ein Block-RAM (Cluster) pro 32 Rechenelemente für Parameter.	Geringere Speicheranforderungen für Maxpooling. Kein Output Line Buffer nötig.

Anschließend an die Implementierung und Bewertung in Vivado HLS wurden sowohl $\mathrm{HW_A}$ als auch $\mathrm{HW_B}$ mit Vivado synthetisiert und auf der ausgewählten Zielplattform getestet. Darüber hinaus wurde sowohl eine bare-metal Applikation in Xilinx SDK, als auch eine PetaLinux Umgebung aufgesetzt, um von dem programmierbaren System (ARM A53 Quadcore) aus mit dem Beschleuniger interagieren zu können.

3.5.2 Test und Optimierung des CNN Hardware-Beschleunigers mit der PDet Applikation

Die Performanz der beiden Hardware Architekturen wurden im Laufe der Projektzeit inkrementell verbessert und die Hardwarearchitekturen auf die Anforderungen der Applikation abgestimmt. Der konfigurierbare HLS Code wurde mit zwei verschiedenen CNN-Architekturen (SqueezeNet und SmallVGG) synthetisiert. Somit wurde die universelle Verwendbarkeit getestet und verifiziert.

Die erste Version des Beschleunigers erzielte eine Latenz pro Frame von über einer Minute. Diese Zeit konnte mit den nachfolgenden Optimierungen zunächst auf ca. 200ms und schließlich auf ca. 70ms reduziert werden. Die initialen Optimierungsschritte umfassten:

- Optimierte Puffer für Eingangsdaten, Ausgangsdaten und Parameter inkl. vorausschauendem Laden von in der Zukunft benötigten Daten
- Verwendung von Pipelining wo möglich
- Integration von Pooling in den Ausgangspuffer
- Verschmelzung von Zwischenschichten des neuronalen Netzes zur Reduzierung der Speichertransfers

Darüber hinaus wurde das Neuronale Netz modifiziert, sodass eine effizientere Berechnung bei gleicher Performanz ermöglicht wird. Konkret wurde das Netz folgendermaßen angepasst:

- Modifikation der Pooling-Größe, sodass sie dem Stride Parameter entspricht
- Reduzierung der Anzahl der Filter im ersten Layer
- Optimierung der Eingangsauflösung

Anschließend wurde ein Konzept für die Parallelisierung auf mehreren Rechenkernen entwickelt. Das Konzept ist für den Fall von zwei Rechenkernen in Abbildung 3 veranschaulicht. Des Weiteren wurden folgende zusätzliche Verbesserungen untersucht, implementiert und getestet: Größerer Ausgangs-Buffer, Kernel Linearisierung, breiterer Parameterbus, 8-bit Quantisierung.

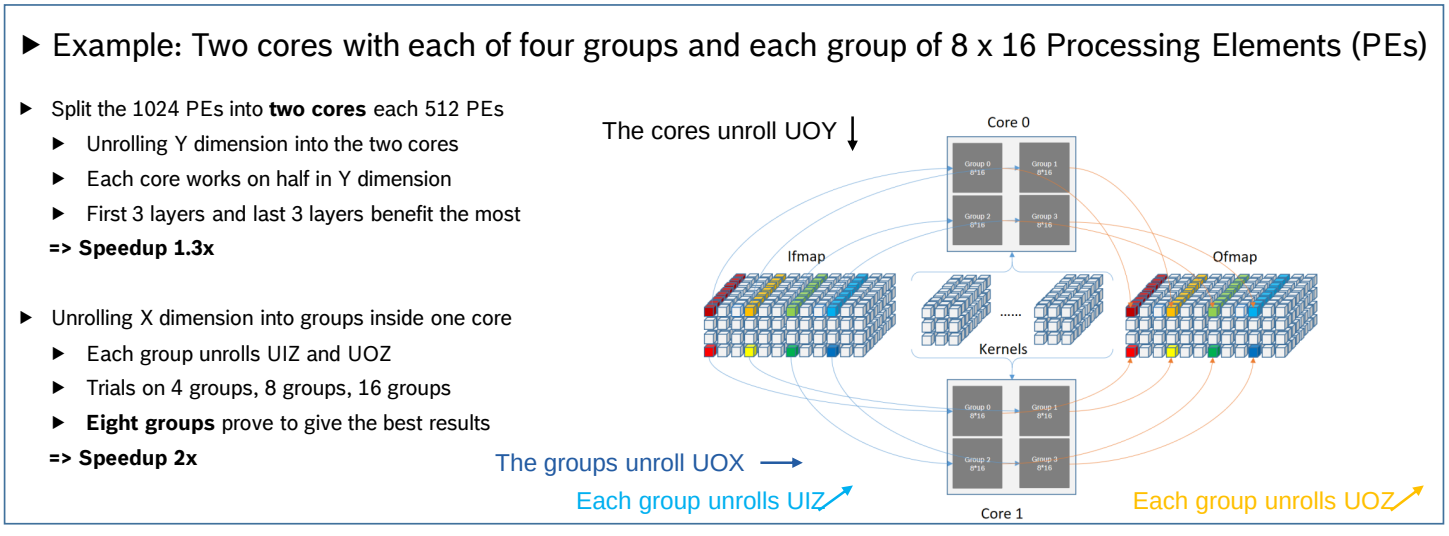

Abbildung 3: Konzept für die Parallelisierung am Beispiel von zwei Rechenkernen.

Alle Entwicklungsschritte führten zu einer schnelleren Verarbeitungsgeschwindigkeit von 5 bis hin zu 14 Frames pro Sekunde. Abbildung 4 zeigt die die jeweiligen Verarbeitungsgeschwindigkeiten (Frames pro Sekunde) für SqueezeNet nach den einzelnen Optimierungsschritten.

Für den entwickelten CNN IP-Core mit 1024 Recheneinheiten (PEs) konnte der Ressourcenverbrauch, die Leistungsaufnahme und die Verarbeitungsgeschwindigkeit mit dem Xilinx Tool Vivado 2018.3 wie folgt ermittelt werden: FPGA

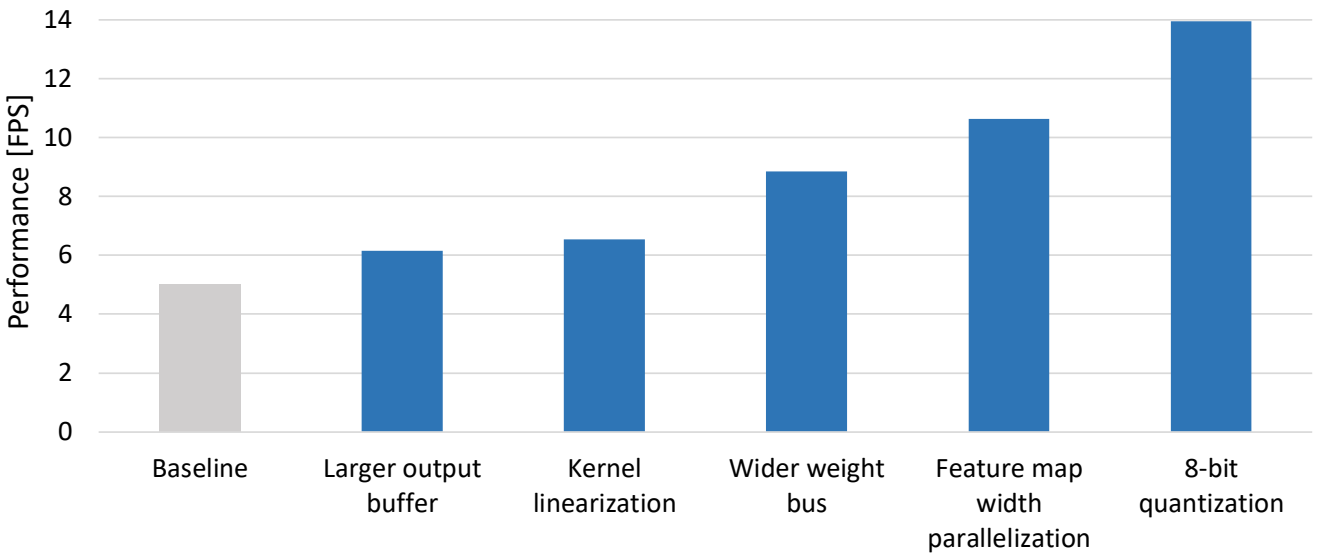

Abbildung 4: Verarbeitungsgeschwindigkeiten für SqueezeNet nach jeweils hinzugefügten Hardware Optimierungen.

