FPGA-beschleunigte Objekterkennung von kleinen Objekten in der Erdbeobachtung

I. Nussbaumer, F. Schramka, L. Etesi, R. Müller, N. Venturi

Edge Computing stösst in den letzten Jahren zunehmend auch auf Interesse in der satellitengestützten Erdbeobachtung. In diesem Projekt entwickeln wir ein Objekterkennungsmodell, welches kleine Objekte auf Satellitenbildern detektiert, aber als Embedded System direkt auf der jeweiligen Plattform ausgeführt wird. Es wurde ein Xilinx UltraScale+ FPGA verwendet, um das ML Modell effizient auf der Xilinx Deep Learning Processor Unit (DPU) auszuführen. Als Detektor wurde ein YOLOX-Modell gewählt, welches quantisierungssensitives Training unterstützt. Vorläufige Ergebnisse zeigen, dass ein quantisierungssensitives Training die Auswirkungen der Quantisierung auf ein akzeptables Mass begrenzt. Wir vergleichen zwei YOLOX-Modelle in drei verschiedenen Konfigurationen (Float-Modelle, quantisierte Modelle und trainierte Modelle mit Quantisierung) und messen die Genauigkeit, die Ausführungsgeschwindigkeit und das Leistungsprofil auf zwei verschiedenen DPU-Konfigurationen.

Edge Computing stösst in den letzten Jahren zunehmend auch auf Interesse in der satellitengestützten Erdbeobachtung. In diesem Projekt entwickeln wir ein Objekterkennungsmodell, welches kleine Objekte auf Satellitenbildern detektiert, aber als Embedded System direkt auf der jeweiligen Plattform ausgeführt wird. Es wurde ein Xilinx UltraScale+ FPGA verwendet, um das ML Modell effizient auf der Xilinx Deep Learning Processor Unit (DPU) auszuführen. Als Detektor wurde ein YOLOX-Modell gewählt, welches quantisierungssensitives Training unterstützt. Vorläufige Ergebnisse zeigen, dass ein quantisierungssensitives Training die Auswirkungen der Quantisierung auf ein akzeptables Mass begrenzt. Wir vergleichen zwei YOLOX-Modelle in drei verschiedenen Konfigurationen (Float-Modelle, quantisierte Modelle und trainierte Modelle mit Quantisierung) und messen die Genauigkeit, die Ausführungsgeschwindigkeit und das Leistungsprofil auf zwei verschiedenen DPU-Konfigurationen.

Edge Computing stösst in den letzten Jahren zunehmend auch auf Interesse in der satellitengestützten Erdbeobachtung. In diesem Projekt entwickeln wir ein Objekterkennungsmodell, welches kleine Objekte auf Satellitenbildern detektiert, aber als Embedded System direkt auf der jeweiligen Plattform ausgeführt wird. Es wurde ein Xilinx UltraScale+ FPGA verwendet, um das ML Modell effizient auf der Xilinx Deep Learning Processor Unit (DPU) auszuführen. Als Detektor wurde ein YOLOX-Modell gewählt, welches quantisierungssensitives Training unterstützt. Vorläufige Ergebnisse zeigen, dass ein quantisierungssensitives Training die Auswirkungen der Quantisierung auf ein akzeptables Mass begrenzt. Wir vergleichen zwei YOLOX-Modelle in drei verschiedenen Konfigurationen (Float-Modelle, quantisierte Modelle und trainierte Modelle mit Quantisierung) und messen die Genauigkeit, die Ausführungsgeschwindigkeit und das Leistungsprofil auf zwei verschiedenen DPU-Konfigurationen.

1. Einleitung

Weltrauminstrumente sind seit jeher auf bordseitige Datenverarbeitungsmöglichkeiten angewiesen, z. B. um die Daten neu zu formatieren oder zu kalibrieren, bevor die Beobachtungen an den Boden gesendet werden (Krucker et al., 2020). Das Edge Computing geht jedoch darüber hinaus: Es bedeutet, dass hochentwickelte bodengestützte Verarbeitungsschritte und autonome Entscheidungsalgorithmen in weltraumgestützte Systeme verlagert werden, um die kostspielige Übertragung von Telemetriedaten zu optimieren oder neue Anwendungsfälle wie die Echtzeit-Erkennung von Naturkatastrophen in der Erdumlaufbahn zu ermöglichen (Mateo-Garcia et al., 2021). Angesichts dieses Potenzials ist maschinelles Lernen zu einem strategischen Schwerpunkt für Raumfahrtbehörden auf der ganzen Welt geworden.

