University of Applied Sciences and Arts Northwestern Switzerland FHNW

Master of Science in Engineering

Edge ML Camera for Citizen Science

Edge ML Kamera für Citizen Science

Author:	Andri Wild
Supervisor:	Prof. Thomas Amberg
Advisors:	Prof. Thomas Amberg
Start Date:	01.01.2024
Submission Date:	01.01.2024

I confirm that this Master of Science in Engineering's thesis is my own work and I have documented all sources and material used.

Munich, 01.01.2024 Andri Wild

Acknowledgements

First, I'd like to thank coffee for fueling my brain cells and making this thesis possible.

A big shoutout to my advisor for your patience and for not laughing (too hard) at my wild ideas.

To my family, your snack supplies and constant reminders to "just finish it already" were invaluable.

Finally, to my pet, your keyboard sit-ins ensured I took breaks, whether I wanted to or not.

Abstract

Note:

1. **paragraph:** What is the motivation of your thesis? Why is it interesting from a scientific point of view? Which main problem do you like to solve?
2. **paragraph:** What is the purpose of the document? What is the main content, the main contribution?
3. **paragraph:** What is your methodology? How do you proceed?

Zusammenfassung

Note: Insert the German translation of the English abstract here.

Inhaltsverzeichnis

1 Einleitung

1.1 Ausgangslage

- Mitwelten Projekt (Ziel, Kontext)
- Analyse Bestäuber (Kurzbeschreibung der Mitwelten Pipeline wie sie eingesetzt war)

Das SNF Projekt „Mitwelten - Medienökologische Infrastrukturen für Biodiversität“ hat IoT-Technologie in urbanen Naturgebieten installiert, um dort vorhandene Tiere, Pflanzen und Umweltbedingungen genauer zu untersuchen. Ein Teilprojekt beschäftigte sich mit der Erkennung von Bestäuber-Arten auf Blumenblüten mittels Raspberry Pi Kameras. Die Machine-Learning-basierte Detektion und Kategorisierung geschieht zurzeit in einem zentralen Cloud-Backend.

Diese Architekturlösung bringt folgende Konsequenzen mit sich: Zum Einen muss der Betrieb eines Backends sichergestellt sein. Dies erfordert einen gewissen Wartungsaufwand und insbesondere ein erhöhtes technisches Verständnis für die Projektumsetzung. Für Forschende bedeutet dies, dass zusätzliches IT-Fachpersonal für den Aufbau und Betrieb der IT-Infrastruktur benötigt wird. Zum Anderen werden die aufgenommenen Bilder zur Analyse über Mobilfunknetze versendet, wodurch die Betriebskosten erhöht werden und das Einsatzgebiet auf dessen Abdeckung begrenzt. Beide der genannten Aspekte sind mit zusätzlichen Kosten verbunden und erhöhen die Gesamtkomplexität des Projekts.

1.2 Zielsetzung

- Fragestellung / Hypothese

Das Ziel ist die Portierung der ML-Pipeline des SNF Projekts Mitwelten auf eine Edge ML Kamera, um Citizen Science Projekte zu ermöglichen:
Dank der fortschreitenden Entwicklung von Edge-Computing-Hardware sind auf Machine Learning spezialisierte Geräte zu einem vergleichbaren Preis wie konventionelle Edge Computer erhältlich. Dies wirft die Frage auf, ob durch diesen technologischen Fortschritt die IT-Infrastruktur eines Edge-Kamera basierenden Systems so weit reduziert werden kann, dass die selbe Funktionalität ohne zentrales Backend umsetzbar ist. Der Einsatz von Edge-basierten Systemen könnte den Zugang zu Machine Learning Projekten für Forschende erheblich erleichtern, da ein System ohne zentrale Abhängigkeit mit weniger

Aufwand betreibbar ist. Diese stand-alone Lösung öffnet auch die Türen für Citizen Science Projekte. Einzelpersonen oder Gemeinschaften können so eigenständig und unabhängig Forschungsprojekte durchführen.

- Welche Hardware und Frameworks gibt es, um Machine Learning (ML) dezentral, im Feld, mittels Edge Computing umzusetzen?
- Was sind Anforderungen an eine ML-Kamera für typische Citizen Science Anwendungen im Bereich Biodiversitätsmonitoring?
- Wie sieht eine konkrete Edge ML Kamera aus, für Monitoring von Biodiversität, wie im Projekt SNF Mitwelten angewendet?

1.3 Stand der Forschung

(Viele neue Hardware im Bereich ML on Edge)

1.4 Abgrenzung

Kein ML Training

1.5 Aufbau des Berichts

Der Bericht besteht im wesentlichen aus vier Teilen. Im ersten teill werden die grundlagen vermittelt, welche relevant sind für das Verständnis des Berichts. Anschliessend folgt die Analyse, welche die Zielgruppen und Anforderungen definiert. In der Umsetzung wird die konkrete Umsetzung des Projekts beschrieben und abschliessend wird die Evaluation der Resultate vorgenommen.

2 Grundlagen

(Dinge die man auf wikipedia lesen kann, was ist schon hier, die Dinge, die ich nachher brauche)

2.1 Citizen Science

- Was ist das?
- Motivation (Warum machen Leute mit?)
- Warum ist es wichtig / notwendig?

Eine treffende Definition von Citizen Science liefert das Grünbuch Citizen Science Strategie 2020 für Deutschland:

> „Citizen Science beschreibt die Beteiligung von Personen an wissenschaftlichen Prozessen, die nicht in diesem Wissenschaftsbereich institutionell gebunden sind. Dabei kann die Beteiligung in der kurzzeitigen Erhebung von Daten bis hin zu einem intensiven Einsatz von Freizeit bestehen, um sich gemeinsam mit Wissenschaftlerinnen bzw. Wissenschaftlern und/oder anderen Ehrenamtlichen in ein Forschungsthema zu vertiefen. Obwohl viele ehrenamtliche Forscherinnen und Forscher eine akademische Ausbildung aufweisen, ist dies keine Voraussetzung für die Teilnahme an Forschungsprojekten. Wichtig ist allerdings die Einhaltung wissenschaftlicher Standards, wozu vor allem Transparenz im Hinblick auf die Methodik der Datenerhebung und die öffentliche Diskussion der Ergebnisse gehören."
>
> — A. Bonn *u. a.* [1]

Bürgerinnen und Bürger können also einen Beitrag zur Wissenschaft leisten, indem sie Daten sammeln, analysieren und interpretieren. Dies kann von der einfachen Datenerhebung bis hin zur intensiven Auseinandersetzung mit einem Forschungsthema reichen. Die Beteiligung an Citizen Science Projekten ist nicht an eine akademische Ausbildung gebunden, jedoch ist die Einhaltung wissenschaftlicher Standards unabdingbar. Es stellt sich die Frage, worin die motivation liegt sich freiwillig solchen Projekten zu widmen. Ein treibender Faktor kann der Wissenszuwachs für ein bestimmtes Thema sein. Die Auseinandersetzung in einem Bereich, für den man sich interessiert. Das Buch The Science of Citizen Science [2, p 283] diskutiert die verschiedenen Aspekte des Lernens in einem Citizen Science Projekt.