Utilization: 1601 BRAM (87%), 1084 DSP (43%), FF 124117 (22%), LUT 123384 (45%) Power: ca. 10W (Dynamic 9,12W = 92%, Static 0,78W = 8%) Clock: 3ns Die Laufzeit konnte durch Funktionstest mit einer Bildgröße von 1472×736 Pixel gemessen werden. Für zwei verschieden Netze (SmallVGG & SqueezeNet): CNN@8bit ≈ 70ms, CNN@16bit ≈ 120ms per Frame. Die Abbildungen 5 und 6 zeigen die Ausgabe des Xilinx Tool Vivado 2018.3 für den Ressourcenverbrauch und die Leistungsaufnahme des IP-Cores.

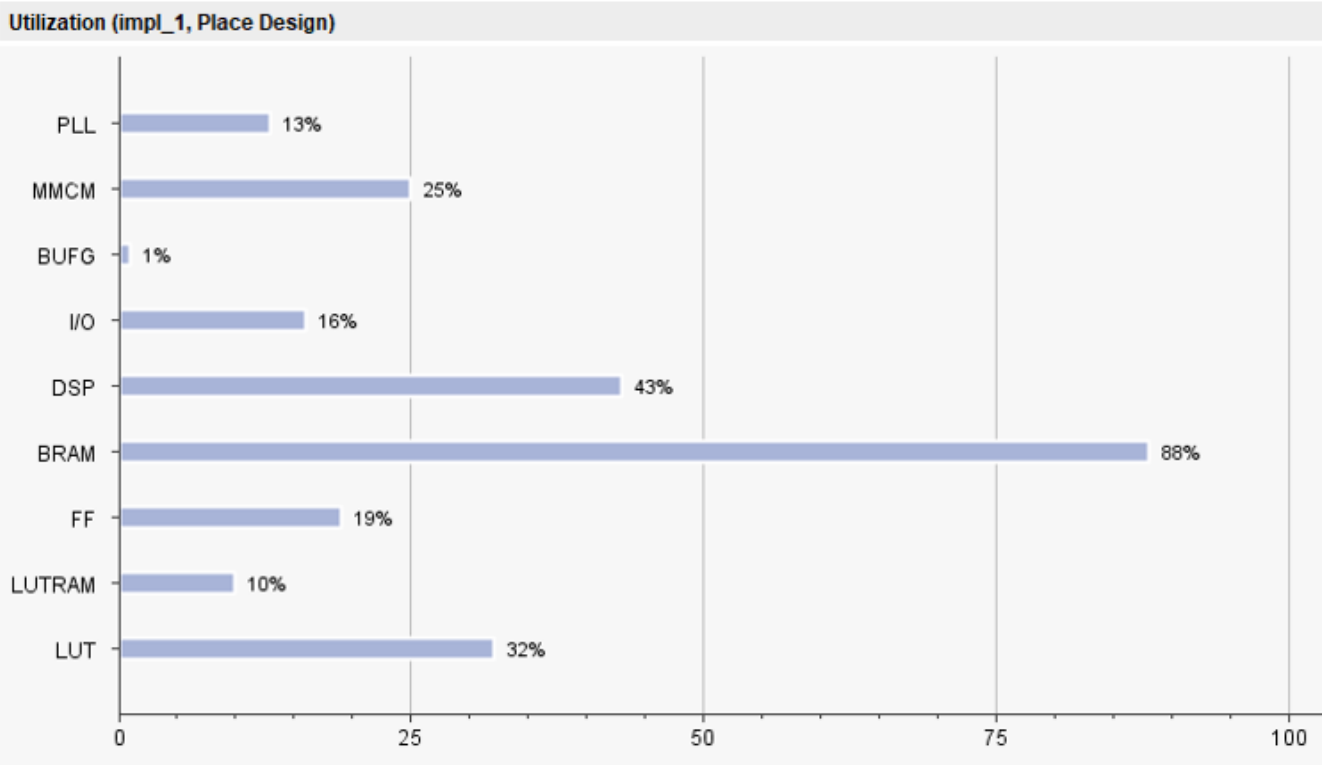

Abbildung 5: Ausgabe des Xilinx Tool Vivado 2018.3 zum Ressourcenverbrauch des IP-Cores.

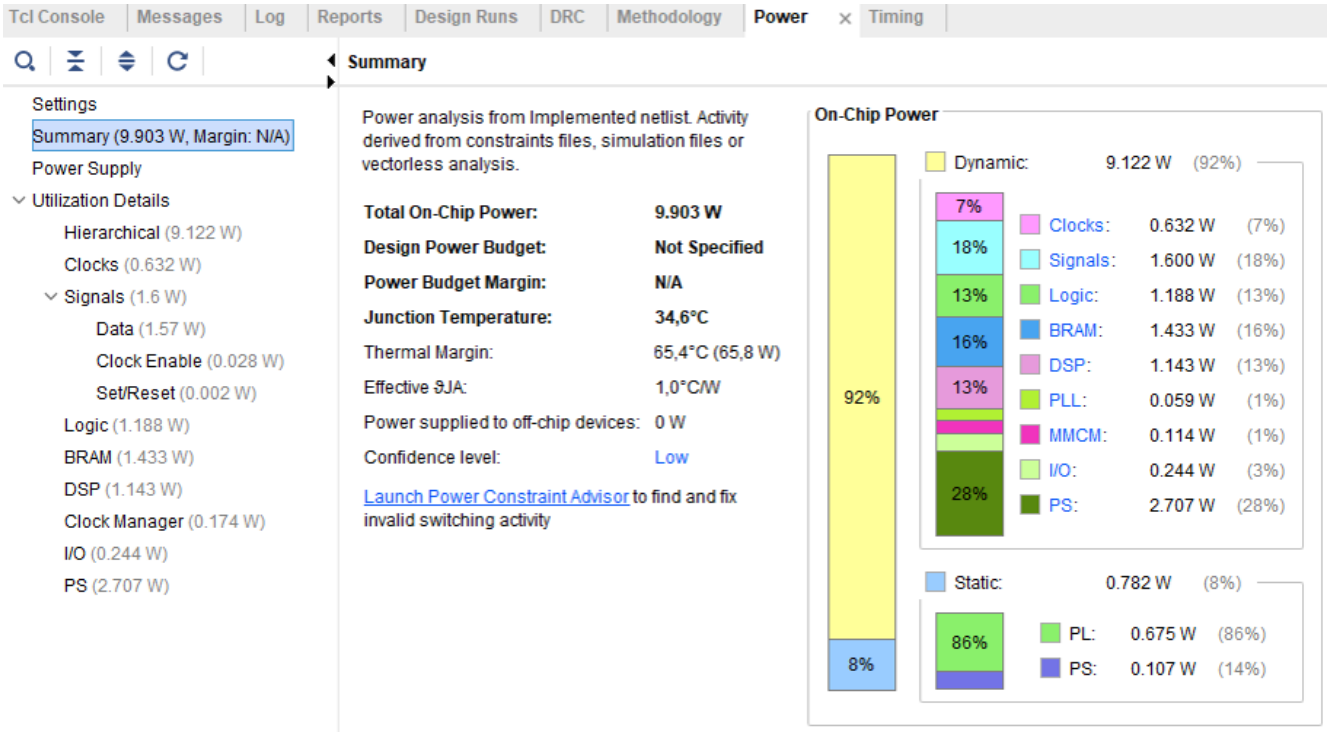

Abbildung 6: Ausgabe des Xilinx Tool Vivado 2018.3 zur Leistungsaufnahme des IP-Cores.

