Gesteigerte Anwendungsvielfalt bei Vision-Systemen

Deep Learning in Embedded-Geräten in der Edge

Bild 1: Embedded-Vision-Kit aus Kameramodul, Processing Board und weiterem Zubehör [Quelle: Basler AG]

Autor:
Martin Cassel,
Redakteur bei Silicon Software

Silicon Software GmbH
https://silicon.software

Eingebettete Bildverarbeitungsgeräte in der Edge, die unabhängig von einer Cloud mit künstlicher Intelligenz ausgestattet sind, bieten eine leistungsstarke Lösung für Anwendungen in der Fertigung, (kollaborativen) Robotik, Medizintechnik, Logistik, Transportdrohnen und Fahrerassistenzsystemen sowie beim autonomen Fahren. Beim Aufbau des Bildverarbeitungssystems sind jedoch Besonderheiten in Hinblick auf Hardware, Prozessoren, Performance und den eingesetzten neuronalen Netzen zu beachten.

Der Markt für eingebettete Vision-Systeme präsentiert sich insgesamt sehr fragmentiert ohne Dominanz großer Player. Dies korreliert mit dem heterogenen Aufbau der Systeme abhängig von der angestrebten Bildverarbeitungsanwendung, was Systemintegratoren immer wieder aufs Neue herausfordert. Sie müssen die verfügbare Hardware, Verarbeitungsressourcen, Speicherart und-größe wie auch Echtzeit- und Geschwindigkeitsanforderungen (Bildfrequenz, Verarbeitung, Bandbreite) berücksichtigen. Welche Auflösung bzw. Bildqualität ist erforderlich, beispielsweise bei der Objektdifferenzierung? Welche Leistungsaufnahme ist maximal erlaubt? Nicht zuletzt spielen auch Baugröße und Kosten des Systems eine wichtige Rolle für potenzielle Kunden. Die Palette eingebetteter Bildverarbeitungssysteme erstreckt sich daher von (smarten) Kameras über Vision-Sensoren bis hin zu Single Board Computern (SBC).

Trainierte Netze

Für Deep-Learning-Anwendungen findet auf den eingebetteten Geräten kein Training des neuronalen Netzes statt, sondern lediglich das Ausführen (Inferenz) des trainierten Netzes. Die Ausführgeschwindigkeit wiederum hängt von den Performanz-Restriktionen ab. Vollständig trainierte Netze sind generell auf eine bestimmte Anwendung wie eine Oberflächeninspektion ausgerichtet, sind aber auf verschiedenen Embedded-Systemen lauffähig. Hierfür werden sie in spezielle Formate umgewandelt, beispielsweise das Open Neural Network Exchange Format (ONNX), Neural Network Exchange Format (NNEF) oder in eine Netzbeschreibung und Datei mit Gewichten (shared weights). Aus der Vielzahl neuronaler Netze stechen die Convolutional Neural Networks (CNN) für Deep Learning hervor, die wegen ihrer hohen Performanz und geringen Leistungsaufnahme aufgrund weniger Gewichte im Vergleich zu anderen Netzen für die meisten Embedded-Anwendungen in Frage kommen.

Um Ressourcen zu sparen

lassen sich CNNs zum einen verschlanken durch Maßnahmen wie Kompression (der Deep-Learning-Algorithmen und Daten) und Pruning (Entfernen von Teilen des Netzes wie mancher Merkmale, Neuronen und Gewichte, wenn sie nur geringe Auswirkung auf das Ergebnis haben). Zum anderen lassen sich CNNs vereinfachen durch Verringern der Rechengenauigkeit auf 8 bit oder gar 4 bit Fixed Point (Quantisierung). Eine weitere Verringerung nehmen Binarized Neural Networks (BNN) vor. Solche Netze arbeiten mit binären Gewichten und reduzieren Fixed-Point-Multiplikationen in den Schichten zu 1 bit Operationen. Sie benötigen geringere Rechenleistung, Leistungsaufnahme und Taktrate, jedoch auf Kosten der Rechengenauigkeit. Diese ist für Embedded-Anwendungen jedoch oftmals zu gering im Vergleich zu CNNs. Der Genauigkeitsverlust wirkt sich dabei je Anwendung unterschiedlich aus, hilft jedoch den Speicherplatzbedarf zu reduzieren. Einen ergänzenden Ansatz bietet Trained Ternary Quantization (TTQ). Dieses Verfahren wandelt Gewichte in ternäre Werte um, die jeweils auf 2 Bits gespeichert sind. Die Gewichte werden in drei Werte quantisiert, die für jede Schicht spezifisch sind: Null zum Abbrechen nutzloser Verbindungen, ein positiver Wert und ein negativer Wert.

