Personenzählung on the Edge mit neuronalen Netzen als Teil von Gebäudesensorik

Robert Hecker

Hochschule Darmstadt, Fachbereiche Mathematik und Naturwissenschaften & Informatik
Referentin: Prof. Dr. Elke Hergenröther - Korreferentin: Prof. Dr. Antje Jahn

Hintergrund und Motivation

Nach Einschätzungen des Umweltbundesamtes entfiel im Jahr 2021 mehr als ein Viertel des gesamten Energieverbrauchs in Deutschland auf die Beheizung und Kühlung von Gebäuden [4]. Ineffizientes Beheizen von ungenutzten Flächen verschwendet Energie, zu spätes Lüften führt eventuell zu Beschwerden bei Anwesenden. Um diese Probleme zu vermeiden, werden Informationen über die tatsächliche Belegung einzelner Räume benötigt. Klassische Sensoren wie Bewegungsmelder können keine Aussage über die Anzahl der anwesenden Personen geben, Messungen der Luftqualität leiden unter einer zu hohen Latenz. Eine einfache Möglichkeit zum Erkennen der Auslastung besteht in der Verwendung von Kameras zum Erfassen eines Raumes und dem Zählen von Personen durch Verfahren wie Convolutional Neural Networks. Kamerasysteme, bei denen Bilder zur Zählung an eine Zentrale übertragen werden, stellen jedoch ein mögliches Sicherheitsrisiko dar; optimalerweise sollte die Zählung direkt an der Kamera erfolgen. Gleichzeitig verfügen kleinste eingebettete Systeme durch den technologischen Fortschritt heutzutage über eine Leistungsfähigkeit, welche möglicherweise zur Ausführung neuronaler Netze zur Personenzählung ausreichend sind. In dieser Arbeit wurde diese Problemstellung der Ausführung auf eingebetteten Systemen zur Personenzählung untersucht.

Personenzählung durch Objektdetektoren

Die Zählung von Personen auf Kamerabildern baut auf dem Fachgebiet der Objektdetektion auf. Objektdetektoren sind Verfahren, welche auf einem Eingabebild verschiedene Objekte lokalisieren und klassifizieren können. Eine Personenzählung ist somit nur eine Objektdetektion von Objekten der Klasse “Mensch” mit anschließender Zählung aller unterschiedlichen Objekte. Standardmäßig werden hierzu Convolutional Neural Networks eingesetzt, welche für jedes erkannte Objekt eine Bounding Box zur Lokalisierung und eine Wahrscheinlichkeitsverteilung zur Klassifikation ausgeben, wie in in der folgenden Abbildung erkennbar.

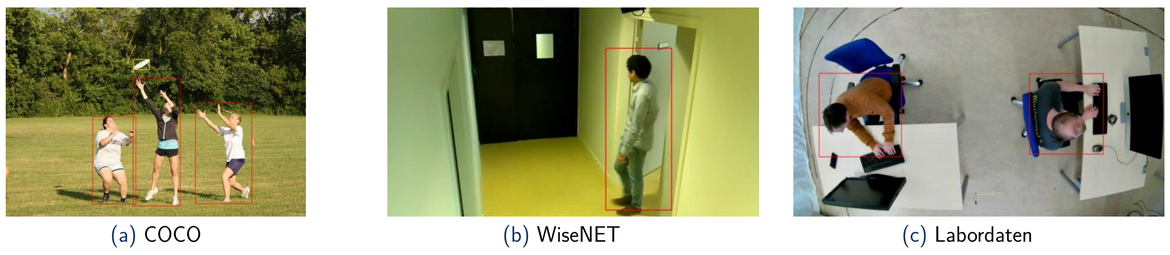
Konzept

Die Ausführung von CNNs fordert sowohl Speicher– als auch Rechenkapazitäten eines Computersystems. Moderne CNNs zur Objektdetektion besitzen Hunderttausende bis Millionen an Parametern, welche im Rahmen der Berechnungen teilweise hunderte bis tausende Male für verschiedene Teile eines Eingabebildes wiederverwendet werden. Aus dieser Größe können für die genannten Aspekte folgende Anforderungen formuliert werden:

- *Prozessorleistung*: Durch die Wiederverwendung von Parametern müssen hunderte Millionen bis wenige Milliarden an Berechnungen pro Bild vorgenommen werden. Die Hardware muss diese Anzahl an Berechnungen somit in annehmbarer Zeit durchführen, somit sind Prozessorgeschwindigkeiten von hunderten MHz bis wenigen GHz notwendig.
- *Festwertspeicher*: jeder Parameter belegt, je nach Datentyp, ein oder mehrere Bytes. Flash-Speicher o. Ä. muss somit $>$ 100 kByte bzw. $>$ 1 MByte umfassen, je nach tatsächlicher Größe des Netzes und Datentyp der Parameter.
- *Arbeitsspeicher*: die Speicherung der Zwischenergebnisse belegt einen Bruchteil des benötigten Festwertspeichers im RAM. Besonders speicherintensiv sind jedoch auch die unkomprimierten Eingabebilder, welche pro Bild mehrere hundert Kilobyte Speicher benötigen.

Als eine Netzarchitektur, welche auf die Besonderheiten von eingebetteten Geräten zugeschnitten ist, wurde die YOLOv5$_n$-Architektur identifiziert, welche trotz einer geringen Anzahl an Modellparametern ausreichende Detektionsqualität verspricht. Um eigene Versionen eines YOLOv5-Detektors zu trainieren, mussten Trainingsdatensätze gefunden werden, welche Personen abbilden. Hierfür wurden mehrere Datensätze identifiziert:

- Microsoft COCO [1]: allgemeiner Benchmark-Datensatz zur Objekterkennung. Beinhaltet auf einem Großteil der Bilder Personen, welche in sehr vielen verschiedenen Perspektiven abgebildet sind. Nur ein geringer Anteil der Bilder entspricht einer Top-Down-Perspektive wie sie von einer deckenmontierten Kamera zu erwarten wäre.
- WiseNET-Datensatz [2]: Videoaufnahmen aus einem Bürogebäude als Teil einer Veröffentlichung von Marroquin et al. Einheitliche Perspektive und geringere Varianz, aber nicht repräsentativ für den angestrebten Anwendungsfall.
- Labordatensatz [3]: im Rahmen einer vorhergegangenen Masterarbeit erstellter Datensatz durch eine deckenmontierte Kamera. Sehr geringe Größe, Perspektive ist für den Anwendungsfall repräsentativ.



(a) COCO (b) WiseNET (c) Labordaten

Abbildung: Beispielbilder aus den Datensätzen mit eingezeichneten Ziel-Bounding-Boxen.

Es wurden auf dem COCO– und dem WiseNET-Datensatz jeweils zwei Varianten der YOLOv5$_n$-Architektur trainiert:

- 1class$_n$: Variante von YOLOv5$_n$, welche nur auf die Objektklasse “Person” spezialisiert ist (YOLOv5 erkennt in seiner vortrainierten Form 80 verschiedene Klassen des COCO-Datensatzes).
- half$_n$: weitere Abwandlung der vorherigen Variante, wobei die Anzahl der Filter in jedem Convolutional-Layer halbiert wurde, um eine Reduktion der Parameter und der “Wissenskapazität” zu bewirken.

Das Training eigener Varianten von YOLOv5$_n$ konnte auf diesen Daten mit vertretbaren Aufwänden umgesetzt werden; die Netze wurden jeweils für 75 Epochen auf Consumer-Hardware innerhalb 2,5 bis 7 Stunden trainiert (je nach Datensatz und Netzvariante). Das Neutrainieren eigener neuronaler Netze zur Objektdetektion ist für kleinste Netze, welche auf eingebetteten Geräten eingesetzt werden sollen, mit vertretbarem Aufwand möglich.

Es wurden mehrere Varianten von YOLOv5$_n$ auf dem COCO– und dem WiseNET-Datensatz trainiert. Der Labordatensatz war nicht umfangreich genug, um ein Training zu ermöglichen. Sie wurden auf den jeweils verbleibenden Datensätzen evaluiert, um die Übertragbarkeit des erlangten Wissens zwischen den Datensätzen zu prüfen. Um die Auswirkung der Bildgröße auf die Qualität der Objektdetektion und der Zählung zu untersuchen, wurden unterschiedliche Bildgrößen zwischen 128×128 und 512×512 Pixeln untersucht.