3.6 Absicherung gegenüber Hardware-Fehlern und Sicherheitsanalyse

3.6.1 Maßnahmen zur Absicherung des Systems auf algorithmischer Ebene

Zur Untersuchung der Auswirkung zufälliger Hardwarefehler auf die Ausgabe eines CNNs wurde ein Fehlerinjektions-Tool zur Simulation von zufälligen Bitflip-Fehlern in einzelnen Neuronen des neuronalen Netzes entwickelt. Das Tool basiert auf dem Keras Programmier-Framework mit Tensorflow Backend und ermöglicht somit eine effiziente GPU-gestützte Simulation. Zudem wurde das finale Datenformat (Quantisierung) der Zielhardware bei der Simulation mit berücksichtigt.

Experimente mit verschiedenen bekannten CNN Architekturen haben gezeigt, dass bereits einzelne Bitflip-Fehler zu Abweichungen der Ausgabe des Netzes führen können. Es wurde ein algorithmischer Ansatz zur Absicherung gegenüber kritischen Bitflip-Fehlern entwickelt. Dieser besteht in einer Anomalieerkennung in den Zwischenergebnissen des neuronalen Netzes. Die Ausgaben der Neuronen in allen Schichten den CNNs werden hierfür in einem sogenannten Feature Activation Trace (FAT) zusammengefasst. Um die Dimension dieses FAT zu reduzieren, wird jede Feature Map jeweils zu einem einzelnen Wert aufsummiert (siehe Abbildung 7).

Es wurde ein trainierbarer Anomaliedetektor in Form eines im Vergleich zum CNN kleinen Multi-Layer Perceptron Netzes gewählt (siehe Abbildung 8). Dieses Netz kann neben einem Ausgang für die Wahrscheinlichkeit eines Fehlers auch mit einem Ausgang für eine Korrektur des überwachten Netzes versehen werden.

Das Training des Anomaliedetektors erfolgt mittels überwachtem Lernen. Hierfür werden durch Fehlersimulation FATs mit kritischen und mit unkritischen bzw. keinen Fehlern erzeugt und als Trainingsdaten verwendet. Als Zielwerte für die Korrekturausgabe kann die Ausgabe des CNNs im fehlerfreien Fall verwendet werden. Das Verfahren wurde als Konferenzbeitrag [SGA18b] veröffentlicht.

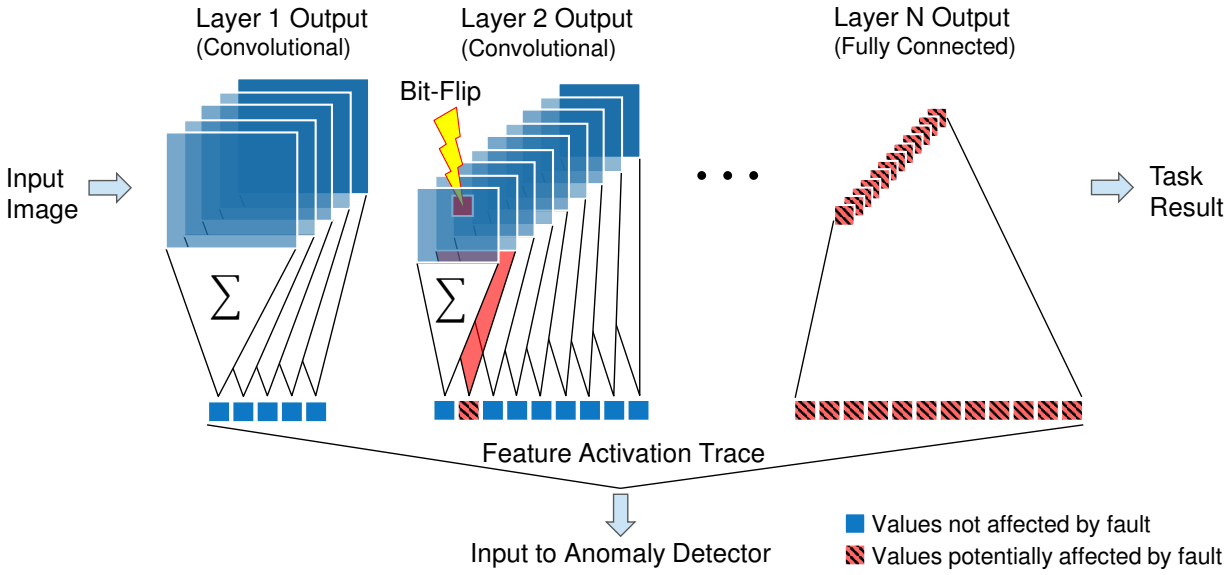

Abbildung 7: Erzeugung von Feature Activation Trace für die Anomalieerkennung in einem CNN [SGA18b].

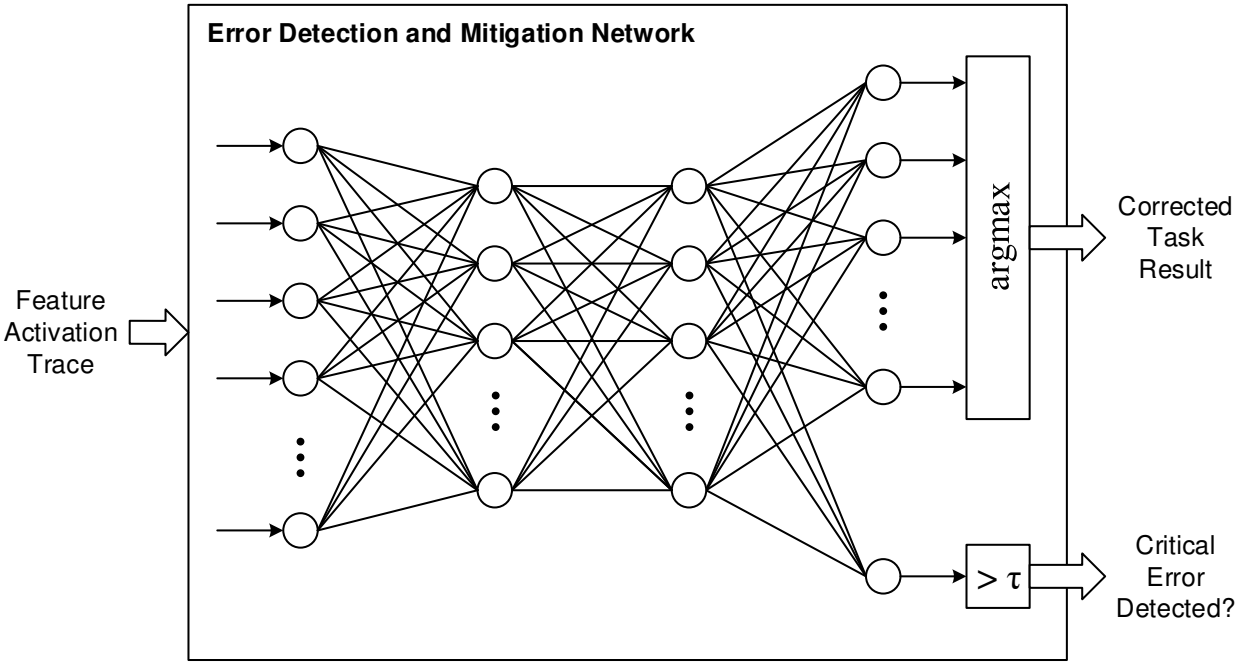

Abbildung 8: Neuronales Netz als Anomaliedetektor zur Erkennung von Fehlern des CNNs und Ermittlung eines korrigierten Ausgabewertes [SGA18b].