Das Edge Computing im Weltraum ist jedoch mit Herausforderungen verbunden. Die Umgebung schränkt die verfügbaren Hardwareressourcen (elektrische Leistung, Rechenleistung oder Speicher) aufgrund von Problemen mit der Wärmeableitung und Strahlung stark ein und begrenzt die verfügbare Bandbreite. Schlechte autonome Entscheidungen oder eine schlechte Datenverarbeitung an Bord können unwiderrufliche Folgen haben, von der Datenkorruption bis hin zum Scheitern einer Mission.

Unser langfristiges Ziel ist es, eine durchgängige Pipeline für die Erfassung und Verarbeitung von Erdbeobachtungsdaten sowie für das Bildverarbeitungssystem zu entwerfen und zu implementieren, die an Bord eines Satelliten laufen und selbstständig Alarme erzeugen kann. Zu diesem Zweck entwickeln wir: i) eine kompakte Bildverarbeitungspipeline, um Sensor und optische Reaktionen zu korrigieren, die Erdgeometrie zu berücksichtigen und das Bild hinreichend auf eine Position auf der Erdoberfläche abzubilden, ii) ein robustes maschinelles Lernmodell, das auf begrenz-

ter Hardware effizient ausgeführt werden und kleine Objekte wie Schiffe erkennen kann, und iii) ein Einbettungskonzept der Bildverarbeitungspipeline und des Objekterkennungsmodell für einer Plattform, welche im Weltraum eingesetzt wird.

2. Verwandte Arbeiten

Wie bereits erwähnt, ist Edge Computing im Weltraum sehr vielversprechend, birgt jedoch verschiedene technische, umweltbedingte und operationelle Herausforderungen (Furano et al., 2020a, 2020b). Die Evaluierung geeigneter Hardwarekonfigurationen und -plattformen für das High-Performance-Computing in der Raumfahrt ist ein wichtiger Bestandteil der Arbeit. Alltägliche handelsübliche Komponenten wie leistungsstarke CPUs oder GPUs eignen sich nicht für den Einsatz im Weltraum, vor allem aufgrund von Problemen mit der Wärmeableitung, die die nutzbare Energie einschränken, und Umweltgefahren wie Strahlungseffekte und ionisierte Partikel, die Berechnungen und Speicher stören. Traditionell wurden komplexe Datenverarbeitungsaufgaben an Bord mit massgeschneiderten FPGA-Implementierungen (Field-Programmable Gate Arrays) realisiert. Diese zeichnen sich durch hohe Zuverlässigkeit und Rechenleistung aus und bieten grosse Flexibilität in Kombination mit anderen Onboard-Ressourcen, z. B. als System-on-Module (SoM) Lösung (George and Wilson, 2018). Etablierte Hersteller von Verarbeitungskomponenten passen ihre Plattformen so an, dass sie gefährlichen Umwelteinflüssen besser widerstehen und weniger Strom verbrauchen. Der im Januar 2022 erfolgreich gestartete Nanosatellit La Jument von Lockheed Martin verwendet einen NVIDIA Jetson-Grafikprozessor zur Verbesserung der Bildqualität. Und der im September 2020 gestartete Nanosatellit PhiSat-1 der Europäischen Weltraumorganisation nutzt eine Intel Movidius Myriad 2 Vision Processing Unit (VPU), um die Bildverarbeitungsaufgaben an Bord zu beschleunigen. Andere konzentrieren sich auf aufkommende Plattformen, die auf neuen Architekturen wie RISC-V basieren, einer Open-Source-Hardware-Architektur, die von etablierten Herstellern von Raumfahrt-Hardware wie Gaisler übernommen wurde. RISC-V, insbesondere unter Verwendung der RISC-V-Vektorerweiterung, ist sehr vielversprechend für Anwendungen des maschinellen Lernens (ML) im Weltraum (Di Mascio et al., 2021).
YOLO-Modelle (You Only Look Once) haben gute Ergebnisse bei der Erkennung kleiner Objekte in Erdbeobachtungsdaten erzielt. Mehrere Arbeiten zeigen Implementierungen in ähnlichen Schiffserkennungsszenarien auf der Grundlage von YOLOv3 (Wang et al., 2022; Xu et al., 2022) oder YOLOv4 (Verma et al., 2022). Der Nachweis einer hohen Empfindlichkeit gegenüber kleinen Objekten ist für unsere Arbeit direkt relevant, obwohl in keiner der genannten Arbeiten die Erkennungsmodelle an Bord eines eingebetteten Systems ausgeführt werden.