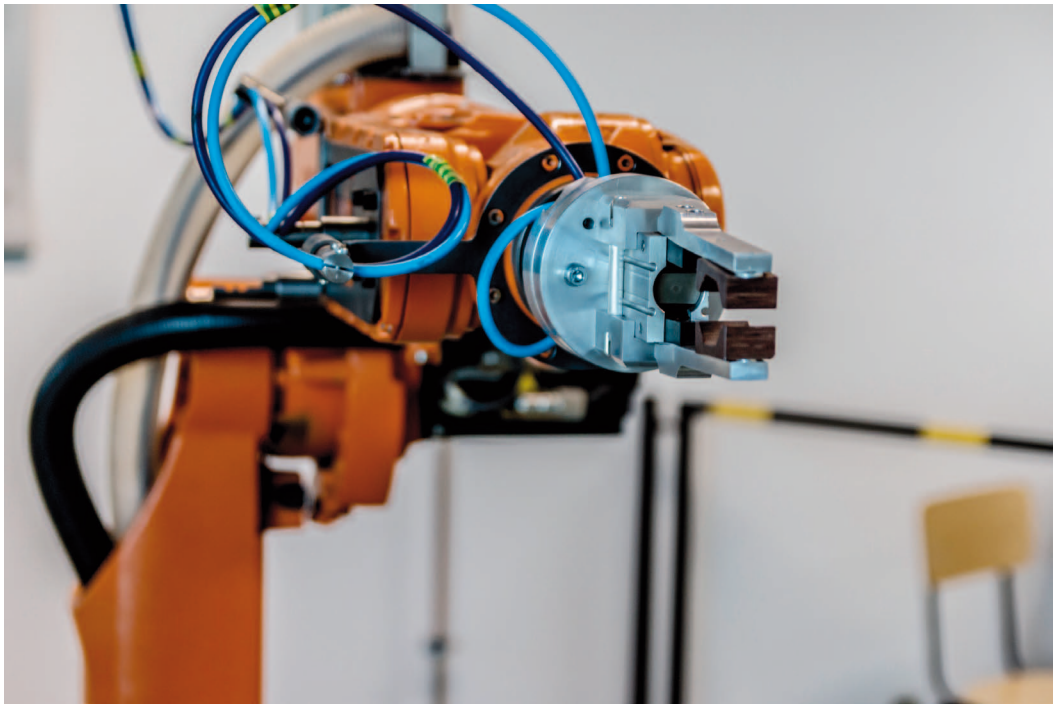

Bild 2: Roboter sind anhand von Deep Learning flexibler an neue Aufgaben anpassbar

Unterm Strich muss für ein Embedded-System jeweils die richtige Balance zwischen der Performanz und einer Verschlankung der Gewichte gefunden werden.

Passende Chipsets je nach Anwendung

CNNs für Deep Learning verarbeiten Bilder, etwa durch Objektdetektion, Muster- oder Anomalienerkennung, und geben ein Ergebnis wie zum Beispiel eine Klassifizierung aus. Eingebettete Systeme hinken jedoch der Leistung von PCs immer noch hinterher, was angesichts der entstehenden großen Datenmengen beim Deep Learning eine Beschränkung darstellt. Umso mehr, als in der Edge eingesetzte Technologien wie Sensordatenfusion oder Datenanalysen per Cognitive Manufacturing den Datendurchsatz zusätzlich erhöhen. Embedded-Systeme mit CNN-Fähigkeiten benötigen daher eine sehr hohe Rechenleistung von 5 bis 50 TOPS (tera-operations per second) und entsprechend große Bandbreiten-Kapazitäten bei außerdem möglichst geringer Leistungsaufnahme von etwa 5 bis 50 Watt. Die Performanz des Systems wird besonders durch die Anforderungen an die Leistungsaufnahme bestimmt und hängt zusätzlich von der Bereitschaft ab, einen bestimmten Preis für das System zu bezahlen.