3.6.2 Maßnahmen zur Absicherung des Systems auf Hardware-Ebene

Für die Absicherung von angebundenen Speichern gegenüber zufälligen Hardwarefehlern stehen mit Error Detection and Correction Codes (z.B. Hamming Codes) bereits etablierte Maßnahmen auf Hardware (HW) Ebene zur Verfügung. Um eine Ende-zu-Ende Absicherung zu erzielen, müssen jedoch auch die Recheneinheiten des Beschleunigers abgesichert werden. Hier ist insbesondere eine effiziente Absicherung der Multiply-Accumulate (MAC) Arithmetik wünschenswert, da diese einen Großteil der Operationen in CNNs ausmacht. Es wurde ein Verfahren entwickelt, dass die Linearität der MAC Operationen ausnutzt um einen Fehlererkennung zu realisieren. Hierfür wird in jeder Schicht

des CNNs ein zusätzliches Prüfneuron (bzw. eine Prüf-Feature Map) hinzugefügt, wobei sich dessen Gewichte (bzw. Filter Kernel) als Summe der Gewichte (bzw. Filter Kernel) über alle anderen Neuronen (bzw. Features) der Schicht ergeben. Dies ist in Abbildung 9 skizziert.

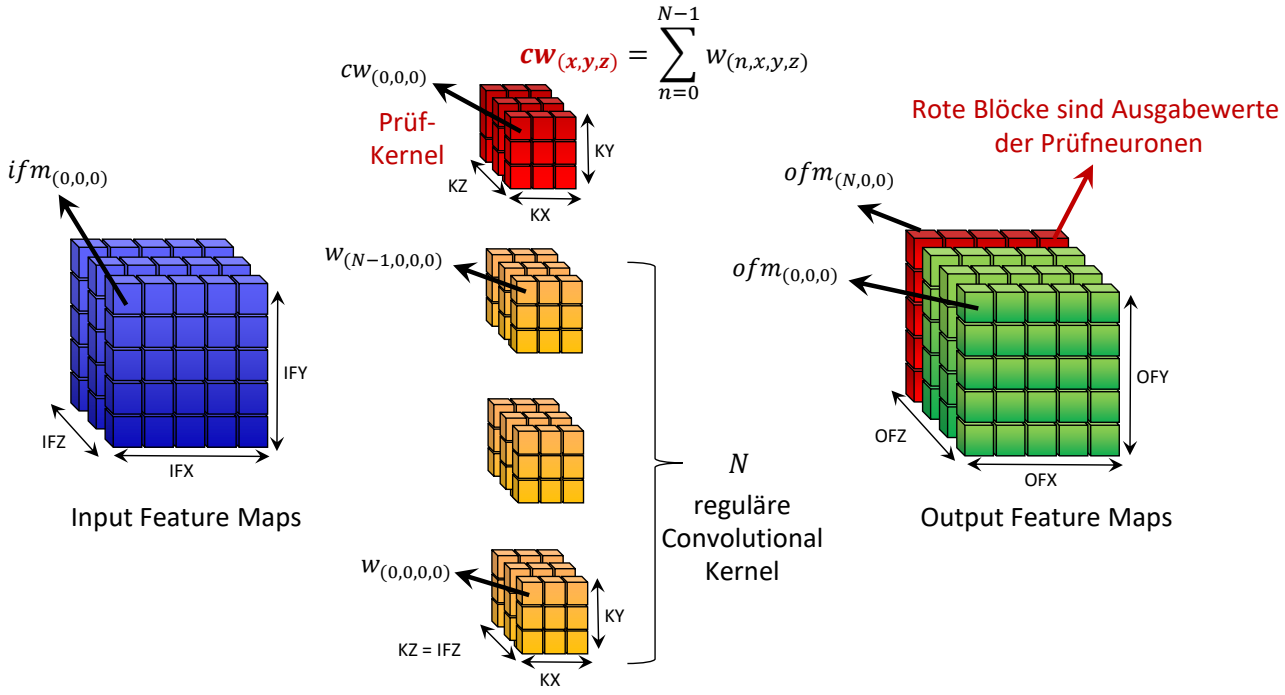

Abbildung 9: Prüf-Kernel für die Absicherung der MAC Operationen von CNNs auf Hardware-Ebene.

Im fehlerfreien Fall gilt: Die Summe aller regulären Neuronen Outputs entlang der Feature Dimension (OFZ) entspricht der Ausgabe des Prüf-Neurons. Somit ist eine Fehlererkennung möglich. Die Berechnung der Prüf-Werte sollte auf einem separaten Rechenelement erfolgen, um Common Cause Failures zu vermeiden. Ein entsprechender Implementierungsansatz wurde prototypisch für einen FPGA-basierten Beschleuniger untersucht. Der relative Mehraufwand, gemessen in MAC Operationen, liegt bei diesem Ansatz für ein typisches CNN zur semantische Segmentierung (1088×1920 Pixel Eingangsauflösung) bei unter 3%.

3.6.3 Neue Anforderungen an Standards für die funktionale Sicherheit von Machine Learning Systemen im Automobilbereich

Der Standard ISO 26262, der die funktionale Sicherheit von Elektrischen-/Elektronischen Systemen im Kraftfahrzeug gewährleisten soll, ist von der IEC 61508 abgeleitet und an die Automobilbranchen-spezifischen Gegebenheiten angepasst. Dabei wird der gesamte Produktentwicklungsprozess anhand eines V-Modells, inkl. Validierung und Verifikation betrachtet. Es werden keine Funktionen gesondert behandelt, die sich Methoden des Machine Learning und Deep Learning bedienen. Allerdings ergeben sich hier neue Eigenschaften, die deshalb mit der ISO 26262 nicht mehr erfasst werden können.

Beim Machine Learning (ML) werden Funktionen nicht mehr händisch programmiert, sondern mit Daten trainiert. Das Model wird in der Regel auf einem Testdatensatz statistisch evaluiert. Trotz allem ist eine umfassende Verifikation herausfordernd, weil es keine ausreichend genaue Spezifikation gibt, gegen die verifiziert werden kann. In vielen Fällen werden ML-basierte Funktionen dort eingesetzt, wo eine genaue Spezifikation fehlt und die Funktionalität durch Daten widergespiegelt wird. Die Spezifikation einer Fußgängererkennung im selbstfahrenden Fahrzeug zum

Beispiel müsste alle Varianten der Fußgänger kombiniert mit allen Umgebungsbedingungen enthalten. Bereits die Wahl der Beschreibungsart der Spezifikation fällt schwer. Eine Modellierung oder Strukturierung des potentiellen Eingaberaums ist auch herausfordernd. Erste Ansätze bestehen und sollten, wenn sie sich etablieren können, in einen Standard aufgenommen werden.

