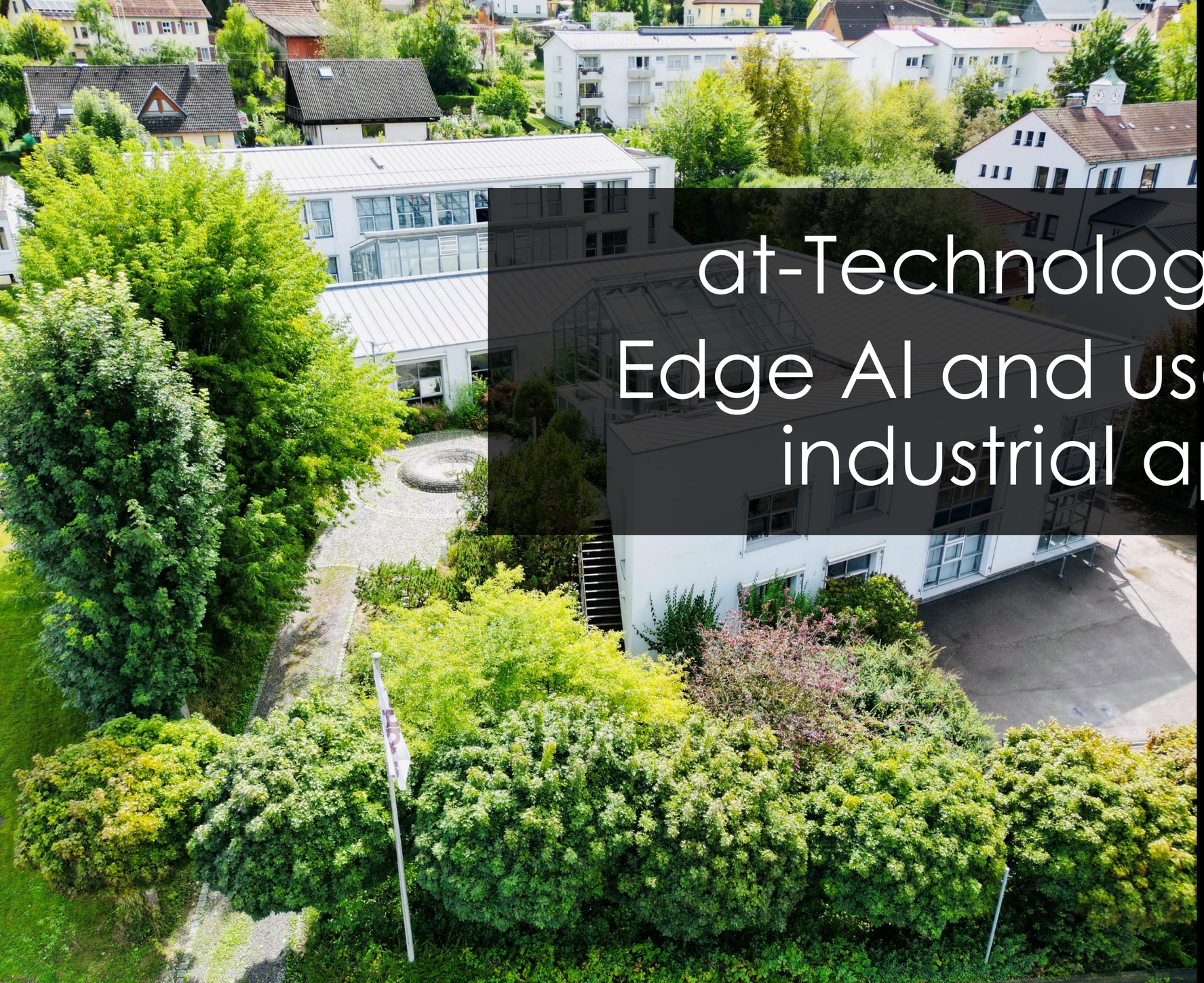


# at-Technologietag 2026
# Edge AI and use cases for industrial applications

endrich



# Über die endrich Group

- 14 Standorte in Deutschland und 7 Standorte in weiteren europäischen Ländern
- Im Oktober 2007 wurde der Distributor Novitronic AG in der Schweiz übernommen und damit das endrich Produktportfolio und die Mitarbeiterzahl enorm erweitert
- endrich, ein familiengeführtes deutsches Unternehmen, verbindet technische Unterstützung mit individuellen Logistikkonzepten – optimale Produktlösungen

**1976** — Gründung endrich



**2007** — Zukauf Novitronic AG

novitronic powered by endrich

**2009** — Gründung Joint Venture SunCon Europe GmbH

SunCon

**2018** — Neues Logistikzentrum in Nagold



**2019** — Gründung Inpaq Europe GmbH

INPAQ EUROPE GmbH

**2025** — Anbau Logistikzentrum in Nagold



**2026** — Gründung endrich R&D Ungarn

endrich R&D

System Integrity Team

endrich

# Agenda

1. Edge KI & Hardware
2. KI Use Cases
3. Projektbeispiel
4. Zusammenfassung

# Edge KI & Hardware

endrich

# KI: “Deep Learning” in Kürze

$$f(x) \rightarrow y$$

KI-Modelle

Text, Audio,
Bilder, usw.

Antwort auf
Fragen, Diktieren,
Objekte, usw.

Was die KI-Entwicklung antreibt

01 **Daten**

riesige Datensätze Modell lernt automatisch aus Beispielen

02 **Rechenleistung**

spezialisierte Chips (GPU, NPU)