3. Hintergrund

Dieser Abschnitt beschreibt die Xilinx-Plattform, die wir in diesem Projekt verwenden, ihre spezialisierte Modellausführungseinheit für maschinelles Lernen, die DPU sowie die Vitis AI Toolchain und das Quantization-aware Training (QAT).

Xilinx UltraScale+ System-on-Chip (SoC)

Wir verwenden für dieses Projekt ein Mercury-XU8 System-on-Module (SoM), das von Enclustra hergestellt wird. Die Hauptkomponente des SoM ist der Xilinx SoC, der ein ARM Cortex-basiertes Verarbeitungssystem mit programmierbarer FPGA-Logik kombiniert. Obwohl der Stromverbrauch dieses Geräts anwendungsspezifisch ist, liegt er deutlich unter dem eines GPU-basierten Systems für universelle Anwendungen. Durch die Möglichkeiten der parallelen Verarbeitung im FPGA sind die Inferenzoperationen immer noch wesentlich effizienter und schneller als die sequentielle Ausführung auf einem klassischen eingebetteten Mikrocontroller ohne spezielle Beschleunigungshardware. Ein solches Gerät stellt einen interessanten Kompromiss zwischen den verfügbaren Rechenressourcen und dem Stromverbrauch dar.

Deep Learning Processor Unit (DPU)

Das Modell für maschinelles Lernen wird auf dem FPGA in einem von Xilinx bereitgestellten IP-Block namens DPU ausgeführt. Diese DPU ist ein Beschleuniger, der auf Berechnungen für Anwendungen des maschinellen Lernens zugeschnitten ist. Sie verfügt über einen speziellen Befehlssatz, um gängige Operationen wie Konvolutions- und ReLU-Funktionen effizient zu berechnen. Die DPU nutzt die triviale Parallelisierbarkeit vieler ML-Operationen, indem sie die Arbeitslast auf mehrere Verarbeitungselemente (PE) verteilt. Diese führen die Berechnungen an den Eingabedaten gemäss dem kompilierten Befehlssatz aus, der angibt, wie die Inferenz für ein bestimmtes Netz durchgeführt werden muss.
Die DPU ist nicht speziell für ein einzelnes Netz konfiguriert, sondern kann für die Berechnung verschiedener Netze mit derselben Hardware verwendet werden. Die DPU verfügt ausserdem über mehrere Ebenen des On-Chip-Cache, wodurch die Anzahl der Zugriffe auf den externen DRAM minimiert und die Energieeffizienz und Leistung verbessert wird.

Xilinx Vitis AI Toolchain

Xilinx bietet die Vitis AI Toolchain für DPU-bezogene Arbeiten an. Diese Tools konvertieren Machine-Learning-Modelle aus gängigen Frameworks (TensorFlow, Caffe, PyTorch) in eine Liste von Anweisungen für die DPU. Zunächst wird ein reguläres Modell mit einem der unterstützten ML-Frameworks wie YOLOX trainiert. Anschliessend wird das Modell quantisiert, indem die Modellgewichte von Fliesskommazahlen in 8-Bit-Ganzzahlen konvertiert werden. Die meisten weit verbreiteten Hardware-Beschleuniger unterstützen nur 8-Bit-Präzision. Der Präzisionsverlust geht in der Regel mit einem – manchmal erheblichen – Genauigkeitsverlust einher. Um die Leistung des quantisierten Modells zu verbessern, kann es mit quantisierungssensitivem Training feinabgestimmt werden, wobei