Auch die Anforderungen an die Daten sollten in der Spezifikation eingebunden werden. Diese Vorgabe gibt es so noch nicht, sollte aber in Betracht gezogen werden, da die Daten von besonderer Wichtigkeit sind. Verzerrte oder einseitige Daten können die Funktionalität und Zuverlässigkeit genauso beeinträchtigen wie fehlerhaft bezeichnete Daten.

Zudem sollte nach dem Training einer ML-Funktion, die tatsächlich erlernte Mustererkennung überprüft werden. Dabei reicht eine statistische Auswertung auf einem Testdatensatz nicht aus. Einzelnen Unterräume des Eingaberaums müssen auf systematische Fehler überprüft werden wie auch die Fähigkeit zur Generalisierung. Sind funktionale Unzulänglichkeiten bekannt, sollten diese überprüft werden. Methoden zu Erklärbarkeit und Interpretation von ML-Funktionen sind derzeit noch Gegenstand der Forschung, aber sollten zur Absicherung in Betracht gezogen werden.

Weiterhin wurden zwei Veröffentlichungen, [GMB18, BGS+19], auf einschlägigen Konferenzen erzielt, um das Bewusstsein für Anforderungen an zukünftige Versionen der ISO 26262 zu fördern.

3.7 Verifikation und Validierung

3.7.1 FPGA Evaluation

Die Integration des FPGA IP-Cores in die Bildverarbeitungskette des Gesamtsystems konnte mit Hilfe von mehreren Robot Operating System (ROS) Modulen realisiert werden. Abbildung 10 zeigt die modulare Pipeline und die in C++ entwickelten ROS Module (Nodes).

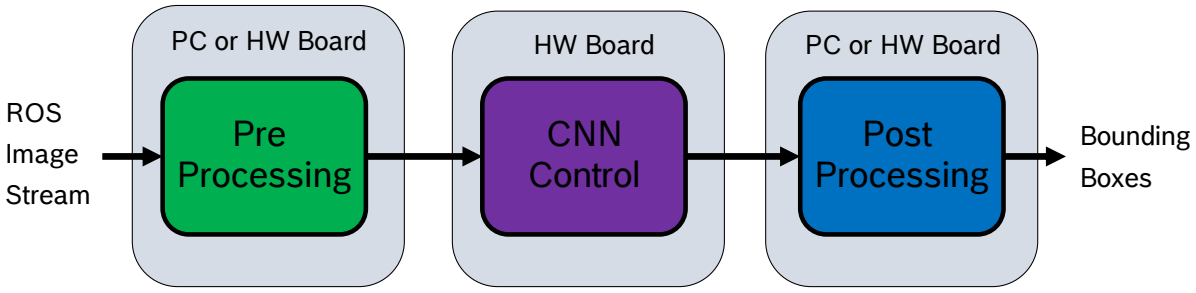

Abbildung 10: Modulare Verarbeitungskette bestehend aus ROS-Nodes.

Nach der Optimierung wird das Eingabebild zur Synchronisierung der Ergebnisse nicht mehr durch die Pipeline übertragen, stattdessen gibt es einen Bildpuffer mit Synchronisation der Ergebnisse an zentraler Stelle. Die weiteren ROS Knoten sind modular und konfigurierbar, um flexibel für die Integration verschiedener CNN-Topologien zu sein. Die ROS-Knoten wurden in C++ implementiert, außerdem wurde auf dem HW-Board die ARM NEON Engine verwendet, um eine weitere Steigerung der Effizienz zu erzielen.

Tabelle 4 zeigt die Laufzeit der einzelnen ROS Module (linke Spalte) im Versuchsträger (mittlere Spalte) und rein embedded (rechte Spalte). Die in Klammern angegebenen Zahlen entsprechen den Laufzeiten vor der Durchführung von Optimierungen der Module.

Hierzu wurde die komplette Bildverarbeitungskette auch ohne Host PC (rein embedded) in Betrieb genommen und als Demonstrator vorbereitet. Das heißt eine AVT Kamera ist an die Netzwerkschnittstelle des HW-Boards

Tabelle 4: Laufzeiten der einzelnen ROS Module im Versuchsträger und rein embedded. Initiale Werte ohne Optimierung stehen in Klammern.

Modul	Host PC + HW Board	HW Board only
CV-bridge	0ms (5ms)	0ms (5ms)
Resize	3ms	13ms
Transpose	5ms	22ms
CV-normalize	7ms	34ms (114ms) – Implementation on ARM NEON
CNN write image	25ms (53ms)	25ms (53ms)
CNN process	110ms (220ms)	110ms (220ms)
Post-processing	1ms (16ms)	11ms (260ms)
TOTAL	151ms (309ms)	215ms (686ms)

angeschlossen und liefert das Videobild direkt an den CNN IP-Core im FPGA. Dann erfolgt die Ausgabe des Videobildes und der Ergebnisse der Personenerkennung über den Display Port an einen Bildschirm.

3.7.2 Validierung im Versuchsfahrzeug

Für die Integration in das Versuchsfahrzeug wurde vorab eine stabile Halterung für das HW-Board konstruiert. Bei einem zusätzlichen Integrationsworkshop beim IKA in Aachen konnten wir die HW-Referenz-Plattform im Versuchsträger (VT) des IKA einbauen.

Die Abbildung 11 veranschaulicht die Integration des bei Bosch entwickelten CNN HW-Beschleunigers in das Gesamtsystem eines Testfahrzeugs des IKA.