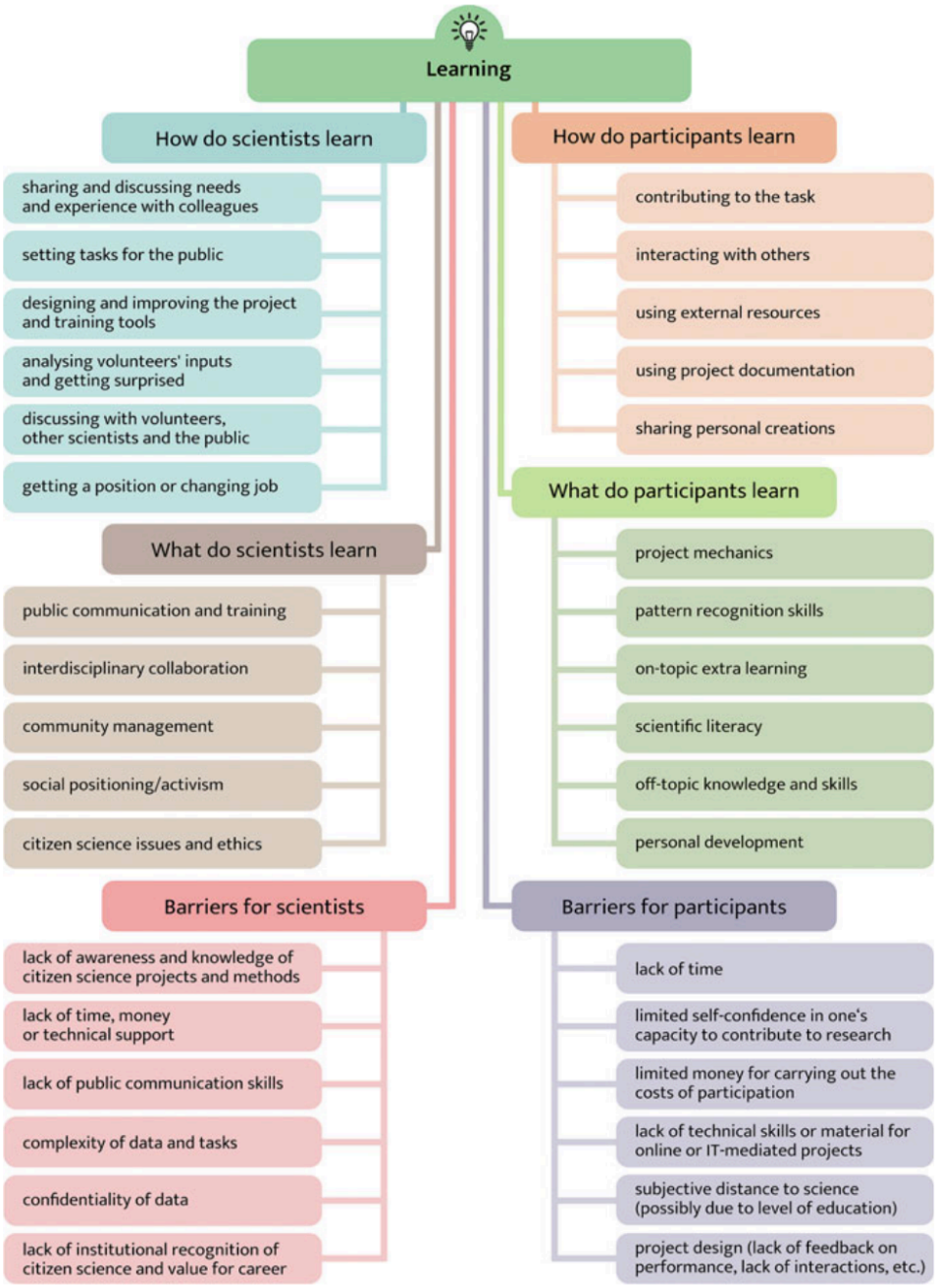

Abbildung 1: Erweiterte thematische Karte zum Lernen von Freiwilligen [2, p 300]

Aus der illustration geht hervor, dass verschiedenste Aspekte die Bürgerinnen und Bürger zu einer aktiven Beteiligung motivieren kann. Nebst den Fachlichen sind auch Soziale Aspekte von Bedeutung. Auf der anderen Seite provitieren auch die Projekt initianten. Die Organiation und Zusammenarbeit mit Aussenstehenden kann eine spannende Aufgabe sein.

2.2 Biodiversität Monitoring

(Timeo's Arbeit zitieren - nicht das selbe machen)

- Sinn und Zweck
- Mögliche Anwendungszwecke / Szenarien / Perspektiven
- Mitwelten Pipeline (Von der Kamera zu den Daten)

Biodiversitäts Monitoring befasst sich grundsätzlich mit dem Beobachten und Analysieren von Tier - und Pflanzenwelt.

> „Die biologische Vielfalt bildet eine Lebensgrundlage der Schweiz. Deshalb ist es wichtig, ihren Zustand und ihre Entwicklung zu kennen."
>
> — bdm [3]

Aus diesem Grund ist es von grosser Bedeutung die Umwelt zu beobachten und Veränderungen festzustellen. Um diese Arbeit zu vereinfachen, sind automatisierte Lösungen für ein Monitoring gefragt.

Ein System zur automatisierten Datenerfassung wurde im Rahmen des Projekts Mitwelten entwickelt [4], [5]. Hierfür wurden verschiedene Sensoren zu einem IoT-Toolkit kombiniert. Dieses Toolkit ermöglicht es, Daten von dezentralen Systemen zu erfassen und an ein zentrales Backend weiterzuleiten.

Das IoT-Toolkit umfasst unter anderem eine Kamera, die in regelmässigen Abständen Fotos von Blumentöpfen aufnimmt. Im Rahmen des Projekts wurden so insgesamt 1,5 Millionen Bilder generiert. Abbildung Abbildung 2 zeigt eine im Feld aufgestellte Kamera.