Parameter	FM0.1	AM0.1	FM0.5	AM0.5	FL0.5	AL0.5
Datentyp	f32	i8	f32	i8	f32	i8
Modellgrösse	M	M	M	M	L	L
Datensatz	0.1	0.1	0.5	0.5	0.5	0.5
Epochen (E)	200	60	400	60	400	60
Warmup E	5	5	20	5	20	5
Keine Aug. E	15	15	200	60	150	60
Multiscale	1	1	1	1	2	1
Mosaic	1	1	1	1	0.5	0.5
Mixup	1	1	1	1	0.5	0.5
hsv	1	1	1	1	0.5	0.5

Tab. 1: Parameter für das Training.

alle Werte im Forward Pass quantisiert werden, um die Quantisierungseffekte zu simulieren. Schliesslich wird das quantisierte Modell konvertiert und auf das Gerät mit der DPU hochgeladen. Vitis AI bietet auch Support-Code zur einfachen Integration von Modellen in C++-Code.

4. Datensatz und ML-Modell-Konfigurationen

In diesem Kapitel wird der Datensatz beschrieben, der für das Modelltraining und das Design des maschinellen Lernmodells verwendet wurde.

Datensatzbeschreibung

Swisstopo stellt ihren SWISSIMAGE-Datensatz mit einer Ground Sampling Distance (GSD) von 0.1m/Pixel im Flachland und 0.25m/Pixel über den Alpen zur Verfügung, der 1 Quadratkilometer pro Bild abdeckt. Basierend auf diesen Bildern haben wir einen neuen Datensatz erstellt, der 102 Quadratkilometer des Thunersees abdeckt und Annotationen für 3'449 Schiffe auf dem Wasser, in einem Hafen oder an Land enthält.

Wir zerlegen die Bilder in 500 × 500 Pixel, um die Objekterkennungsmodelle zu trainieren. Zusätzlich haben wir die Bilder um den Faktor 5 verkleinert, um zusätzliche Testbilder mit einer GSD von 0,5m/Pixel zu erzeugen.

Abbildung 1 zeigt einen 50m × 50m Ausschnitt des Bielersees mit vier verschiedenen GSDs zum Vergleich. Für unseren Anwendungsfall möchten wir kleine Privatschiffe erkennen, was bei GSDs von 1m/Pixel und höher unmöglich wird.

Trainierte Modelle

Wir haben das YOLOX-Modell für unser Schiffserkennungsszenario in unserem Datensatz verfeinert. YOLOX ist das komplexeste Objekterkennungsmodell, das von unseren Werkzeugen unterstützt wird, und kann quantisierungssensitiv trainiert werden. Wir haben sechs Varianten des Modells trainiert, wie in Tabelle 1 dargestellt. Die Modellnamen werden aus dem Modelltyp (Float, Quantized oder QAT), der Modellgrösse (Medium oder Large) und der GSD/Datensatz gebildet:

5. Vorläufige Ergebnisse

Wir haben die Modelle direkt in PyTorch evaluiert. Zeit und Leistung wurden gemessen, während die QAT-Modelle AM0.1, AM0.5 und AL05 auf zwei verschiedenen DPU-Konfigurationen liefen, mittel (1024 PE) als Basis und gross (4096 PE) als Obergrenze. Für jedes QAT-Modell wurden über 401 Bilder, die durchschnittliche Bildverarbeitungszeit sowie der Spitzen- und Durchschnittsstromverbrauch gemessen. Die Leerlaufleistung wurde vor und nach jedem Lauf gemessen.