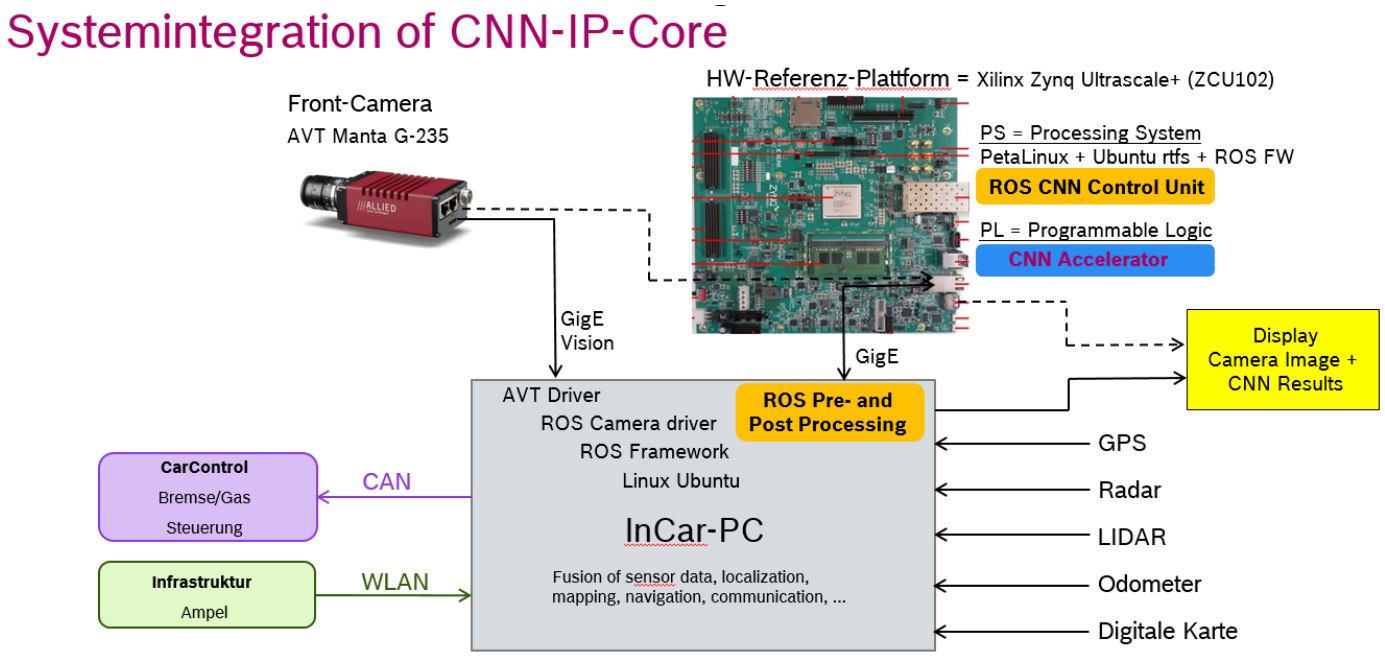

Abbildung 11: Integration des bei Bosch entwickelten CNN HW-Beschleunigers.

Als nächstes konnte die gesamte Bildverarbeitungskette inklusive Kameraanbindung, Vor- und Nachverarbeitung auf dem InCar-PC und Ansteuerung des IP-Cores sowie dessen CNN-Funktion auf der MPSoC Referenzplattform

im VT in Betrieb genommen werden (siehe Abbildung 12). Bei den Funktionstests des eingebetteten Computer-Vision-Systems sind weitere Testaufnahmen und neue Ergebnisse in einem realen Umfeld mit Bounding Boxen der Fußgänger Erkennung (PDet) entstanden.

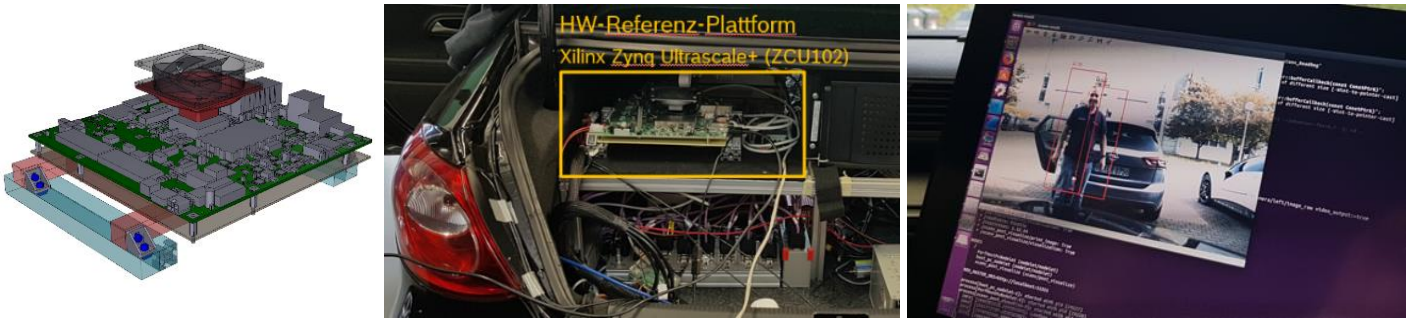

Abbildung 12: Die drei Bilder zeigen die Validierung im Versuchsfahrzeug.

4 Voraussichtlicher Nutzen und Verwertbarkeit der Ergebnisse

Die entwickelten Methoden zur Komprimierung und Quantisierung von Convolutional Neural Networks (CNNs) wurden in ein firmeninternes Entwicklungstool integriert und werden für die Hardware-Optimierung der CNNs in verschiedenen Machine Learning-basierten Produkten eingesetzt. Dabei konnte auch ein Transfer in andere Produktbereiche außerhalb des automatisierten Fahrens realisiert werden. Des Weiteren werden die entwickelten Robot Operating System (ROS) Module in der Vorausentwicklung für die Integration eingebetteter CNN Anwendungen in Kamera-basierten Prototypen eingesetzt.

Die bereits erfolgten Veröffentlichungen der relevanten Forschungsergebnisse auf wissenschaftlichen Fachkonferenzen können anderen Forschergruppen zukünftig als Referenz bei der Optimierung der Hardware-Beschleunigung neuronaler Netze dienen. Zudem können die veröffentlichten Anforderungen und Vorgehensweisen für die Validierung der Performanz von Machine Learning-basierten Funktionen im hochautomatisierten Fahren als ein Bestandteil für zukünftige Standardisierungen innerhalb der Automobilindustrie und verwandten Fachgebieten dienen.

5 Bekanntgewordener Fortschritt Dritter während der Projektlaufzeit

Während der Projektlaufzeit sind auf dem Gebiet der Hardware-Beschleunigung neuronaler Netze für das automatisierte Fahren an vielen Stellen Fortschritte erzielt worden. Neue System-on-Chip (SoC) Lösungen, die auf diesen Anwendungsfall optimiert sind, wurden vorgestellt (z.B. NVIDIA Tegra Xavier [Wik19], Tesla FSD Chip [Wik20]) Die Entwicklungen im akademischen und industriellen Umfeld wurden von Bosch fortlaufend beobachtet und berücksichtigt. Eine direkte Anpassung des Projektinhalts war dadurch allerdings nicht erforderlich.

6 Erfolgte und geplante Veröffentlichungen

Teile der im Projekt gewonnenen Erkenntnisse sind in die Veröffentlichungen [BGS+19, GMB18, SGA18b, VSGA19] eingeflossen. Darüber hinaus wurden nach Projektende zwei weitere Beiträge [SG20, GHST20], die auf den erzielten Projektergebnissen aufbauen, auf wissenschaftlichen Konferenzen veröffentlicht.

Die Forschung und Vorausentwicklung von Bosch veröffentlicht fortlaufend aktuelle Forschungsergebnisse in Form von Beiträgen auf Fachkonferenzen und in Fachzeitschriften, um den Austausch mit anderen Forschungsgruppen zu ermöglichen.

Literaturverzeichnis

[BGS+19] Burton, Simon ; Gauerhof, Lydia ; Sethy, Bibhuti B. ; Habli, Ibrahim ; Hawkins, Richard: Confidence Arguments for Evidence of Performance in Machine Learning for Highly Automated Driving Functions. In: Romanovsky, Alexander (Hrsg.) ; Troubitsyna, Elena (Hrsg.) ; Gashi, Ilir (Hrsg.) ; Schoitsch, Erwin (Hrsg.) ; Bitsch, Friedemann (Hrsg.): *Computer Safety, Reliability, and Security (SAFECOMP Workshops)* Bd. 11699. Cham, Switzerland : Springer, 2019 (LNCS), S. 365–377

[CMBB14] Cho, Kyunghyun ; Merrienboer, Bart van ; Bahdanau, Dzmitry ; Bengio, Yoshua: *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. E-print arXiv:1409.1259, 2014

[COR+16] Cordts, Marius ; Omran, Mohamed ; Ramos, Sebastian ; Rehfeld, Timo ; Enzweiler, Markus ; Benenson, Rodrigo ; Franke, Uwe ; Roth, Stefan ; Schiele, Bernt: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016

[CV95] Cortes, C. ; Vapnik, V.: Support-Vector Networks. In: *Machine Learning* 20 (1995), Nr. 3, S. 273–297