Als Ergebnis wurde erhalten, dass einfachere Sachverhalte wie auf dem WiseNET-Datensatz bereits mit kleinen Bildauflösungen und kleineren Netzen erkannt werden können. Komplexere Daten wie aus dem COCO-Datensatz benötigen größere Auflösungen und komplexere Netze, auf dem COCO-Datensatz trainierte Modelle konnten in Teilen auf einfachere Datensätze wie den WiseNET-Datensatz übertragen werden, umgekehrt funktionierte dies nicht. Auf den Labordatensatz ließen sich auf COCO trainierte Modelle besser auf die ungewöhnliche Perspektive anwenden. Die Netze neigten in allen Fällen durchgehend dazu, die wahre Anzahl an Personen zu unterschätzen, der MAE schwankte je nach Evaluationsdatensatz zwischen 3,5 und 0,2 Personen.

Konzept

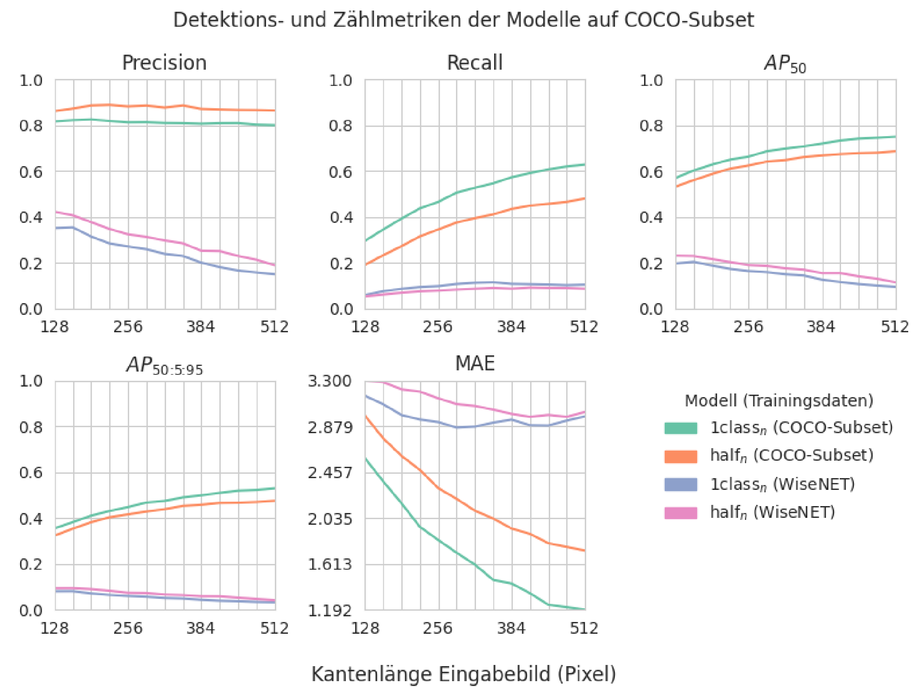


Abbildung: Gemessene Metriken der Modelle auf dem COCO-Datensatz (gefiltert auf Bilder, welche Personen enthielten). Die Metriken Precision, Recall, AP_{50} und $AP_{50:\,5:\,95}$ bewerten die Qualität der Objektdetektion, während der MAE den Zählfehler (in Personen) abbildet. Diese Versuche wurden auf dem WiseNET– und dem Labordatensatz wiederholt (hier nicht abgebildet).

Implementierung auf echter Hardware

Eine Implementierung der erstellten neuronalen Netze wurde auf folgenden Hardwaregeräten vorgenommen:

- ESP32-CAM: Mikrocontroller mit 2-Kern-CPU mit 160 MHz, 4 MB RAM und 4 MB Flash.
- Raspberry Pi Zero 2 W: Einplatinencomputer mit 4-Kern-CPU mit 1 GHz, 512 MB RAM und Anschluss für SD-Karten.
- Luckfox Pico Mini A: Einplatinencomputer mit 1-Kern-CPU mit 1,2 GHz, 64 MB RAM und Anschluss für SD-Karten. Verfügt über eine Neural Processing Unit (NPU) mit bis zu $0{,}5 * 10^{12}$ Operationen pro Sekunde zur Beschleunigung von neuronalen Netzen.