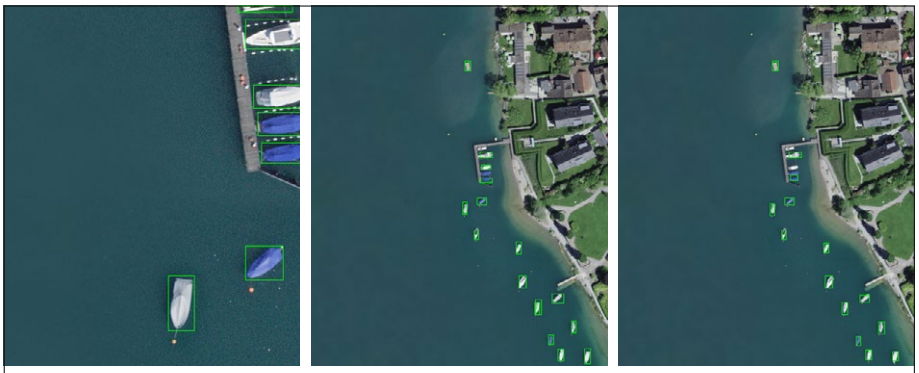
(a) 0.1m/pixel (b) 0.5m/pixel (c) 1.0m/pixel (d) 2.0m/pixel

Abb. 1: Luftbilder mit unterschiedlichen Auflösungen. Kleine, private Schiffe sind bei niedrigeren Auflösungen nur schwer zu erkennen.

Einige Werte benötigen eine Erläuterung:

- Eine leicht unterschiedliche Trainingskonfiguration kann die geringfügig besseren Recall- und Präzisionswerte für Modell AM0.5 gegenüber Modell AL0.5 erklären.
- Die gemessenen hohen Leistungswerte im Leerlauf für beide DPUs zeigen, dass unsere Hardwarekonfiguration noch nicht optimiert ist (z. B. keine Verwendung der CPU- und FPGA-Low-Power-Modi). Ausserdem umfasst die gemessene Leistung das gesamte System: CPU, FPGA, Speicher und Peripheriegeräte.
- Die Werte für die durchschnittliche Bildverarbeitungszeit, den Leerlauf, die durchschnittliche Leistung und die Spitzenleistung werden für die mittlere und grosse DPU als mittel/gross angegeben.

Abbildung 2 zeigt von der eingebetteten Plattform erzeugte Bilder, die erkannte Schiffe mit den QAT-Modellen AM0.1, AM0.5 und AL05 mit einer GSD von 0,1m/Pixel bzw. 0,5m/Pixel anzeigen.



(a) Model AM0.1 (b) Model AM0.5 (c) Model AL0.5

Abb. 2: Die Bilder wurden von dem Embedded System aufgenommen; Schiffe wurden mit verschiedenen Modellen und Auflösungen erkannt. Alle drei Modelle identifizieren die meisten Schiffe, aber die Bilder mit geringerer Auflösung führen dazu, dass mehr kleine Schiffe übersehen werden.

6. Diskussion und Fazit

Wir haben gezeigt, dass die Erkennung von kleinen Objekten wie Schiffen auf einem Xilinx Ultrascale+ FPGA und einer 8-Bit-DPU unter Verwendung eines YOLOX-Modells mit quantisierungssensitivem Training mit vielversprechender Genauigkeit möglich ist. Die Ergebnisse sind besonders ermutigend, wenn man die Leistung pro Watt trotz begrenzter Optimierungsbemühungen betrachtet. Mit unserem AL0.5-Modell auf der grossen DPU erreichen wir ein Äquivalent von 575 GFLOP/s bei 10,6 W gemessener Systemleistung.

Was das maschinelle Lernmodell betrifft, so glauben wir, dass die Erkennungsgenauigkeit durch zusätzliche Trainingsdaten, umfangreicheres Training und eine bessere Erweiterung des GSD-Datensatzes mit 0,5m/Pixel verbessert werden kann. Die Quantisierungskonfiguration und die Parameter der Modellkompilierung sollten weiter optimiert werden. Andere YOLO-Varianten könnten eine bessere Genauigkeit und/oder eine schnellere Inferenz ermöglichen.

Die eingebettete Plattform kann optimiert werden, insbesondere im Hinblick auf den Stromverbrauch, indem die gesamte Konfiguration, einschliesslich der DPU-Konfiguration und Design-Optimierungen, angepasst wird oder indem bestimmte Aufgaben von der CPU auf den FPGA verlagert werden.