[DWSP11] Dollar, Piotr ; Wojek, Christian ; Schiele, Bernt ; Perona, Pietro: Pedestrian detection: An evaluation of the state of the art. In: *IEEE transactions on pattern analysis and machine intelligence* 34 (2011), Nr. 4, S. 743–761

[FS95] Freund, Y. ; Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: *Journal of Computer and System Sciences, Bd* 55 (1995), Nr. 1, S. 119–139

[GB10] Glorot, X. ; Bengio, Y.: Understanding the Difficulty of Training Deep Feedforward Neural Networks. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010

[GDDM14] Girshick, R. ; Donahue, J. ; Darrell, T. ; Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014

[GHST20] Gauerhof, Lydia ; Hagiwara, Yuki ; Schorn, Christoph ; Trapp, Mario: Considering Reliability of Deep Learning Function to Boost Data Suitability and Anomaly Detection. In: *5th IEEE International Workshop on Reliability and Security Data Analysis (RSDA – ISSRE 2020 Workshop)*, 2020

[GLSU13] Geiger, A ; Lenz, P ; Stiller, C ; Urtasun, R: Vision meets robotics: The KITTI dataset. In: *The International Journal of Robotics Research* 32 (2013), Nr. 11, S. 1231–1237

[GMB18] Gauerhof, Lydia ; Munk, Peter ; Burton, Simon: Structuring Validation Targets of a Machine Learning Function Applied to Automated Driving. In: Gallina, Barbara (Hrsg.) ; Skavhaug, Amund (Hrsg.) ; Bitsch, Friedemann (Hrsg.): *Computer Safety, Reliability, and Security (SAFECOMP)* Bd. 11093. Cham, Switzerland : Springer, 2018 (LNCS), S. 45–58

[HLMW17] Huang, Gao ; Liu, Zhuang ; Maaten, Laurens van d. ; Weinberger, Kilian Q.: Densely Connected Convolutional Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017

[HS97] Hochreiter, S. ; Schmidhuber, J.: Long short-term memory. In: *Neural Computation, Bd* 9 (1997), Nr. 8, S. 1735–1780

[IHM+16] IANDOLA, F. N. ; HAN, S. ; MOSKEWICZ, M. W. ; K, Ashraf ; DALLY, W. J. ; KEUTZER, K.: *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and* < 0.5*MB model size*. E-print arXiv:1602.07360, 2016

[KRA+20] KUZNETSOVA, Alina ; ROM, Hassan ; ALLDRIN, Neil ; UIJLINGS, Jasper ; KRASIN, Ivan ; PONT-TUSET, Jordi ; KAMALI, Shahab ; POPOV, Stefan ; MALLOCI, Matteo ; KOLESNIKOV, Alexander ; DUERIG, Tom ; FERRARI, Vittorio: The Open Images Dataset V4. In: *International Journal of Computer Vision* 128 (2020), Nr. 7, S. 1956–1981

[KSH12] KRIZHEVSKY, A. ; SUTSKEVER, I. ; HINTON, G.E.: Imagenet Classification with Deep Convolutional Networks. In: *Neural Information Processing Systems (NIPS)*, 2012

[LAE+16] LIU, Wei ; ANGUELOV, Dragomir ; ERHAN, Dumitru ; SZEGEDY, Christian ; REED, Scott ; FU, Cheng-Yang ; BERG, Alexander C.: SSD: Single Shot MultiBox Detector. In: LEIBE, Bastian (Hrsg.) ; MATAS, Jiri (Hrsg.) ; SEBE, Nicu (Hrsg.) ; WELLING, Max (Hrsg.): *Computer Vision – ECCV 2016*. Cham : Springer International Publishing, 2016, S. 21–37

[LBBH98] LECUN, Y. ; BOTTOU, L. ; BENGIO, Y. ; HAFFNER, P.: Gradient-based Learning applied to Document Recognition. In: *Proceedings of the IEEE* 86 (1998), Nr. 11, S. 2278–2324

[LSD15] LONG, J. ; SHELHAMER, E. ; DARRELL, T.: Fully Convolutional Networks for Semantic Segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015

[NORBK17] NEUHOLD, Gerhard ; OLLMANN, Tobias ; ROTA BULO, Samuel ; KONTSCHIEDER, Peter: The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017

[NVI15] NVIDIA: *NVIDIA Jetson TK1*. http://www.nvidia.de/content/tegra/automotive/pdf/jetson-tk1-brochure-web.pdf. Version: 05.05.2015, Abruf: 28.10.2020

[PMB13] PASCANU, R. ; MIKOLOV, T. ; BENGIO, Y.: On the difficulty of training recurrent neural networks. In: *Journal of Machine Learning Research* 28 (2013)

[RHGS15] REN, Shaoqing ; HE, Kaiming ; GIRSHICK, Ross ; SUN, Jian: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: CORTES, C. (Hrsg.) ; LAWRENCE, N. D. (Hrsg.) ; LEE, D. D. (Hrsg.) ; SUGIYAMA, M. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., 2015, S. 91–99

[SEV+20] SCHORN, Christoph ; ELSKEN, Thomas ; VOGEL, Sebastian ; RUNGE, Armin ; GUNTORO, Andre ; ASCHEID, Gerd: Automated design of error-resilient and hardware-efficient deep neural networks. In: *Neural Computing and Applications* (2020)

[SG20] SCHORN, Christoph ; GAUERHOF, Lydia: FACER: A Universal Framework for Detecting Anomalous Operation of Deep Neural Networks. In: *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, 2020

[SGA18a] SCHORN, Christoph ; GUNTORO, Andre ; ASCHEID, Gerd: Accurate Neuron Resilience Prediction for a Flexible Reliability Management in Neural Network Accelerators. In: *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018

[SGA18b] SCHORN, Christoph ; GUNTORO, Andre ; ASCHEID, Gerd: Efficient On-Line Error Detection and Mitigation for Deep Neural Network Accelerators. In: GALLINA, Barbara (Hrsg.) ; SKAVHAUG, Amund (Hrsg.) ; BITSCH, Friedemann (Hrsg.): *Computer Safety, Reliability, and Security (SAFECOMP)* Bd. 11093. Cham, Switzerland : Springer, 2018 (LNCS), S. 205–219

[SGA19] SCHORN, Christoph ; GUNTORO, Andre ; ASCHEID, Gerd: An Efficient Bit-Flip Resilience Optimization Method for Deep Neural Networks. In: *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, S. 1486–1491

[SHK+14] Srivastava, N. ; Hinton, G. ; Krizhevsky, A. ; Sutskever, I. ; Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In: *Journal of Machine Learning Research* 15 (2014), S. 1929–1958

[SHZ+18] Sandler, Mark ; Howard, Andrew ; Zhu, Menglong ; Zhmoginov, Andrey ; Chen, Liang-Chieh: MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018

[SLJ+14] Szegedy, C. ; Liu, W. ; Jia, Y. ; Sermanet, P. ; Reed, S. ; Anguelov, D. ; Erhan, D. ; Vanhoucke, V. ; Rabinovitch, A.: *Going Deeper with Convolutions*. 2014

[SZ15] Simonyan, Karen ; Zisserman, Andrew: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *International Conference on Learning Representations (ICLR)*, 2015

[VSGA19] Vogel, Sebastian ; Springer, Jannik ; Guntoro, Andre ; Ascheid, Gerd: Self-Supervised Quantization of Pre-Trained Neural Networks for Multiplierless Acceleration. In: *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, S. 1094–1099