03 **Modellarchitektur**

immer effizientere Modelle „f“

# NVIDIA GTC 2025

## KI Roadmap nach J. Huang

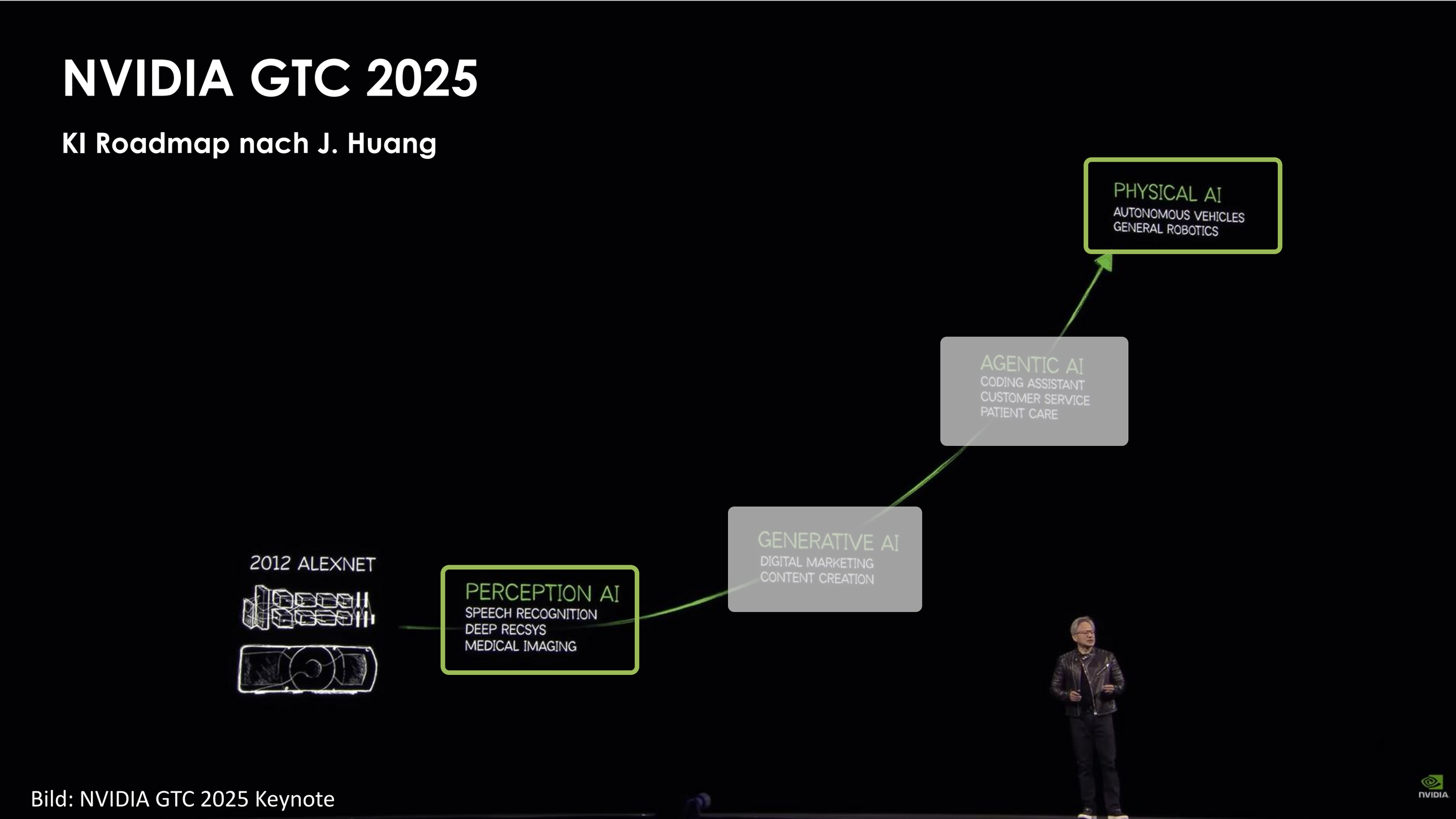



Bild: NVIDIA GTC 2025 Keynote

# Jagd nach Effizienz

Hardwarespezialisierung
Softwareoptimierung

**2012 – AlexNet Paper + GPU**

60 Mio. Parameter -
gewinnt “ImageNet Challenge: ILSVRC 2012”
Training: ~6 Tage auf 2 Grafikkarten
~250 W pro Grafikkarte

**2017 - MobileNet + NPU**

4,2 Mio. Parameter
14× kleiner als AlexNet
Und trotzdem genauer (57%→71%)
Beispiel:
Face ID auch im Flugmodus