Abbildung 2: Pollinator Kamera im Feld

Die enorme Menge an Bilddaten lässt sich nicht manuell analysieren. Deshalb entwickelte das Projekt-Team eine Machine-Learning-Pipeline, die Bestäuber automatisch erkennt. Die technische Architektur des Systems verbindet die Kamera mit einem leistungsstarken Backend. Die Kamera, bestehend aus einem Raspberry Pi und einer angeschlossenen Kameraeinheit, erfasst die Bilder und überträgt diese über einen Access Point an das Backend. Auf dem Server führt eine Machine-Learning-Pipeline die Analyse der empfangenen Bilder durch, um Bestäuber zu erkennen. Im Kontext dieser Entwicklung wurden ebenfalls alternative Architekturen untersucht. Under anderem ein System, auf dem die Machine-Learning-Pipeline auf einem Raspberry Pi 3 ausgeführt wurde. Aufgrund der zu langen Analysezeiten wurde jedoch ein Server basierter Ansatz gewählt.

2.3 Machine Learning Grundlagen

- Frameworks (Unterschiede, Vor- und Nachteile)
- Anatomie eines Modells
  - Architektur
    - welche parameter eines modells werden beider Erstellung definiert
      - Input output tensor
  - Art: typen von ML modellen (object detection, segmentation, pose)
  - welche messdaten gibt es für object detection (iou, mAP, perfomance / inferentzeit)
  - training eines models (sehr oberflächlich)
  - infernz, was ist das ?

2.3.1 ML Frameworks

Ein Machine-Learning-Framework bildet die Grundlage für die Entwicklung, das Training und die Bereitstellung von Modellen. Es bietet Werkzeuge und Bibliotheken, die den gesamten ML-Workflow unterstützen. Im Gegensatz dazu ist ein Machine-Learning-Modell das konkrete Ergebnis des Trainingsprozesses, das spezifische Aufgaben wie Klassifikation oder Objekterkennung ausführt. Während ein Framework die Infrastruktur bereitstellt, ist das Modell die fertige Anwendung innerhalb dieser Infrastruktur.

2.3.1.1 TensorFlow

TensorFlow [6] wurde ursprünglich von Google entwickelt und ist inzwischen ein Open Source Projekt. Das Framework ist eher für Systeme in Produktion gedacht.

- Vorteile: Weit verbreitet mit grosser Community und umfangreicher Dokumentation. Stellt viele Features zu Verfügung.
- Nachteile: Steile Lernkurve, insbesondere für Einsteiger. Komplex durch die vielen Features.

2.3.1.2 TensorFlow Lite

- Vorteile: Speziell für den Betrieb auf ressourcenbeschränkten Geräten optimiert. Reduzierte Speicher- und Rechenanforderungen für Echtzeit-Anwendungen.
- Nachteile: Eingeschränkte Unterstützung für komplexe Modelle, Quantisierung erforderlich.

2.3.1.3 PyTorch

PyTorch [7] ist ebenfalls Open Source und hat seinen Ursprung im Forschungsteam für künstliche Intelligenz von Facebook. Das Framework eignet sich für Experimente kann aber auch für Systeme in Produktion eingesetzt werden. ChatGPT von OpenAI arbeitet im Hintergrund mit dem Pytorch Framework [8].

- Vorteile: Hat eine gute Community und eine ausführliche Dokumentation. Viele Features und viele Tutorials.
- Nachteile: Für Einsteiger

2.3.1.4 ONNX

ONNX (Open Neural Network Exchange) [9] ist ein generalisiertes Format von Deep-Learning Modellen. Dies ermöglicht den Austausch von Modellen zwischen verschiedenen Frameworks. ONNX wird von Microsoft, Amazon, Facebook und weiteren Partners als Open-Source Projekt entwickelt.

- Vorteile: Austauschformat, das Modelle zwischen verschiedenen Frameworks kompatibel macht. Optimierte Laufzeitumgebungen, die speziell für Edge-Geräte geeignet sind.
- Nachteile: Begrenzte Funktionalität für Modelltraining, primär für den Einsatz trainierter Modelle gedacht. Kleinere Community im Vergleich zu TensorFlow und PyTorch.

2.3.1.5 NCNN

NCNN [10] ist eine high performance computing platform für mobile Endgeräte. Dieses Framework wird of auf Smartphone verwendet, weil es optimiert für Geräte mit beschränkter Rechenleistung ist.

- Vorteile: Schnelle Inferenzzeiten auf der CPU.

- Nachteile: Im Vergleich zu anderen Frameworks weniger ausführlich dokumentiert.

2.3.1.6 Hailo

2.3.1.7 Ultralytics

Das Ultralytics [11] Framework ist ein auf PyTorch basierendes Open-Source-Framework, das vor allem für die Entwicklung von Modellen zur Objekterkennung bekannt ist. Es wurde durch die Implementierung von YOLOv5 (You Only Look Once) populär und bietet eine benutzerfreundliche Umgebung für das Training und die Bereitstellung von Objekterkennungsmodellen. Inzwischen sind neuere Implementierungen der YOLO Familie verfügbar und mit Ultralytics anwendbar.

- Vorteile: Die Verwendung des Framework ist sehr einfach und dadurch geeignet für Einsteiger. Modelle können in andere Formate exportiert werden. Inferenzen werden für PyTorch Modelle angeboten.
- Nachteile: Die Flexibilität im Vergleich zu anderen Frameworks ist eingeschränkt.

2.3.1.8 Trade-Off's

- Flexibilität vs. Spezialisierung: Frameworks wie TensorFlow und PyTorch bieten umfassende Funktionen, benötigen jedoch mehr Ressourcen. Edge-spezifische Frameworks sind dagegen ressourcenschonender, aber eingeschränkter in ihrer Funktionalität.
- Einsatzgebiet: Die Wahl des Frameworks sollte sich an den spezifischen Anforderungen orientieren, wie etwa der Zielplattform (Cloud vs. Edge), der Modellkomplexität und der Verfügbarkeit von Ressourcen.

2.3.2 Anatomie von Machine Learning Modellen

Ein Machine-Learning-Modell besteht aus einer Architektur, die dessen Aufbau und Funktionsweise definiert, und einem Satz von Parametern, die während des Trainings optimiert werden. Die Architektur eines Modells legt grundlegende Eigenschaften fest, wie die Struktur der Neuronen und Layer, die Verbindungen zwischen ihnen sowie Eingabe- und Ausgabetensoren. Die Eingabetensoren bestimmen, welche Form und Dimensionen die Daten haben müssen (z. B. die Größe von Bildern), während die Ausgabetensoren das Ergebnis des Modells beschreiben, etwa Klassenetiketten oder Koordinaten für erkannte Objekte.