Je nach Anwendung und dem entsprechenden Datenaufkommen werden unterschiedliche Chipsets verwendet, die als Beschleuniger im Verbund mit der CPU (Central Processing Unit) fungieren. Die CPUs in Embedded-Systemen basieren häufig auf der ARM-Architektur. CPUs alleine weisen jedoch keine parallelen Strukturen auf und haben eine zu geringe Rechenkapazität, weshalb sie häufig auf Unterstützung weiterer Prozessoren für die Inferenz angewiesen sind (z. B. spezialisierte TPU, Tensorflow Processing Unit). GPUs (Graphics Processing Unit) hingegen trumpfen mit einem komfortablen Programmiermodell, massiver Parallelität und hoher Speicherbandbreite, was den Hauptspeicher entlastet. Sie haben jedoch eine recht hohe Wärmeleistung und sind als Embedded-Version (z. B. NVIDIA Jetson, AMD Ryzen embedded) entweder auf eine Reduktion derselben oder auf mehr Rechengeschwindigkeit optimiert. Bei einer versuchsweisen Anwendung zur Gesichtserkennung wurden mit CPUs lediglich 1 bis 4 Gesichter pro Sekunde erkannt, mit neueren embedded GPUs hingegen bis zu 400 Gesichter. Um diese hohen Leistungsunterschiede zu verringern und GPUs stärker herauszufordern, arbeiten CPU-Hersteller bereits an einer Verstärkung der Rechengeschwindigkeit.

Field-programmable Gate Array

Als Alternative mit hohem Potenzial gelten FPGAs (Field-programmable Gate Array), die eine sehr hohe Rechengeschwindigkeit mit wenig Wärmeleistung und geringsten Latenzen verbinden. FPGAs lassen sich wie eine Software mit

Bild 3: Einsatz eingebetteter Vision-Systeme für Fahrerassistenzsysteme (ADAS) und beim autonomen Fahren

überschaubarem Aufwand anpassen, damit sie verschiedene neuronale Netze ausführen. Falls mit der Zeit mehrere neuronale Netze für eine bestimmte Aufgabe benötigt werden, wären somit FPGAs von Vorteil. Sie punkten auch durch ihre lange, industrielle Verfügbarkeit von über 10 Jahren.

Application-specific Integrated Circuit

ASICs (Application-specific Integrated Circuit) werden von Grund auf für Deep-Learning-Beschleunigung designed, beispielsweise mit einer Engine für schnelle Matrix-Multiplikatoren und direkte Faltung (convolution). Die Hersteller von ASICs versprechen hohe Rechenleistung mit geringer Wärmeleistung. ASICs versuchen, den Speicherzugriff zu minimieren, und halten die maximale Datenmenge auf dem Chip, um Verarbeitung und Durchsatz zu beschleunigen. ASICs sind jedoch nur geringfügig programmierbar und daher unflexibel im Einsatz. Sie eignen sich dennoch im industriellen Umfeld in der Edge und lassen sich untereinander kombinieren. Zudem ist die Herstellung individueller ASICs mit einer sehr hohen Investition verbunden.

Geeignete Processing Boards für Deep Learning

Die Kombination aus kleinen Processing Boards und miniaturisierten Kameramodulen bildet ein Embedded-System. Darin sind Systems on Chip (SoC) die zentrale Recheneinheit, die eine CPU als Anwendungsbeschleuniger sowie weitere Prozessoren wie GPU, FPGA oder ein Deep Learning Chipset enthalten. System-on-Modules (SoMs), auch Computer-on-Module (CoM) genannt, enthalten einen SoC, ergänzt um wichtige Komponenten wie Speicher (RAM) und Powermanagement. Sie machen einen SoC damit praktisch nutzbar. Was häufig noch benötigt wird ist ein Carrier Board (Trägerplatine) mit physikalischen Konnektoren für Peripheriegeräte wie Kameras, das Kunden gemäß ihren spezifischen Anforderungen selbst entwickeln können. Bei einem Single Board Computer (SBC) hingegen ist ein SoC zusammen mit den Komponenten bereits von vorneherein auf einer Trägerplatine mit festen Anschlüssen für Peripheriegeräte angebracht. Zusätzlich wird ein Embedded-Betriebssystem benötigt, das die Einzelkomponenten steuert.