NVIDIA
GTX 580 3GB



NVIDIA
GTX 580 3GB

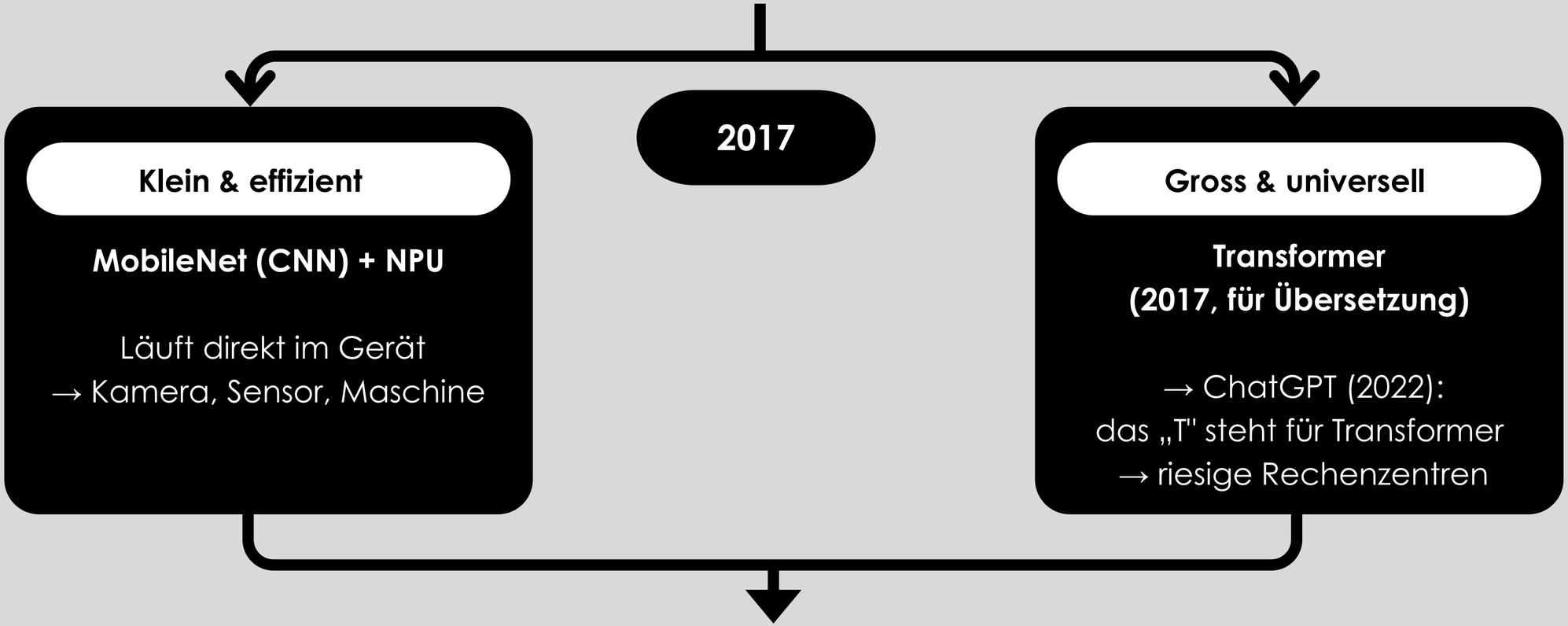



HiSilicon
Kirin 970



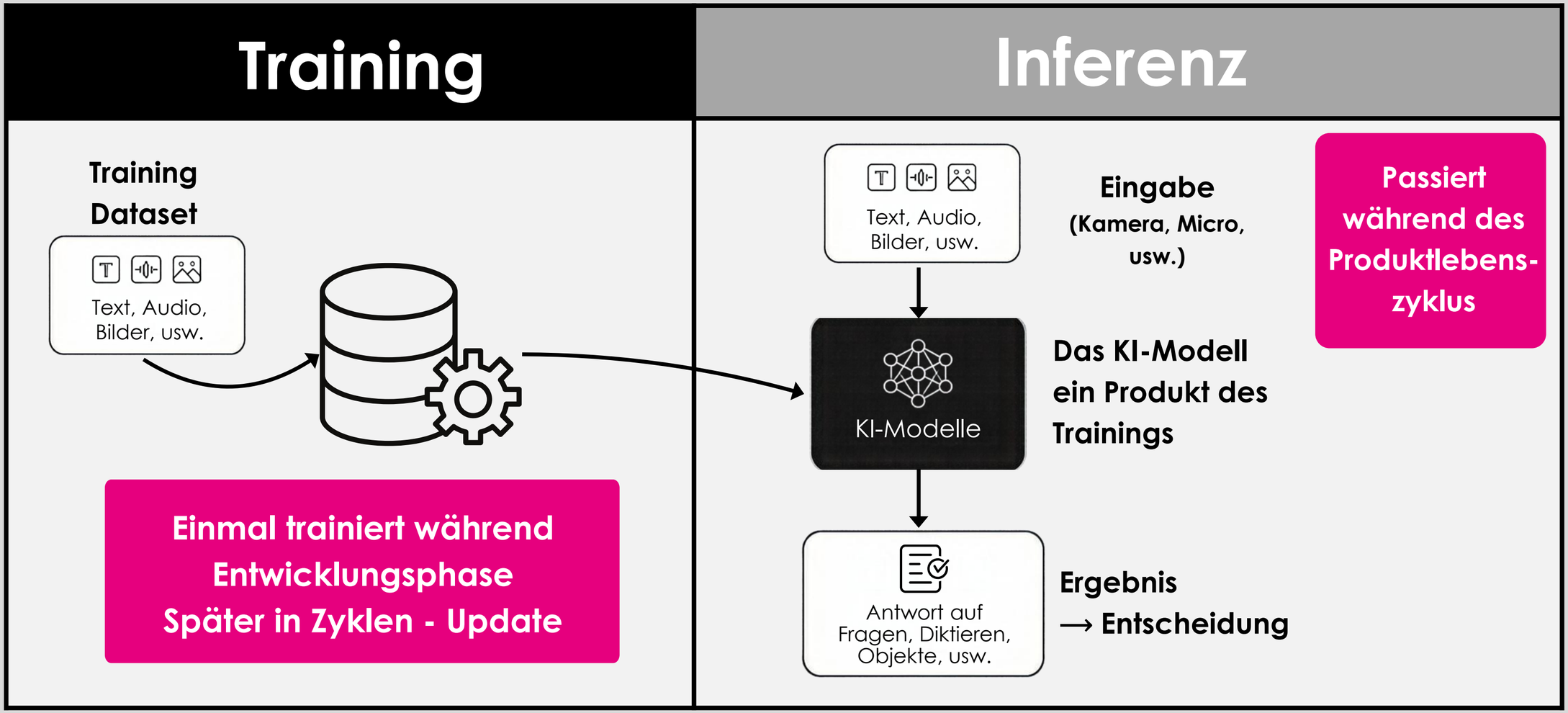

Apple A11
Bionic

# Die KI geht zwei Wege

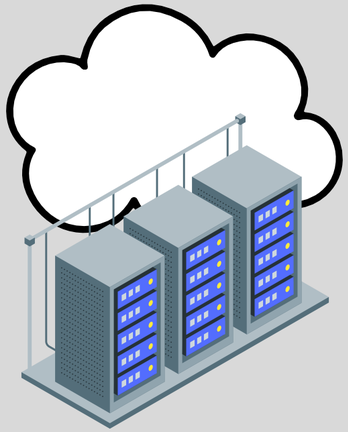



**Die Wege treffen sich: Transformer auch für Bilder (Vision Transformer, 2020)**
**und seit 2024 sogar in kleinsten Edge-NPUs (z. B. Arm Ethos-U85)**