Es gibt verschiedene Typen von Machine-Learning-Modellen, die je nach Anwendungsfall eingesetzt werden. Zu den wichtigsten gehören Modelle für Objekterkennung, die Objekte in einem Bild lokalisieren und klassifizieren, Bildsegmentierung, die jedem

Pixel eines Bildes eine Klasse zuordnet, und Posenschätzung, die die Körperhaltung von Personen oder Tieren analysiert. Für die Objekterkennung werden verschiedene Metriken verwendet, um die Modellleistung zu bewerten. Dazu zählen die Intersection over Union (IoU), die angibt, wie gut vorhergesagte und tatsächliche Begrenzungsrahmen übereinstimmen, und die Mean Average Precision (mAP), die die Präzision über verschiedene Schwellenwerte hinweg aggregiert. Zusätzlich spielen Leistungskennzahlen wie die Inferenzzeit eine wichtige Rolle, insbesondere bei Echtzeitanwendungen.

Das Training eines Modells umfasst die Anpassung der Parameter auf Basis eines großen Datensatzes, um Muster und Zusammenhänge zu lernen. Dabei kommen Optimierungsalgorithmen wie Gradient Descent zum Einsatz, die das Modell iterativ verbessern. Nach dem Training wird das Modell in der Inferenzphase genutzt. Inferenz bezeichnet den Prozess, bei dem ein trainiertes Modell neue Eingaben verarbeitet und Vorhersagen generiert.

2.4 Edge Computing

(vllt auch in der AUsgangslage erwähnen)

- Architektur (Shift von Cloud zu Edge)
- Konsequenzen und Vorteile (Latenz, Kosten, Privacy)
- Herausforderungen (Energie, Rechenleistung)

Edge Computing und Edge Machine Learning markieren einen Paradigmenwechsel in der Systemarchitektur, indem die Datenverarbeitung von zentralen Backend Server oder Cloud-Systemen hin zu Geräten am Rand des Netzwerks (Edge) verlagert wird. Diese Architektur ermöglicht es, Berechnungen und Analysen direkt vor Ort durchzuführen, anstatt die Daten für die Verarbeitung zu versenden.

Dies bringt mehrere Vorteile mit sich. So ermöglicht die lokale Verarbeitung eine geringere Latenz, was vor allem für Echtzeitanwendungen entscheidend ist. Zudem reduziert sich der Datenverkehr zu einem Backend, was nicht nur die Betriebskosten senkt, sondern auch die Abhängigkeit von einer stabilen Internetverbindung minimiert. Nebst der Datenübertragung ist auch die Komplexität eines Systems von Bedeutung. Eine Systemarchitektur mit zentraler Einheit bedeutet nebst zusätzlichen Technologien auch einen erhöhten Wartungsaufwand, welcher gerade bei grosser Projektdauer nicht zu unterschätzen ist. Ein weiterer wesentlicher Vorteil liegt im Bereich der Datensicherheit.

Sensible Informationen bleiben vor Ort und müssen nicht über Netzwerke übertragen werden, wodurch die Privatsphäre der Nutzer besser geschützt wird.

Allerdings bringt dieser Ansatz auch Herausforderungen mit sich. Edge-Geräte verfügen häufig über eingeschränkte Rechenleistung und Speicherressourcen, was die Ausführung komplexer Machine-Learning-Modelle erschweren kann. Darüber hinaus ist die Energieeffizienz ein zentraler Aspekt, da viele Edge-Geräte batteriebetrieben sind und die Optimierung des Energieverbrauchs entscheidend für den Betrieb sein kann. Schliesslich erfordert die Verwaltung und Skalierung einer Vielzahl von verteilten Geräten ohne zentrale Steuerung zusätzlichen Aufwand.

2.4.1 Edge ML

- Hardware die Edge Computing und ML vereint
  - Coral
  - hailo
  - ai camera
- Pipelines / Datenfluss (herkömmlich, AI-Camera)

Edge Machine Learning kombiniert die Prinzipien des Edge Computing mit den Anforderungen moderner Machine-Learning-Modelle, indem die Rechenleistung direkt auf die Endgeräte verlagert wird. Dieser Ansatz wird durch spezialisierte Hardware ermöglicht, die darauf ausgelegt ist, ML-Modelle effizient und ressourcenschonend auszuführen.

Ein Beispiel für solche Hardware sind USB-Dongles wie der Coral USB Accelerator von Google, der eine dedizierte TPU (Tensor Processing Unit) enthält. Diese Geräte lassen sich leicht in bestehende Systeme integrieren und ermöglichen die schnelle Ausführung von ML-Algorithmen ohne signifikante Anpassungen der Infrastruktur. Für Anwendungen, die mehr Rechenleistung benötigen, gibt es leistungsstärkere Lösungen wie die Hailo-TPU, die über eine PCIe-Schnittstelle mit dem Edge-Gerät verbunden wird. Diese Hardware eignet sich für Szenarien mit komplexeren ML-Modellen, da sie eine deutlich höhere Verarbeitungsrate bietet.

Eine weitere Kategorie sind AI-Kameras, die mit integrierten ML-Chips ausgestattet sind. Diese Kameras können nicht nur Bilder aufnehmen, sondern auch direkt auf der Kamera Inferenzdaten bereitstellen, beispielsweise durch die Erkennung und Klassifikation von Objekten. Dieser Ansatz verlagert die intensiven Berechnungen womit der Edge Computer weniger Ressourcen bereitstellen muss.

3 Analyse

(Was braucht es für mich)

Dieses Kapitel befasst sich mit der Analyse und Bewertung der Möglichkeiten für die Umsetzung einer Edge ML Kamera. Zu diesem Zweck wurden anhand von Experimenten verschiedene Messungen durchgeführt. Für die Analyse dieser Experimente ist ein Dashboard [12] entwickelt worden, um interaktiv verschiedene Komponenten miteinander zu verglichen. Die folgende Abbildung 3 zeigt ein Screenshot der Applikation.