[WIJK16] Wu, Bichen ; Iandola, Forrest ; Jin, Peter H. ; Keutzer, Kurt: *Supplementary Material: Designing Low Power Neural Network Architectures*. E-print arXiv:1612.01051, 2016

[WIJK17] Wu, Bichen ; Iandola, Forrest ; Jin, Peter H. ; Keutzer, Kurt: SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017, S. 129–137

[Wik19] WikiChip: *Tegra Xavier - Nvidia*. https://en.wikichip.org/wiki/tegra_xavier. Version: 09.12.2019, Abruf: 28.10.2020

[Wik20] WikiChip: *FSD Chip - Tesla*. https://en.wikichip.org/wiki/fsd_chip. Version: 28.10.2020, Abruf: 28.10.2020

[Xil14] Xilinx Inc.: *Xilinx Automotive Zynq-7000 All Programmable SoCs*. http://www.xilinx.com/publications/prod_mktg/ZynqAuto_ProdBrf.pdf. Version: 24.10.2014, Abruf: 28.10.2020

[YCW+20] Yu, Fisher ; Chen, Haofeng ; Wang, Xin ; Xian, Wenqi ; Chen, Yingying ; Liu, Fangchen ; Madhavan, Vashisht ; Darrell, Trevor: BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, S. 2636–2645

[ZBS17] Zhang, Shanshan ; Benenson, Rodrigo ; Schiele, Bernt: CityPersons: A Diverse Dataset for Pedestrian Detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017

[ZJRP+15] Zheng, S. ; Jayasumana, S. ; Romera-Paredes, B. ; Vineet, V. ; Su, Z. ; Du, D. ; Huang, C. ; Torr, P.H.S.: Conditional Random Fields as Recurrent Neural Networks. In: *International Conference on Computer Vision (ICCV)*, 2015

[ZZLS18] Zhang, Xiangyu ; Zhou, Xinyu ; Lin, Mengxiao ; Sun, Jian: ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018

Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) **Schlussbericht**

3. Titel

Schlussbericht
zum Teilvorhaben der Robert Bosch GmbH
Hardwareunterstütztes Machine Learning für hochautomatisiertes Fahren
im Verbundprojekt PARIS: Parallele Implementierungs-Strategien für das hochautomatisierte Fahren

4. Autor(en) [Name(n), Vorname(n)]	5. Abschlussdatum des Vorhabens **30.04.2020**
Schorn, Christoph **Gauerhof, Lydia** **Kunze, Christoph** **Luther, Marc** **Runge, Armin** **Vogel, Sebastian**	6. Veröffentlichungsdatum
	7. Form der Publikation
8. Durchführende Institution(en) (Name, Adresse) **Robert Bosch GmbH** **Zentralbereich Forschung und Vorausentwicklung** **Robert-Bosch-Campus 1** **71272 Renningen**	9. Ber. Nr. Durchführende Institution
	10. Förderkennzeichen **16ES0610**
	11. Seitenzahl **25**
12. Fördernde Institution (Name, Adresse) **Bundesministerium für** **Bildung und Forschung (BMBF)** **53170 Bonn**	13. Literaturangaben **42**
	14. Tabellen **4**
	15. Abbildungen **12**

16. Zusätzliche Angaben

17. Vorgelegt bei (Titel, Ort, Datum)

18. Kurzfassung

Die benötigte Rechenleistung für neuartige Algorithmen, insbesondere tiefe neuronale Netze, die vor allem im Rahmen der Perzeption von automatisierten Fahrzeugen eingesetzt werden, ist besonders hoch. Da die neuartigen Algorithmen klassische Verfahren deutlich in der Performanz übertreffen, sind sie beim automatisierten Fahren unentbehrlich. Vor diesem Hintergrund wurde die Entwicklung eines dedizierten Hardware-Beschleunigers als IP-Core adressiert. Dabei wurden neue Detektionstechnologien für die Umfelderkennung in Form eines neuronalen Netzes für die Fußgängererkennung evaluiert und in effizienter Weise auf den entwickelten Hardware-Beschleuniger abgebildet. Darüber hinaus wurden Methoden zur Komprimierung und Quantisierung von hochperformanten neuronalen Netzen für die Umfelderkennung entwickelt und evaluiert, welche deren Echtzeitfähigkeit und Energieverbrauch bei der Ausführung auf dedizierten Hardware-Beschleunigern verbessern. Zudem wurden neuartige Verfahren für die Absicherung von neuronalen Netzen auf algorithmischer Ebene und auf Hardware-Ebene entwickelt, welche die Robustheit und Fehlertoleranz der Umfelderkennung bei geringem Mehraufwand hinsichtlich der Rechenoperationen erhöhen. Schließlich wurden neue Anforderungen und Vorgehensweisen für die Validierung der Performanz von Machine Learning-basierten Funktionen im hochautomatisierten Fahren entwickelt und auf wissenschaftlichen Konferenzen veröffentlicht. Alles in allem konnten wichtige Ergebnisse zur Absicherbarkeit von neuronalen Netzen und der hierfür eingesetzte Hardware beim automatisierten Fahren unter Berücksichtigung der Rechenleistung erzielt werden. Die Validierung fand in Zusammenarbeit mit Konsortialpartnern in einem Versuchsfahrzeug statt.

19. Schlagwörter
PARIS, automatisiertes Fahren, neuronale Netze, Hardware Beschleunigung, funktionale Sicherheit

20. Verlag	21. Preis

Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) **report**

3. title
Schlussbericht
zum Teilvorhaben der Robert Bosch GmbH
Hardwareunterstütztes Machine Learning für hochautomatisiertes Fahren
im Verbundprojekt PARIS: Parallele Implementierungs-Strategien für das hochautomatisierte Fahren

4. author(s) (family name, first name(s))	5. end of project **30 April 2020**
Schorn, Christoph **Gauerhof, Lydia** **Kunze, Christoph** **Luther, Marc** **Runge, Armin** **Vogel, Sebastian**	6. publication date
	7. form of publication
8. performing organization(s) (name, address) **Robert Bosch GmbH** **Zentralbereich Forschung und Vorausentwicklung** **Robert-Bosch-Campus 1** **71272 Renningen**	9. originator's report no.
	10. reference no. **16ES0610**
	11. no. of pages **25**
12. sponsoring agency (name, address) **Bundesministerium für** **Bildung und Forschung (BMBF)** **53170 Bonn**	13. no. of references **42**
	14. no. of tables **4**
	15. no. of figures **12**

16. supplementary notes

17. presented at (title, place, date)

18. abstract
The required compute power for modern algorithms, in particular deep neural networks, which are employed within the perception of automated vehicles, is very high. Since novel algorithms outperform classical approaches by a significant margin, they are essential for the realization of automated driving. In this context, the development of a dedicated hardware accelerator was addressed. At the same time, new detection technologies for the environmental perception in form of a neural network for pedestrian detection were evaluated and mapped to the developed hardware accelerator. Moreover, methods for the compression and quantization of high-performant neural networks for perception were developed and evaluated in order to improve the real-time capability and energy consumption when running those networks on dedicated hardware accelerators. Furthermore, new methods for safeguarding neural networks at the algorithm and hardware levels, which increase robustness and fault tolerance of perception with low computational overheads, were developed. Finally, new requirements and approaches for the validation of machine learning performance in highly automated driving functions were developed and published at scientific conferences. All in all, important results regarding the protectability of neural networks and their hardware within automated driving applications were achieved. The validation was performed together with the consortium partners using a test car.

19. keywords
PARIS, automated vehicles, neural networks, hardware acceleration, functional safety

20. publisher	21. price