# Edge KI - wie funktioniert es?



# Cloud oder Edge?











**Trainieren in der Cloud**

→

**Anwenden an der Edge (lokal)**



**Latenz** – Antwort in Millisekunden, direkt an der Maschine



**Ausfallsicherheit** – läuft auch ohne Netzwerk



**Datensouveränität** – Bilder bleiben im Gerät/ im Werk



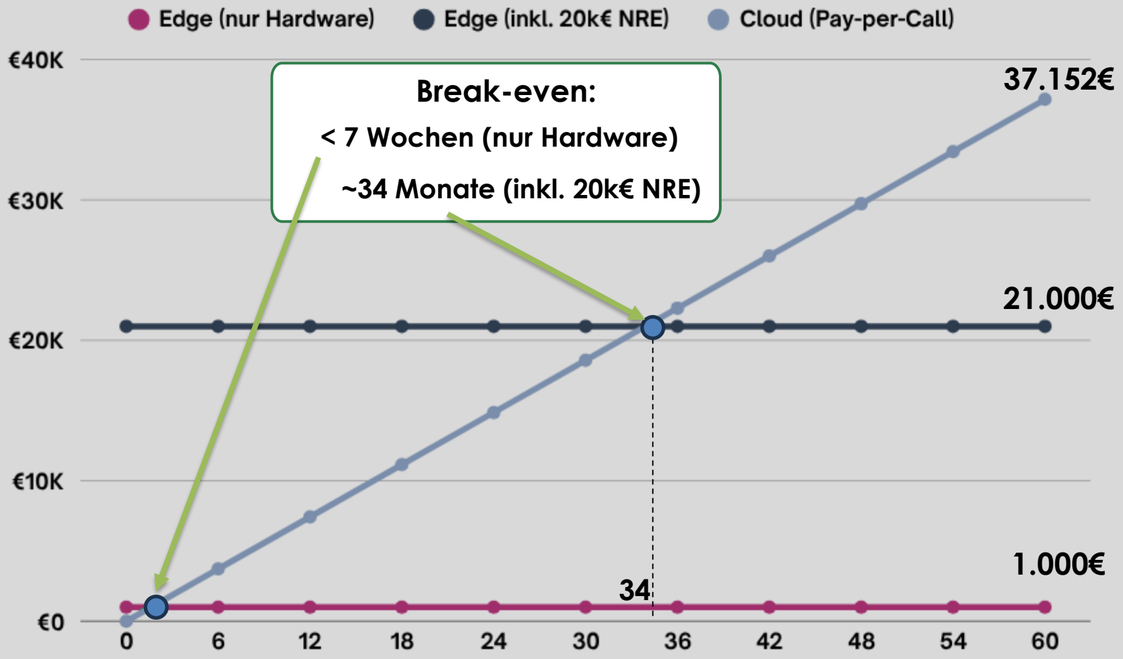
**Kosten** – bei Dauerbetrieb günstiger

GenAI und Agentic AI laufen heute meist noch in der Cloud

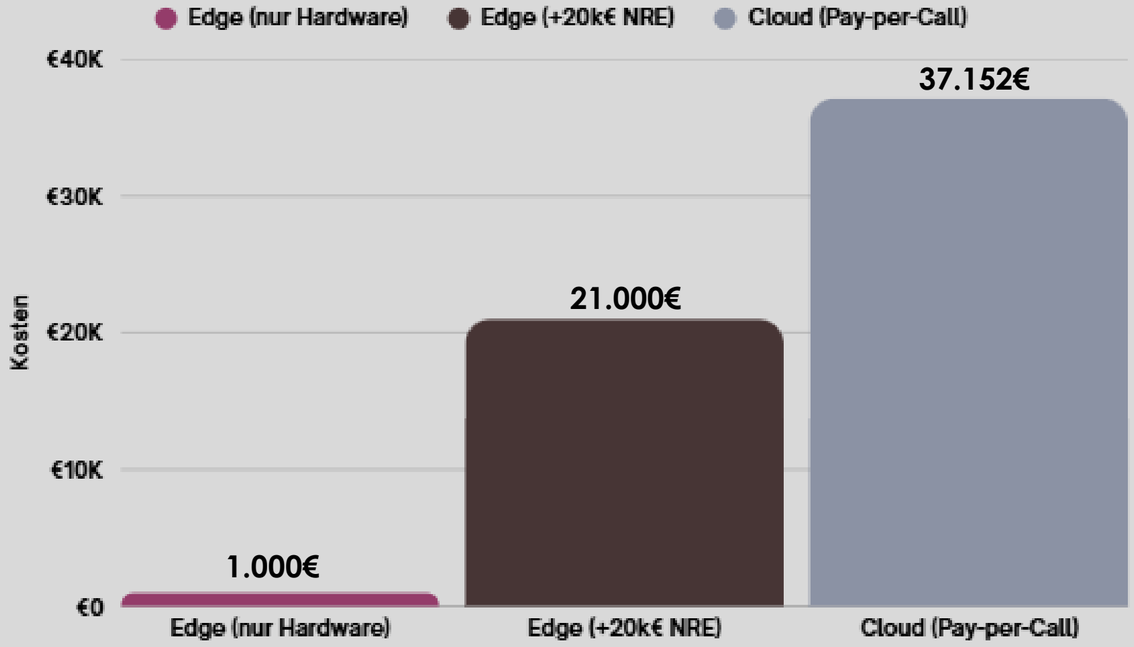
# 5-Jahres-TCO: kumulative Kosten, Edge vs. Cloud

**Szenario: 1 Prüfstation ·~1 Bild/Sek. ·16 Std./Tag ·250 Tage/Jahr → ~14,4 Mio. Inferenzen/Jahr, 72 Mio. über 5 Jahre**
(Annahme: die Cloud Lösung wäre technisch überhaupt geeignet für den Einsatz)