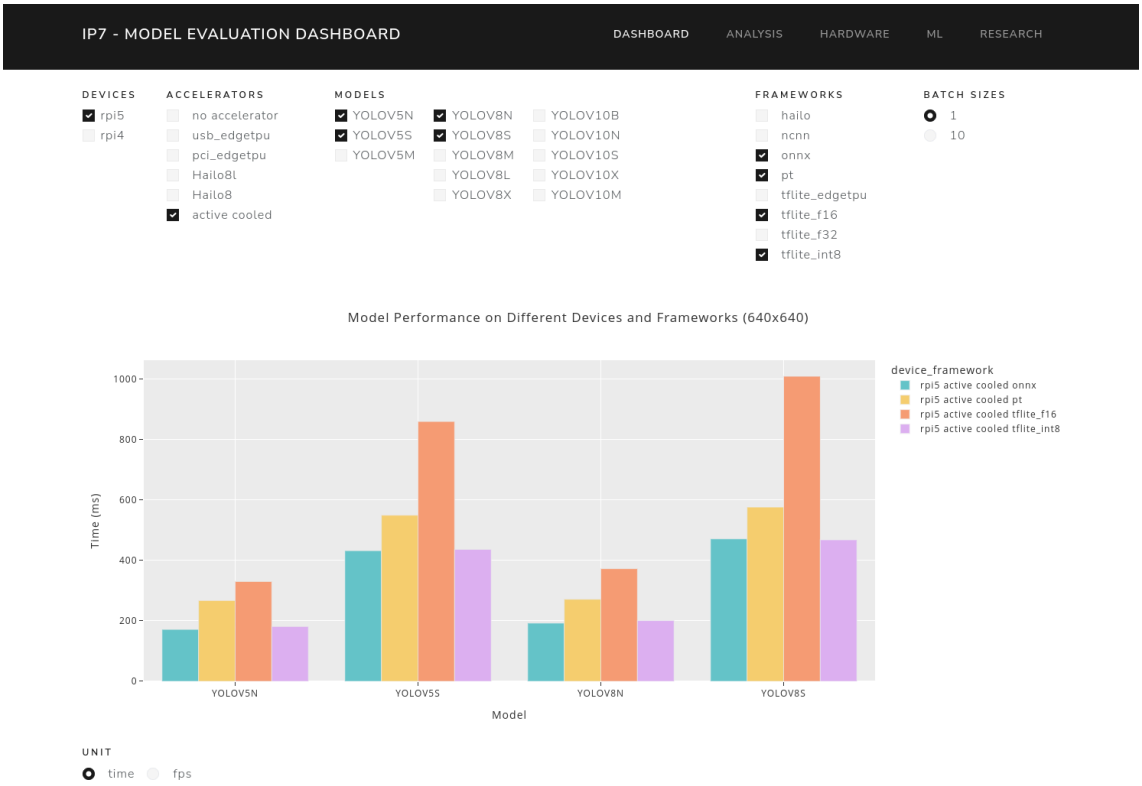

Abbildung 3: ML Exploration Dashboard

3.1 Zielgruppe

- Was müssen Bediener können

Ein Kamera System, welches flexibel für verschiedene Zwecke einsetzbar ist, kann interessant sein für eine vielzahl an Anwendungszwecken und deckt daher auch eine breite Zielgruppe ab:

- Forschende: Wissenschaftlerinnen und Wissenschaftler aus den Bereichen Biodiversität, Ökologie oder Umweltwissenschaften.

- Citizen Scientists: Ehrenamtliche, die sich an wissenschaftlichen Projekten beteiligen.
- Naturschutzorganisationen: Institutionen, die Artenschutz- oder Umweltprojekte durchführen.
- Bildungseinrichtungen: Schulen, Universitäten und andere Lernstätten für Lehrzwecke und studentische Forschungsprojekte.
- Landwirtschaftliche Betriebe: Landwirte, die ihr Wissen über Bestäuber und Schädlingsbekämpfung erweitern möchten.
- Regierungs- und Umweltbehörden: Organisationen, die Umweltschutzmaßnahmen überwachen oder Berichte erstellen.
- Technologie- und Umweltenthusiasten: Personen mit Interesse an IoT und nachhaltiger Technologie.

3.2 Use Cases

Durch die grosse Anzahl an verschiedenen Zielgruppen gibt es auch viele verschiende Scenarien, in dem der Einsatz einer Edge ML Kamera vorstellbar ist.

- Biodiversitätsforschung: Überwachung von Bestäubern, Vögeln, Säugetieren oder Pflanzenwachstum.
- Artenschutz: Identifikation und Überwachung gefährdeter Tier- oder Pflanzenarten.
- Umweltüberwachung: Aufzeichnung und Analyse des Einflusses von Umweltbedingungen wie Temperatur, Feuchtigkeit oder Lichtverhältnissen auf die Pflanzen oder Tierwelt.
- Landwirtschaft: Erkennung von Schädlingen, Optimierung des Pflanzenwachstums.
- Bildungsprojekte: Praktische Anwendungen für Schüler und Studierende in Naturwissenschaften und Technik, als Ausbildungs Objekt für eine Machine Learning Applikation.
- Citizen Science: Ermöglicht Gemeinschaften, eigene wissenschaftliche Projekte durchzuführen, z. B. zur lokalen Artenvielfalt oder sich grösseren Projekten anzuschliessen.

- Was sollen Personen fähig sein zu machen

3.2.1 Referenz Use Case

Um bei der Entwicklung der Edge ML Kamera zielgerichtet arbeiten zu können, ist die Mitwelten Bestäuber Analyse das zu erfüllende Scenario. Durch die grosse Anzahl an verfügbaren und teilweise bereits gelabelten Bilder ist der Weg ein neues Modell zu

trainieren geebnet. Die Bestäuber Detektion erfolgt in folgenden wesentlichen Schritten. Ein Foto von Blüten durchläuft in einem ersten Schritt eine Inferenz zur Detektion von Blüten. Die auf dem Bild detektierten Blüten werden anschliessen ausgeschnitten und somit zu kleineren Fotos. Jedes dieser Fotos wird anschliessend mit einer weiteren Inferenz und einem anderen Machine Learning Modell auf Bestäuber untersucht. Eine grosse Herausforderung in diesem Setup ist, dass die Anzahl Bestäuber-Inferenzen mit der Anzahl detektierten Blüten steigt. Dies kann bei grossen Blütenzahlen zu langen Analysen eines einzigen Bildes führen.

Abbildung 4: Mitwelten Machine Learning Pipeline

3.3 Anforderungen

- Funktionale
  - Features (Cam Preview, Inferenz ausführen, Modelwahl, Framework wahl)
  - Hardware
- Nicht funktionale
  - Inferenz Zeitlimit
  - Einsatzgebiet

Wie in den Kapitel zuvor diskutiert, gibt es eine breite Anwendung für eine Edge ML Kamera. Der Referenz Use Case für die Entwicklung des Systems ist die in der Mitwelten Projekt eingesetzte Bestäuber Analyse. Dieser Anwendungsfall dient somit zur Definierung der Anforderungen an das System.

3.3.1 Funktionale

Die folgenden funktionalen Anforderungen an eine Edge ML Kamera wurden identifiziert:

- Datenerfassung: Das System muss anhand einer angeschlossenen Kameraeinheit Bilder kontinuierlich erfassen können
- Analyse Resultate: Das System muss die Analysedaten für den jeweiligen Anwendungszweck zur Verfügung stellen

- Benutzerinteraktion: Das System muss vom Benutzer konfiguriert werden können
- Energieversorgung: Während des Betriebs muss eine stabile Energieversorgung vorhanden sein

3.3.2 Nicht Funktionale

Die folgenden nicht funktionalen Anforderungen an eine Edge ML Kamera wurden identifiziert: Analysedauer: Im Mitwelten Projekt ist definiert, dass die Analyse von Bestäubern in 15 Sekunden erfolgen muss [5, p 62]. Die Zeit ist von der Verweildauer von Bestäuber auf einer Blume limitiert.