Die Modellgenauigkeit (grössere Modelle sind in der Regel genauer), die Inferenzgeschwindigkeit (kleinere Modelle sind in der Regel schneller) und der Stromverbrauch (grössere, leistungsfähigere oder kleinere, stromsparendere DPU) stehen in Konkurrenz zueinander und müssen gemeinsam optimiert werden. Daher planen wir ausführlichere Leistungsvergleiche, um einen besseren Einblick in das Systemverhalten zu erhalten.

Letztendlich setzt die Betriebsumgebung im Weltraum allen Anwendungen des maschinellen Lernens starre Grenzen, z. B. in Bezug auf den Stromverbrauch oder die Bildauflösung, und es müssen Kompromisse eingegangen werden. Solche Kompromisse müssen nicht unbedingt technischer Natur sein, sondern können auch betrieblicher Natur sein, z. B. die Lockerung von Echtzeitanforderungen, um eine verzögerte Batch-Verarbeitung von Bildern zu ermöglichen.

Metrik	FM0.1	QM0.1	AM0.1	FM0.5	QM0.5	AM0.5	FL0.5	AL0.5
Durschnittlich. Precision @IoU=0.50:0.95	0.6	0.33	0.57	0.21	0.04	0.11	0.24	0.1
Durschnittlich. Recall @IoU=0.50:0.95	0.66	0.48	0.64	0.31	0.13	0.2	0.33	0.19
Durschnittlich. Dauer/Bild [s]			0.78/0.21			0.49/0.13		1.03/0.27
Leerlaufleistung [s]			8/8.1			8/8.1		8/8.1
Durchschnittlich. Leistung [W]			8.8/10.5			8.8/10.5		8.8/10.6
Spitzenleistung [W]			12/19.9			12/16.3		12/17

Tab. 2: Vorläufige Benchmark-Ergebnisse.

Referenzen:

Di Mascio, S., Menicucci, A., Gill, E., Furano, G., Monteleone, C., 2021. On-board decision making in space with deep neural networks and risc-v vector processors. Journal of Aerospace Information Systems 18, 553–570.

Furano, G., Meoni, G., Dunne, A., Moloney, D., Ferlet-Cavrois, V., Tavoularis, A., Byrne, J., Buckley, L., Psarakis, M., Voss, K.-O., others, 2020a. Towards the use of artificial intelligence on the edge in space systems: Challenges and opportunities. IEEE Aerospace and Electronic Systems Magazine 35, 44–56.

Furano, G., Tavoularis, A., Rovatti, M., 2020b. AI in space: Applications examples and challenges, in: 2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT). pp. 1–6.

George, A. D., Wilson, C. M., 2018. Onboard processing with hybrid and reconfigurable computing on small satellites. Proceedings of the IEEE 106, 458–470.

Krucker, S., Hurford, G. J., Grimm, O., Kögl, S., Gröbelbauer, H.-P., Etesi, L., Casadei, D., Csillaghy, A., Benz, A. O., Arnold, N. G., others, 2020. The spectrometer/telescope for imaging X-rays (STIX). Astron Astrophys 642, A15.

Mateo-Garcia, G., Veitch-Michaelis, J., Smith, L., Oprea, S. V., Schumann, G., Gal, Y., Baydin, A.G., Backes, D., 2021. Towards global flood mapping onboard low cost satellites with machine learning. Sci Rep 11, 1–12.

Verma, G., Gupta, A., Bansal, S., Dhiman, H., 2022. Monitoring Maritime Traffic with Ship Detection via YOLOv4, in: 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP). pp. 1–7.

Ivo Nussbaumer
Filip Schramka
Laszlo Etesi
Ateleris GmbH
Badenerstrasse 13
CH-5200 Brugg
ivo.nussbaumer@ateleris.ch

Robin Müller
FHNW Institut für Sensorik und Elektronik
Klosterzelgstrasse 2
CH-5210 Windisch

Nicola Venturi
armasuisse W+T
Feuerwerkerstrasse 39
CH-3602 Thun