Kumulative Inferenzkosten im Zeitverlauf



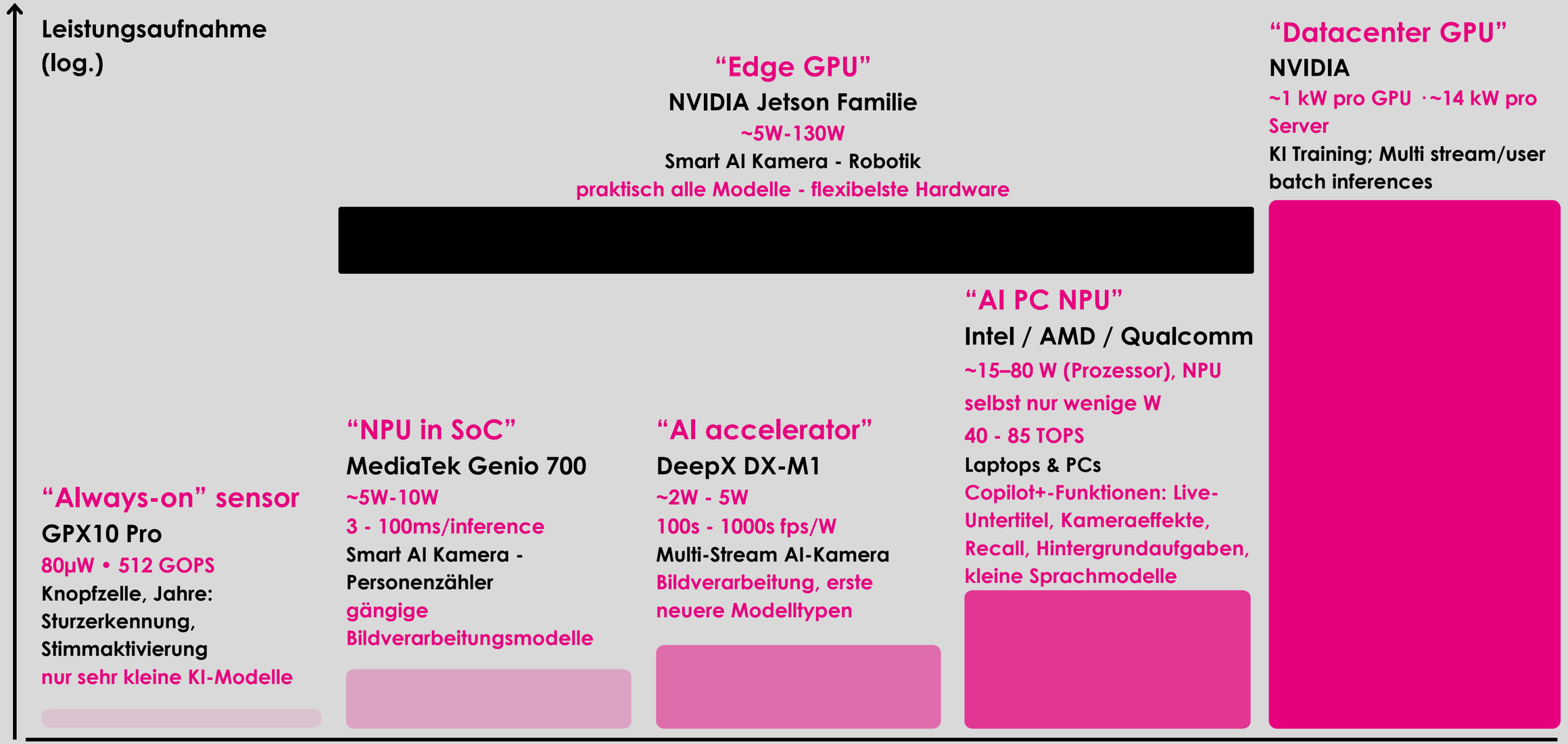

Gesamtkosten nach 60 Monaten – Gleiches Gerät, gleicher 5-Jahres-Zeitraum




**Gesamtersparnis: ~16.152€ über 5 Jahre (inkl. NRE) für 1 Gerät**
**Bei 1000 Einsatzorten (Geräte)** in 5 Jahren:
⟶ **1000x Kosten** sowohl für Cloud-Inferenz als auch Edge-Hardwarezukauf
⟶ **Entwicklungskosten pro Gerät 1000× kleiner** für Edge Lösung

Illustratives Beispiel – Hardwarepreis und NRE basieren auf eigener Fertigungsmenge.
Cloud-Kosten pro Inferenz ~ Azure Vision 37'000 € ÷ 72 Mio. ≈ 0,5 € pro 1000 Abfragen, also ein Mengenrabatt-Tarif