3.4 Evaluation

Im folgenden werden Resultate der Untersuchungen bezüglich ML Frameworks und Hardware aufgezeigt. Dabei sind Inferenzen mit PyTorch, Tensorflow lite, ONNX und Hailo Modellen implementiert worden. Die Inferenzen wurden auf Raspberry Pi's mit und ohne Beschleuniger Hardware ausgeführt und gemessen.

3.4.1 Methodik

Bei den Verwendeten Modellen handelt es sich um die YOLOv5, v8 und v10 Modellreihen in verschiedenen Grössen. Die Grösse des Modells bestimmt die Anzahl der trainierten Gewichte eines Modells. In diesen Versuchen werden die kleineren Modelle verwendet, weil diese grundsätzlich schneller sind. Der Preis für weniger Gewichte und somit schnelleren Inferenzen ist die geringere Genauigkeit.

Die YOLO Modelle wurden gewählt, weil sie zum Zeitpunkt dieser Arbeit gute Ergebnisse in den Bildanalysen liefern und breit eingesetzt werden. Zudem ist die Pipeline des Referenz Use Case Abschnitt 3.2.1 mit Modellen der YOLO Generation 5 umgesetzt. Die Messungen wurden jeweils auf den von Ultralytics vor trainierten YOLO Modellen mit Bildern des COCO Standard Datenset durchgeführt.

3.4.2 ML Framework

- Unterschiede
- Vor und Nachteile
- Auswertung

Im folgenden werden die Inferenzzeiten gleicher Modelle mit unterschiedlichen ML Frameworks verglichen. Dies auf einem Raspberry Pi 5 8GB mit einem aktiven Kühlelement.

3.4.2.1 Vergleich Inferenzzeiten

Die folgende Abbildung Abbildung 5 zeigt auf, wie sich die Inferenzzeiten der jeweiligen ML Frameworks in Bezug auf die YOLO Generation verhält. Ersichtlich ist, dass hauptsächlich die Inferenzzeit der PyTorch Inferenz mit jeder Generation signifikant gestiegen ist. Die anderen Frameworks weisen nur geringe Unterschiede auf, wobei die Tendenz des Anstieges bei jeder Inferenzzeit feststellbar ist. Ein weiteres Merkmal ist, dass die Inferenz mit dem ONNX Model bei allen Versuchen am wenigsten Zeit benötigt.

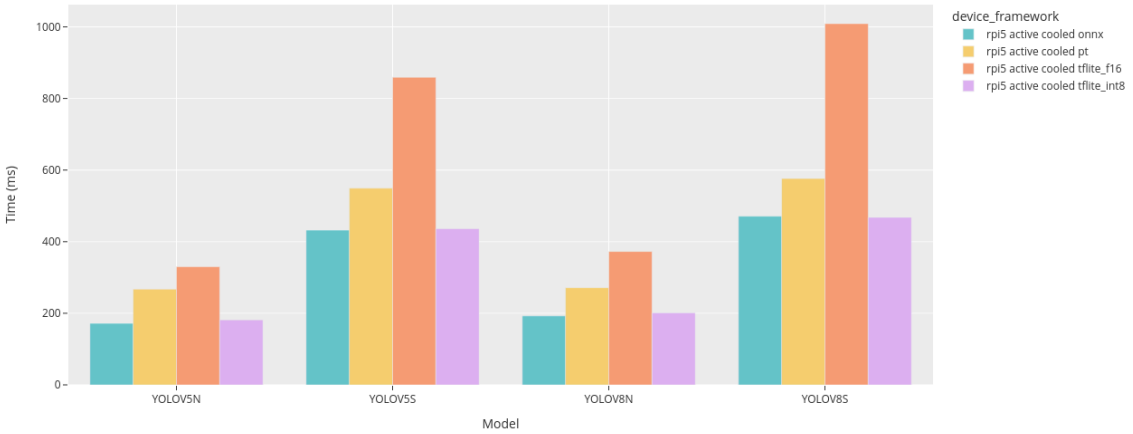

Abbildung 5: Vergleich Inferenzzeit YOLOv5, v8 auf Raspberry Pi 5

In einem nächsten Schritt wird ONNX mit NCNN verglichen. Wie in Abschnitt 2.3.1.5 erwähnt, ist NCNN für Geräte mit begrenzter Rechenleistung optimiert. Dies zeigt sich auch in der folgenden Abbildung Abbildung 6. Die Inferenzzeiten halbieren sich nochmals gegenüber der Inferenz mit ONNX.

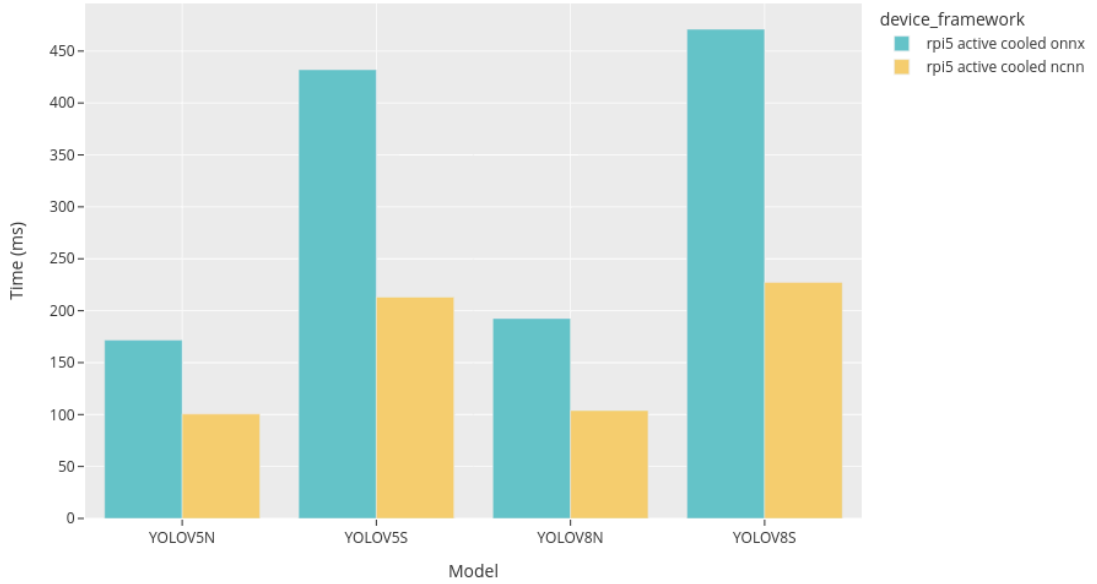

Abbildung 6: Vergleich ONNX und NCNN auf Raspberry Pi 5

Aus diesen Analysen geht hervor, dass mit NCNN auf einer Rasperry Pi 5 CPU die schnellsten Ergebnisse erzielt werden können. Um nun die Inferenzzeiten weiter zu beschleunigen, wird zusätzliche Hardware benötigt.