Als Laufzeitumgebungen mussten für jedes Gerät unterschiedliche Implementierungen genutzt werden, da die starken Unterschiede in der Leistung und in den Fähigkeiten der Geräte dies notwendig machten. Für den ESP32 wurde TensorFlow Lite Micro als Laufzeit speziell für Mikrocontroller, für den Pi Zero eine Python-Umgebung und für den Luckfox Pico die herstellerspezifische RKNN-Laufzeit zur Nutzung der NPU eingesetzt. Als besonders aufwändig wurde hierbei die mehrfache Konversion von Modelldateien zwischen Deep-Learning-Frameworks wie PyTorch und TensorFlow und später auch zwischen Dateiformaten speziell für die Laufzeitumgebungen der Geräte festgehalten.
Mit 10 Bildern aus dem COCO-Datensatz wurden Laufzeitmessungen aller erstellten Netze auf allen Geräten vorgenommen.

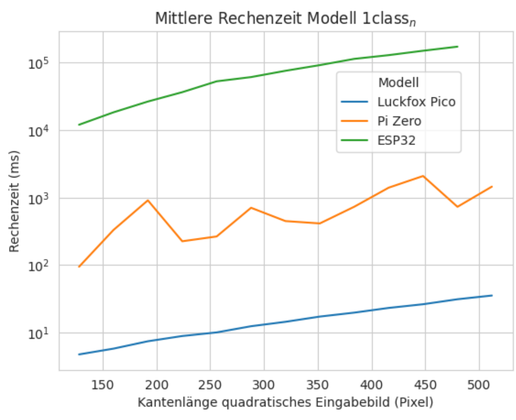


Abbildung: Über zehn Versuche gemittelte Laufzeiten eines ausgewählten Modells auf allen drei Geräten.

Zwischen den Hardwaregeräten konnten starke Unterschiede in der Rechenzeit festgestellt werden. Während der ESP32 ohne Hardwarebeschleunigung mehrere Minuten für größere Bilder benötigte und auf der höchsten Auflösung mangels ausreichendem freien RAM das Modell nicht mehr ausführen konnte, benötigten die Einplatinencomputer für das gleiche Modelle bis zu einem Faktor 1000 weniger Rechenzeit. Bedingt durch die starke Beschleunigung durch die NPU des Luckfox Pico konnte dieser die Bilder innerhalb weniger Millisekunden verarbeiten, sodass er eine Zählung in Echtzeit ermöglichen könnte.

Fazit

Als Fazit dieser Arbeit lassen sich folgende Aussagen festhalten:

- Das Implementieren von neuronalen Netzen zur Objektdetektion auf kleinsten eingebetteten Geräten ist technisch möglich. Es müssen Netzarchitekturen eingesetzt werden, welche auf die besonderen Rahmenbedingungen dieser Geräte zugeschnitten sind. Die Netze erzielen ausreichend genaue Zählungen, um im Rahmen der Gebäudesensorik verwendet zu werden.
- Mangels einer ausreichenden Menge repräsentativer Trainingsdaten bleibt offen, ob die angestrebte Personenzählung “von oben” eventuell durch noch einfachere CNNs bewältigt und auf leistungsschwächeren Geräten umgesetzt werden könnte.
- Einplatinencomputer mit Hardwarebeschleunigung ermöglichen eine Personendetektion in Echtzeit, während Mikrocontroller durch ihre geringere Leistung mehrere Minuten pro Bild benötigen.

Quellen

[1] Tsung-Yi Lin u. a. “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. Hrsg. von David Fleet u. a. Cham: Springer International Publishing, 2014, S. 740–755.

[2] Roberto Marroquin, Julien Dubois und Christophe Nicolle. “WiseNET: An indoor multi-camera multi-space dataset with contextual information and annotations for people detection and tracking”. In: *Data in Brief* 27 (Dez. 2019), S. 104654. ISSN: 2352-3409. DOI: 10.1016/j.dib.2019.104654.

[3] Oskar Rudolf. “Evaluation vortrainierter neuronaler Netzwerke zur Anwendung auf die autonome Zählung von Personen mit Objektdetektion”. Masterarbeit. Hochschule Darmstadt, Mai 2024.

[4] Umweltbundesamt. *Indikator: Energieverbrauch für Gebäude*. https://www.umweltbundesamt.de/daten/umweltindikatoren/indikator-energieverbrauch-fuer-gebaeude. 2024.