# Welche Hardware für welche Aufgabe?

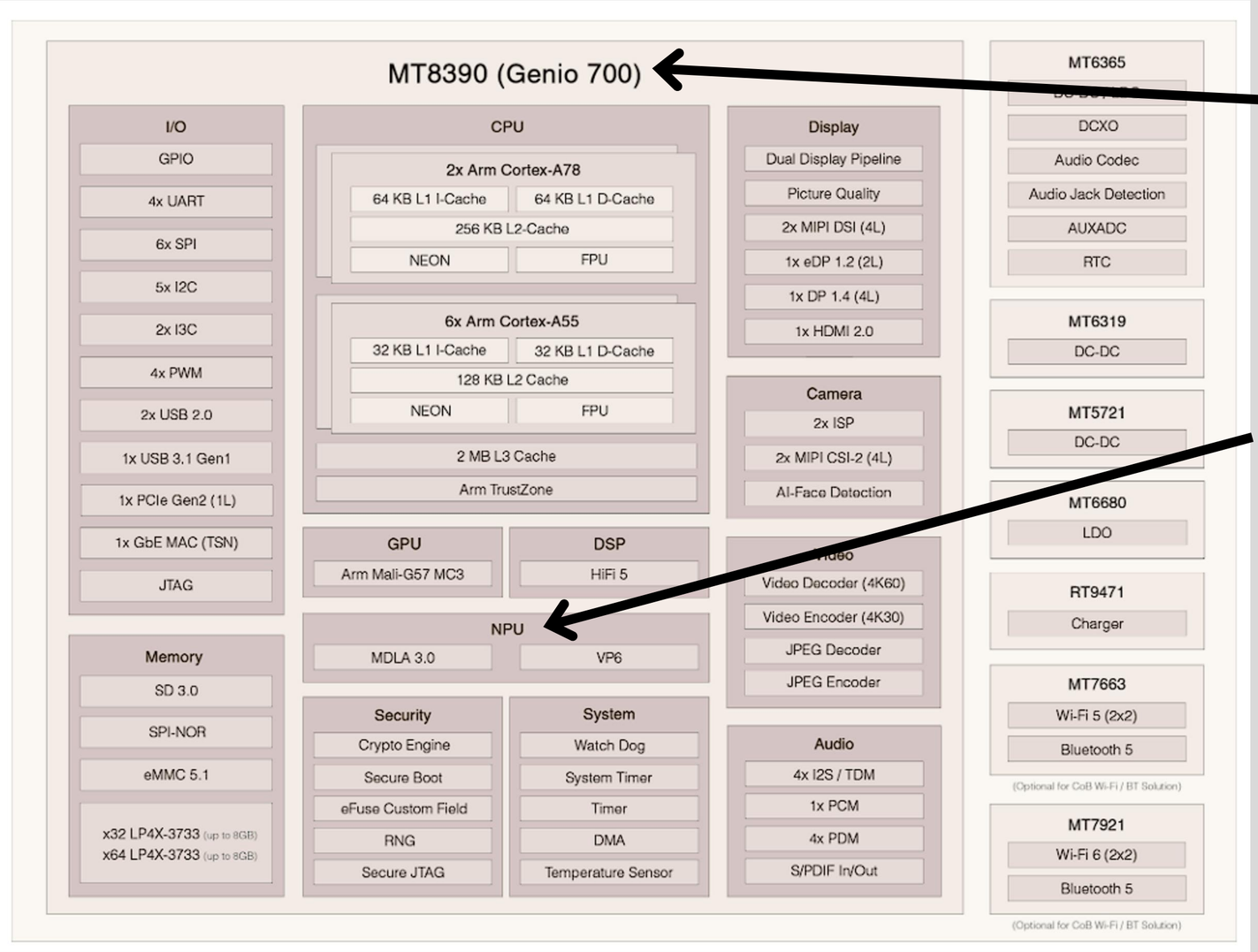


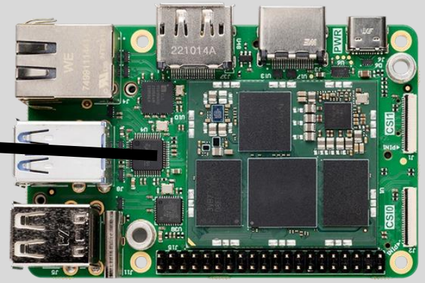
# Welche Hardware? Faustregel

| Anforderung | Sensor-KI | NPU im SoC | KI-Beschleuniger | Edge-GPU | Cloud |
|---|---|---|---|---|---|
| Batteriebetrieb | ● | ◐ | ○ | ○ | ○ |
| Mehrere Kameras / hohe Bildrate | ○ | ◐ | ● | ● | ◐/○ |
| Neueste oder wechselnde Modelle | ○ | ○ (→◐) | ◐ | ● | ● |
| Tiefe Stückkosten in der Serie | ● | ● | ◐ | ○ | ○ |
| Geringer Entwicklungsaufwand | ○ | ◐ | ◐ | ● | ● |