3.4.3 Hardware

- Recherche Hardware
  - Beschleuniger (TPU, AI-Camera)
  - Kamera (Rasperry Pi Cam, Webcam)
  - Edge Computer (Raspberry Pi 4,5)

Im folgenden werden verschiedene Hardware Setups miteinander verglichen. Teilweise beansprucht spezifische Hardware auch definierte ML Frameworks. Somit ist der Vergleich von verschiedener Hardware nicht mit gleichen Frameworks möglich. Dennoch können die Inferenzzeiten verglichen werden.

3.4.3.1 Raspberry Pi Vergleiche

Der erste Verglich zeigt den Fortschritt der Raspberry Pi Generationen auf. Die Inferenzzeiten eines ONNX Models ist auf einem Raspberry Pi 5 mehr als doppelt so schnell.

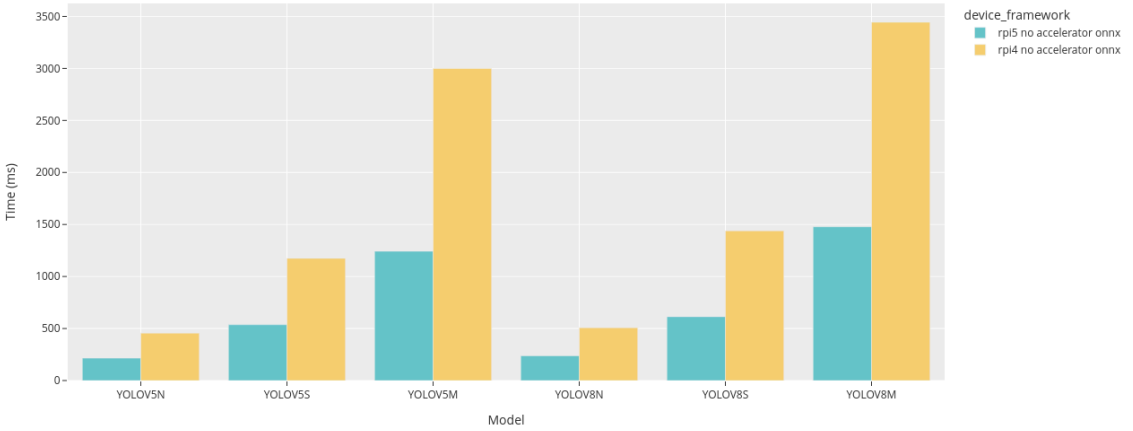

Abbildung 7: Vergleich Raspberry Pi 4 vs. Pi 5

Ein weiterer wichtiger Punkt ist die Kühlung des Systems. Ansonsten wird die CPU Leistungs des Rechners gedrosselt. In der folgenden Abbildung 8 ist ersichtlich, dass die Inferenzzeiten mit Kühlung rund 20% Prozent schneller gegenüber der selben Inferenz auf dem Vorgängermodell Raspberry Pi 4.

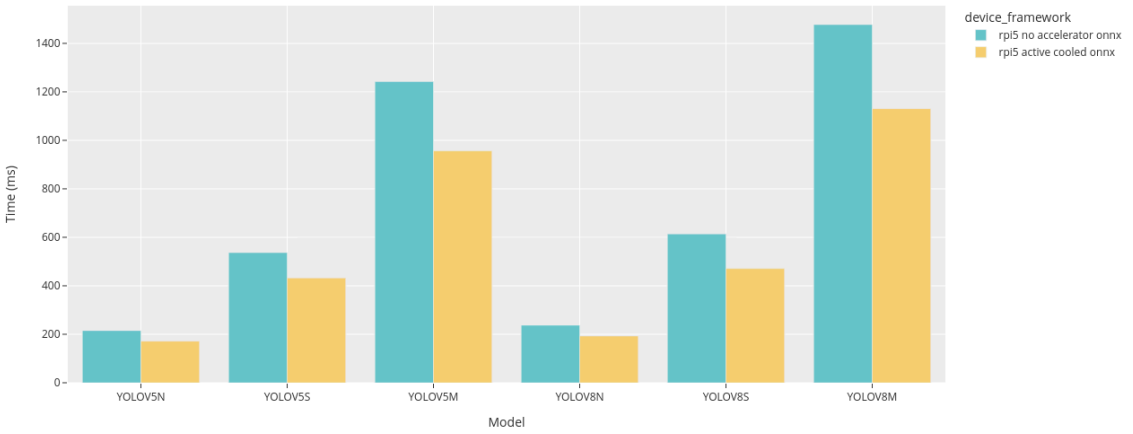

Abbildung 8: Vergleich Raspberry Pi 5 mit und ohne Kühlung

3.4.3.2 Coral Accelerator

Das Toolkit Coral [13] wurde von Google entwickelt und stellt Beschleuniger verschiedener Art zur Verfügung. Im Konstext dieser Arbeit wurden zwei dieser Beschleuniger untersucht: Ein USB Dongle mit einer Edge TPU von 4 TOPS und eine ein über PCI verbundenes Board mit einer TPU von 8 TOPS. Bei den Versuchen hat sich herausgestellt, dass der USB Dongle unzuverlässig ist. Es kam während länger laufenden Tests wiederholt zu Verbindungs unterbrüchen. Die Edge TPU über PCI funktionierte zuverlässig.

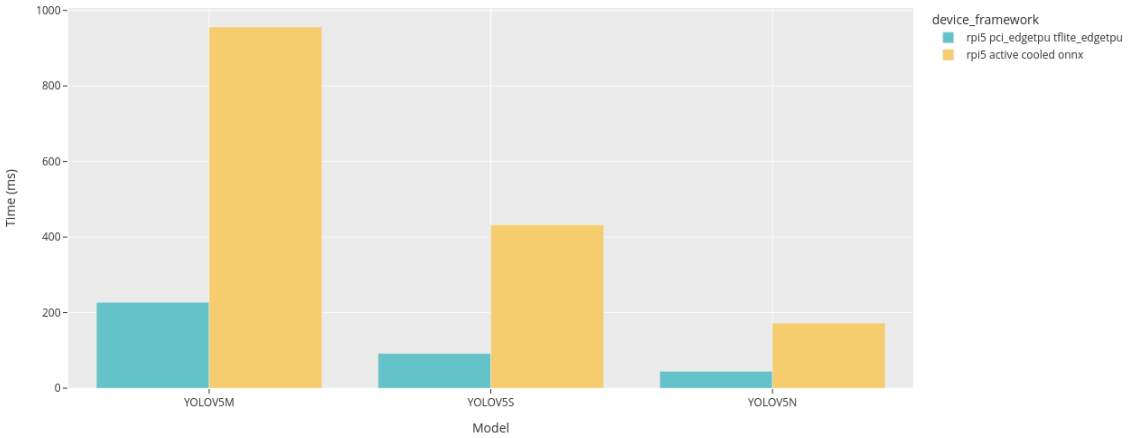

Abbildung 9: Vergleich Raspberry Pi 5 vs. Coral PCI Edge TPU

Es ist sofort ersichtlich, dass die Inferenz auf der Edge TPU um ein vielfaches schneller ist als auf der CPU des Raspberry Pi 5.