Voraussetzungen in der Robotik

Fest verankerte Industrieroboter oder mobile (kollaborative) Roboter nehmen anhand von im Roboterarm oder Greifer integrierten (3D) Vision-Sensoren ihre Umgebung wahr, klassifizieren Objekte, detektieren Anomalien, kollaborieren unfallfrei mit anderen Menschen und Maschinen, positionieren Werkstücke und montieren Geräte. Sie werden in vielen Lebensbereichen und der Industrie eingesetzt und sind anhand von Deep Learning mit tiefen neuronalen Netzen flexibler an ihre unterschiedlichen Aufgaben anpassbar. Beispielsweise lassen sich durch Transfer Learning vortrainierte neuronale Netze zeit- und kosteneffizient gleich für mehrere Robotik-Anwendungen einsetzen. Durch (Deep) Reinforcement Learning wiederum wird das Netz an neue Umgebungen adaptiert, indem ein Roboter sein Verhalten durch Anreize perfektioniert. Für Belohnungen oder Bestrafungen (negative Belohnungen) werden die Parameter angepasst, damit der Roboter gute Aktionen wiederholt. Dies befähigt ihn, neue Arbeitsschritte in relativ kurzer Zeit dazuzulernen, auch für schwierige und variable Inspektionsumfelder wie die genannten Werkstückpositionierung und Gerätemontage. Das jeweils veränderte Roboterverhalten wäre mit herkömmlichen Algorithmen nur mit hohem Aufwand zu programmieren.

Als geeignete Zielanwendungen gelten unter anderem Verpacken und Palettieren, Maschinenbestückung und -entladung, Pick&Place-Anwendungen, Bin Picking und Qualitätsprüfungen im Automobilbau, Elektronikfertigung, Landwirtschaft (Präzisionslandwirtschaft und Automatisierung von Arbeitsschritten) und Medizintechnik (smarte Geräte, Früherkennung von Krankheiten, OP-Assistenzsysteme). Beim Bin Picking sind Roboter anhand von Deep Learning in der Lage, Werkstücke unabhängig von Position und Ausrichtung, also auch wenn sie gewinkelt oder überdeckend liegen, zu erkennen und zu greifen. Der Roboter orientiert sich dabei selbstständig und findet ihm bekannte Teile auf Basis von Inputdaten wie normalen Bildern, Stereo- oder 3D-Bildern.◀

Fazit

Die Verlagerung von Deep Learning in Embedded-Geräte in der Edge hat gegenüber Cloud-Lösungen den Vorteil von reduzierter Netzwerkbelastung und geringeren Latenzen, was mehr Anwendungen ermöglicht.

Hinzu kommt, dass die Verarbeitung von geschützten Bildern in der Cloud ein Sicherheitsrisiko darstellen kann. Die Datenerzeugung und -nutzung findet an einer Stelle statt, was die Gesamtanlageneffektivität erhöht. Gestiegene Rechenressourcen lassen sich anhand immer leistungsfähigerer kleiner Netze, neuer Prozessoren für Embedded-Anwendungen und verbesserter Verfahren wie Kompression, Pruning und Quantisierung ausgleichen.

Geeignete Chipsets und Processing Boards für Deep Learning variieren stark je nach Anwendung und müssen zusammen mit der Software exakt auf diese abgestimmt werden. Durch Techniken wie Transfer und Reinforcement Learning sind die verwendeten Netze und damit die gesamte Anwendung schnell anpassbar, wodurch sich beispielsweise Roboter flexibel einsetzen lassen.

Auch über die Robotik hinaus wird Deep Learning auf eingebetteten Edge-Geräten eine tragende Rolle spielen. Welche Komponenten und Netze sich für welche Anwendung eignen – dies wird Systemintegratoren stetig herausfordern.