● Gut geeignet ◐ bedingt ○ ungeeignet

# Beispiel Hardware-Blockdiagramm



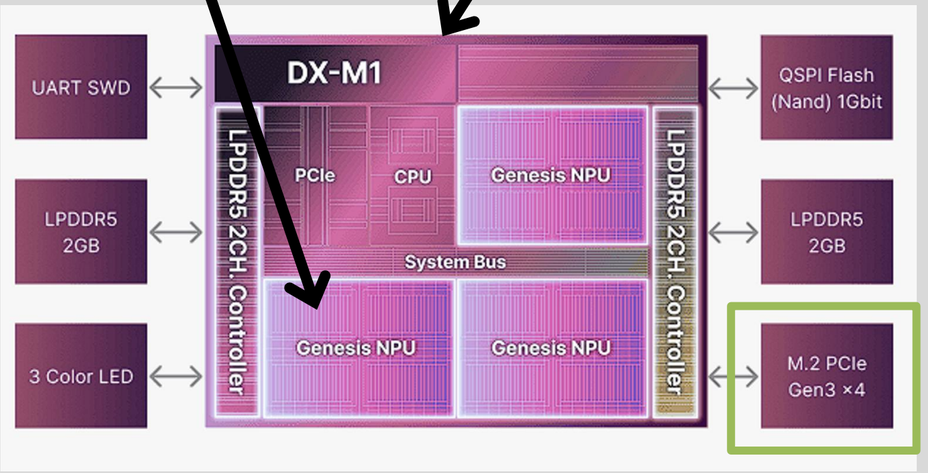



Grinn Mediatek Genio 700 SBC

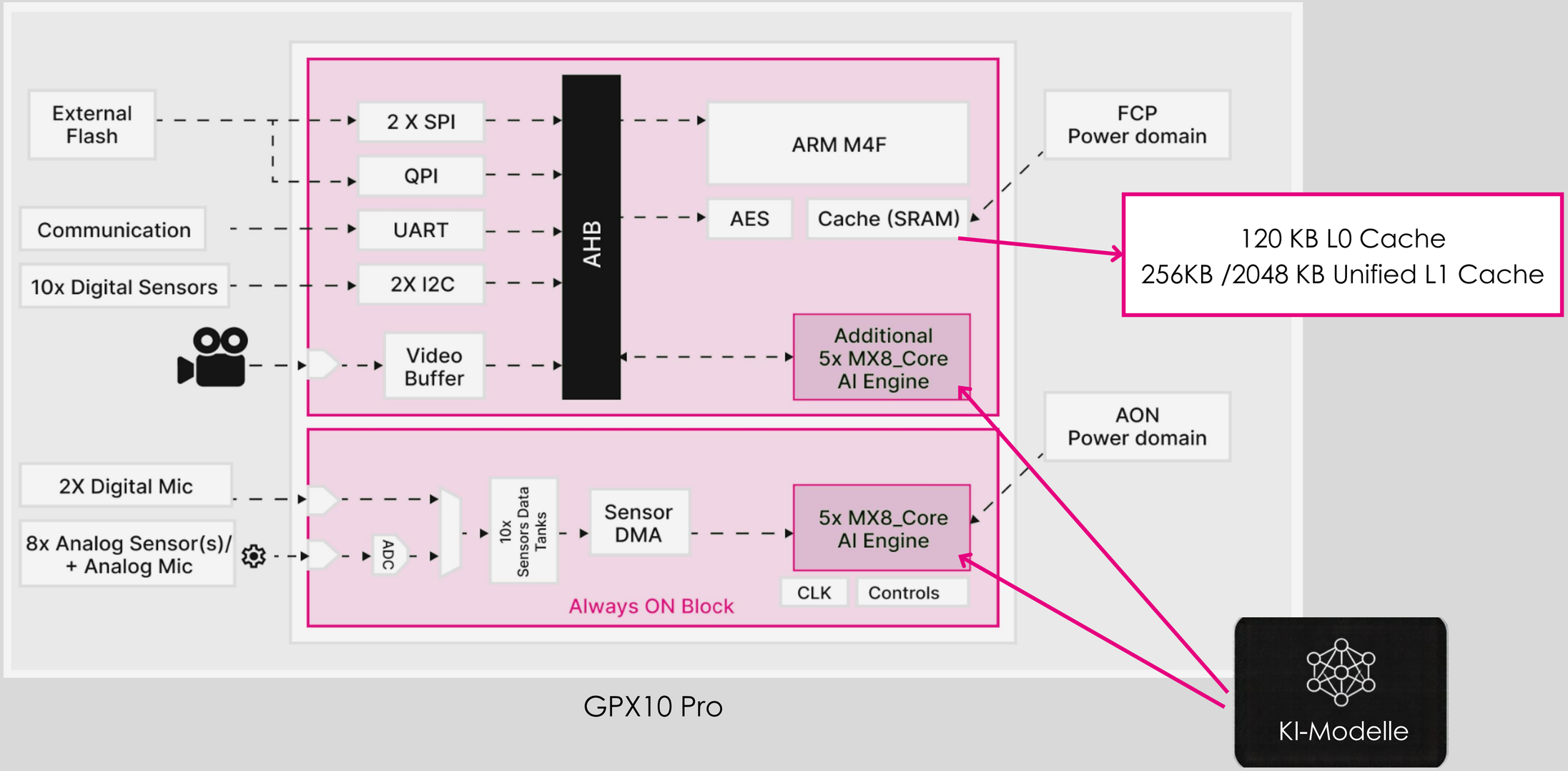



DeepX DX-M1

KI-Modelle



# Beispiel Hardware-Blockdiagramm



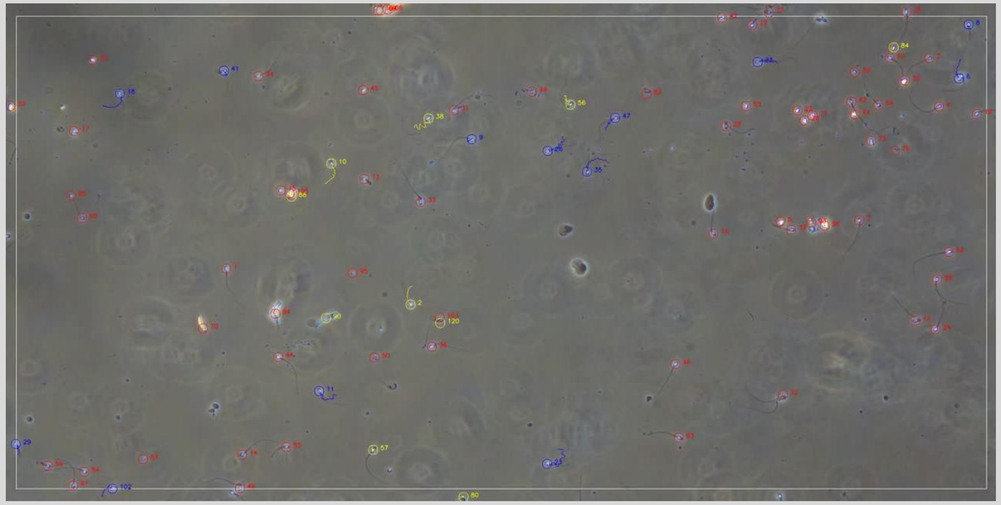

GPX10 Pro

# KI Use-Cases

endrich

# Spermienerkennung & Tracking

Übertragbar auf: Partikel- und Bewegungsanalyse in Mikroskopie und Fertigung



### 1 Spermienerkennung

Jede sichtbare Spermienzelle erkennen und lokalisieren.

### 2 Spermienverfolgung

Jeder erkannten Spermienzelle eine eindeutige ID zuweisen und ihre Bewegung über aufeinander-folgende Frames hinweg verfolgen.

### 3 Geschwindigkeitsbasierte Klassifizierung

Die Bewegungsgeschwindigkeit jeder verfolgten Spermienzelle wird anhand ihrer Trajektorie berechnet:

- Unbeweglich / Tot
- Normale Geschwindigkeit
- Hohe Geschwindigkeit

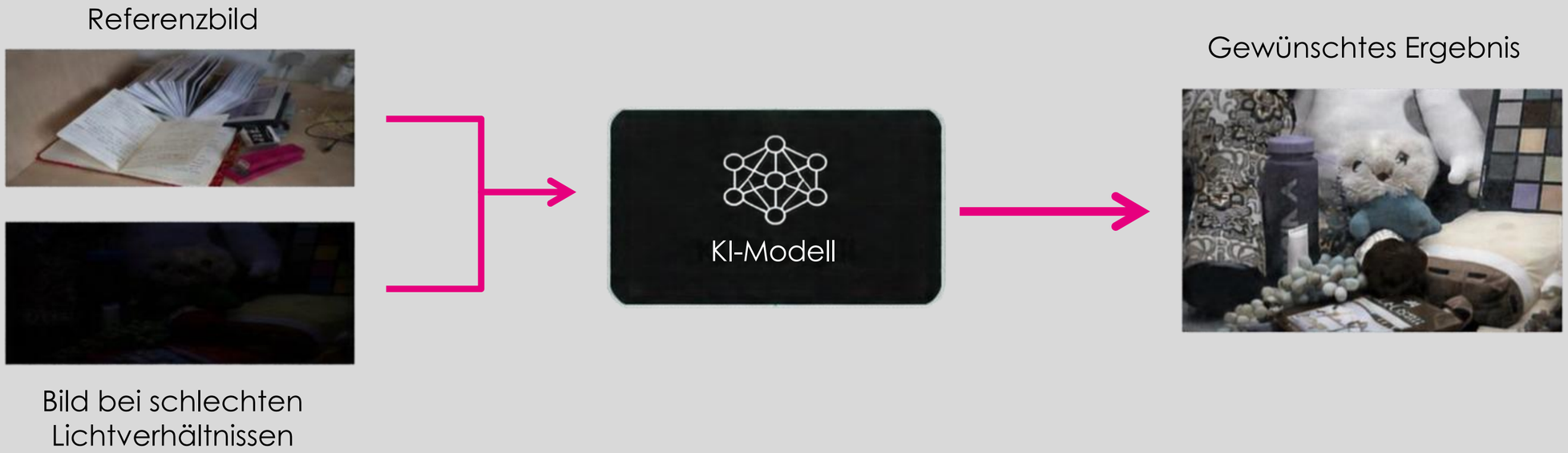
Das Ziel dieser KI-Aufgabe ist die automatische Analyse von Spermien in mikroskopischen Videosequenzen.

# Verbesserung von Bildern bei schlechten Lichtverhältnissen

Verbessert die Sichtbarkeit und visuelle Qualität von Bildern, die bei schlechten oder unzureichenden Lichtverhältnissen aufgenommen wurden, während die ursprüngliche Szenenstruktur und der Inhalt erhalten bleiben.