3.4.3.3 Hailo

Hailo [14] ist ein Unternehmen, welches auf Hochleistungs KI-Edge-Prozessoren spezialisiert hat. In dieser Untersuchung wurden die beiden Modelle Hailo8l (13 TOPS) und Hailo8 (26 TOPS) eingesetzt. Die beiden Module werden jeweils über die PCI Schnittstelle mit dem Raspberry Pi 5 verbunden. Aufgrund der benötigten PCI Schnittstelle kann das Raspberry Pi 4 Modell nicht verwendet werden. Für die Messungen der Inferenzzeit ist das von Hailo zur Vergügung gesetllte ML-Framework verwendet worden. Zudem müssen die Modelle im Hailo eigenen Format .hef sein, um sie auf den Accelerator auszuführen. Hailo stellt ein Model Zoo [15] zur Verfügung, welcher die Modelle YOLOv5s und YOLOv5m beinhaltet. Die folgende Abbildung 10 zeigt den Vergleich mit der Coral Edge TPU.

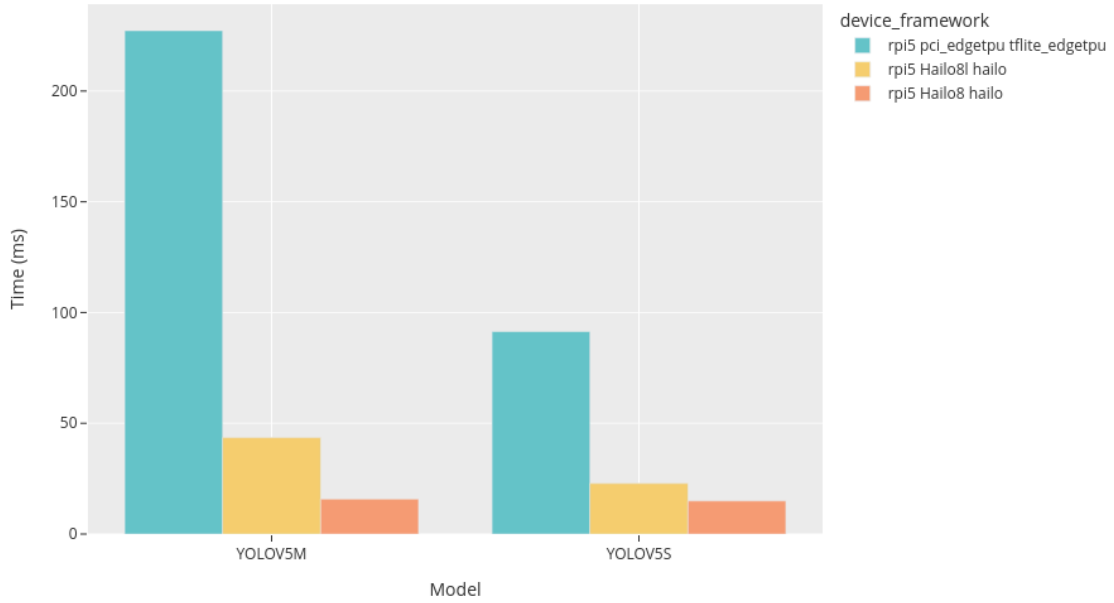

Abbildung 10: Vergleich Hailo Accelerator vs. Coral PCI Edge TPU

Die beiden Hailo Accelerator sind signifikant schneller als der schon etwas ältere Coral Accelerator. Ebenso ist ersichtlich, dass sich die doppelte Anzahl TOPS direkt auf die Inferenz Geschwindigkeit auswirkt. Die schnellste Inferenz wird somit mit dem Hailo8 Accelerator erzielt und beträgt run 15ms.

4 Umsetzung

(Was ich tatsächlich Umgesetzt habe)

4.1 System Architektur (Grobsicht)

- Qualitäten (Erweiterbarkeit, Robustheit, Self-contained, Einfachheit)
- Modularisierung (Komponenten: Cam, ML Framework)

4.2 Software Design

(Aufteilung in Klassen)

- Klassen Diagramm
- Sequenz Diagramm

4.3 Prototypen: verschiedene System Setups

(welche Konfiguration wurde verwendet)

- Prototyp 1
- Prototyp 2

5 Validierung

(Auf MW Pipeline, mit Blumen Bilder)

- Performance
- Energie

Abbildungsverzeichnis

Tabellenverzeichnis

Appendix A: Supplementary Material

– Supplementary Material –

Bibliographie

[1] A. Bonn *u. a.*, „Grünbuch Citizen Science Strategie 2020 für Deutschland“, Berlin, 2016.

[2] K. Vohland *u. a.*, Hrsg., „The Science of Citizen Science“. Springer International Publishing, Cham, 2021. doi: 10.1007/978-3-030-58278-4.

[3] bdm, „Verlässliche Daten über unsere Lebensgrundlage“. Zugegriffen: 15. Januar 2025. [Online]. Verfügbar unter: https://www.biodiversitymonitoring.ch/index.php/de/

[4] T. Wullschleger, „Data Acquisition for Urban Biodiversity Monitoring“.

[5] T. Wullschleger, „Automated Analysis for Urban Biodiversity Monitoring“.

[6] „TensorFlow“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://www.tensorflow.org/

[7] „PyTorch“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://pytorch.org/

[8] „OpenAI standardizes on PyTorch“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://openai.com/index/openai-pytorch/

[9] „ONNX | Home“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://onnx.ai/

[10] H. Ni und The ncnn contributors, „ncnn“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://github.com/Tencent/ncnn

[11] Ultralytics, „Home“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://docs.ultralytics.com/

[12] awild, „andriwild/ip7-ml-model-eval“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://github.com/andriwild/ip7-ml-model-eval

[13] „Products“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://coral.ai/products/#prototyping-products

[14] „KI-Edge-Prozessoren für Höchstleistung - Hailo KI Chip“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://hailo.ai/de/

[15] T. Tapuhi *u. a.*, „Hailo Model Zoo“. Zugegriffen: 16. Januar 2025. [Online]. Verfügbar unter: https://github.com/hailo-ai/hailo_model_zoo