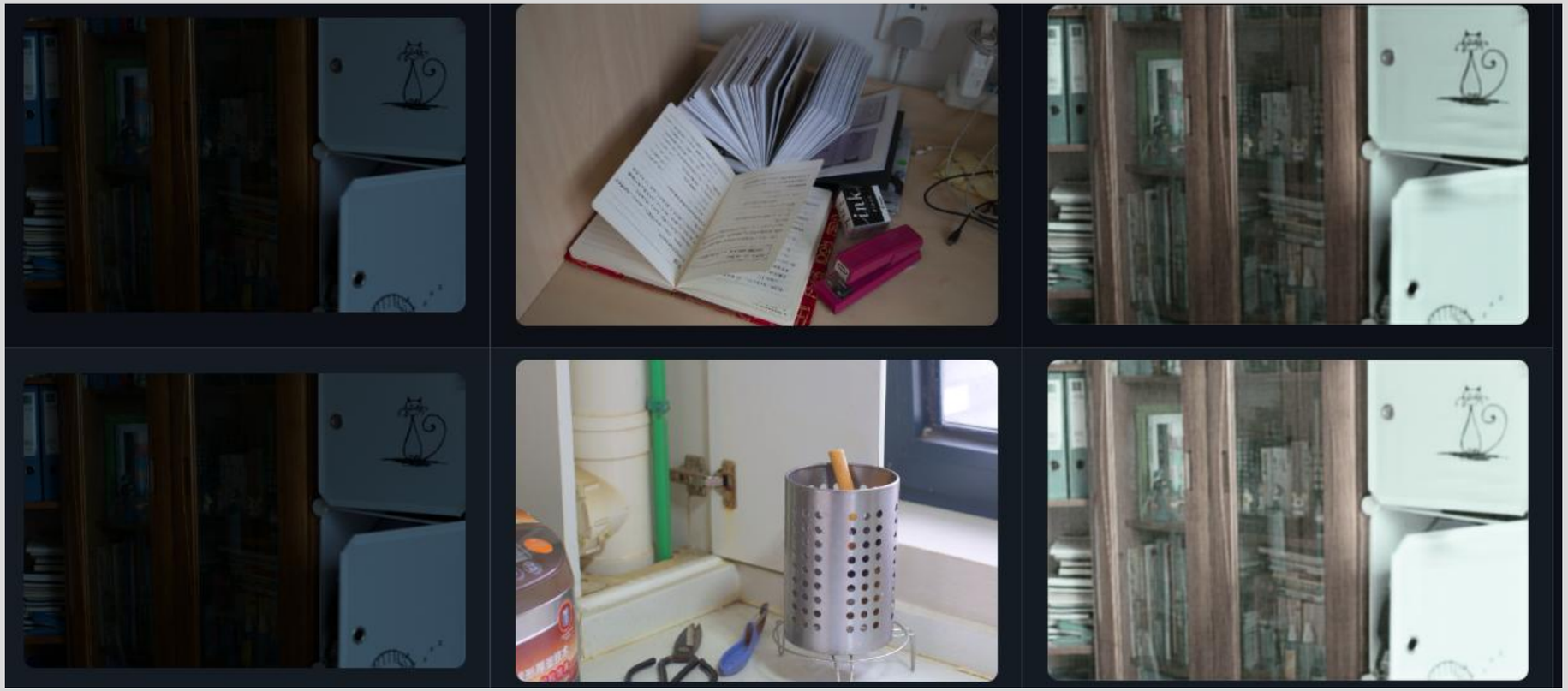
Das KI-Modell verwendet zwei Eingabebilder:

- Low-Light-RGB-Bild – liefert den Szeneninhalt, die Objekte, Texturen und die räumliche Struktur.
- Referenzbild – liefert die gewünschten Lichtmerkmale, wie Helligkeit, Verteilung der Beleuchtung, Kontrast und Farberscheinung.



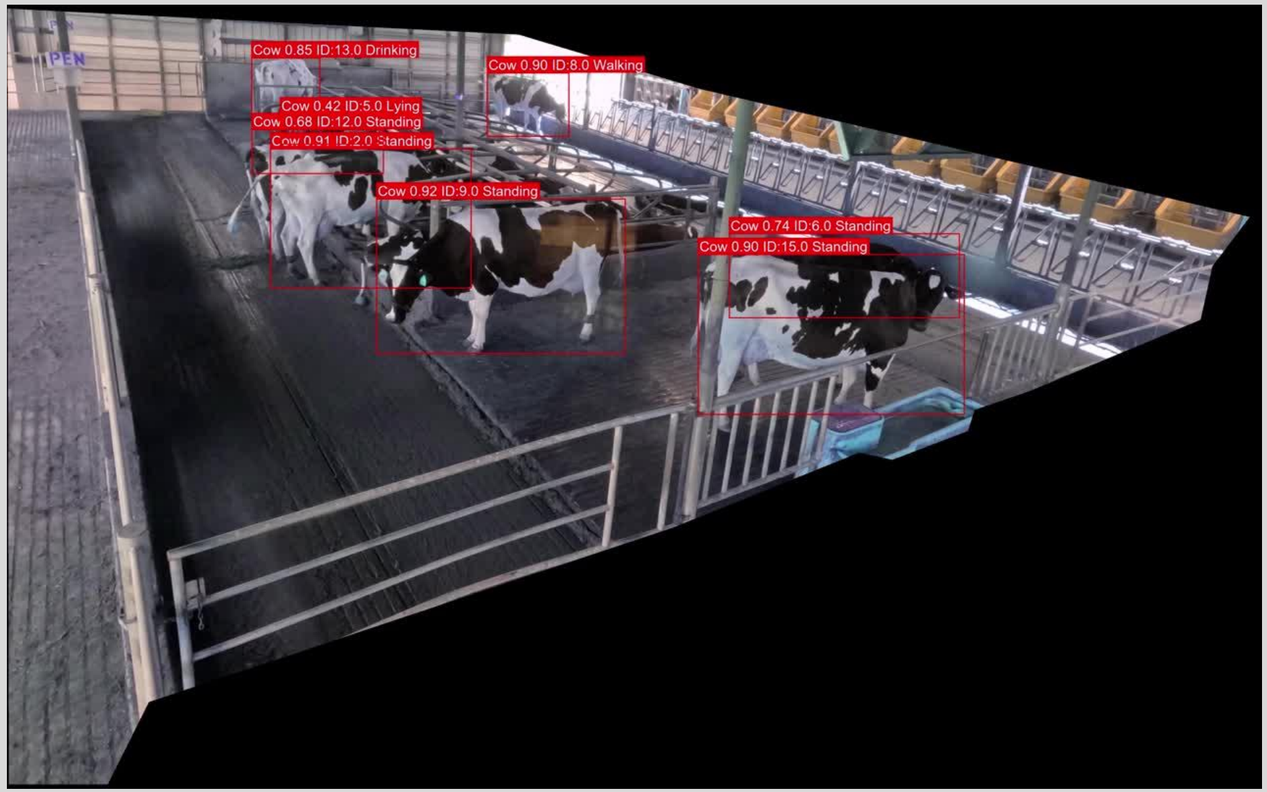
# Verbesserung von Bildern bei schlechten Lichtverhältnissen

Dunkles Bild + Referenzbild → KI-generiert



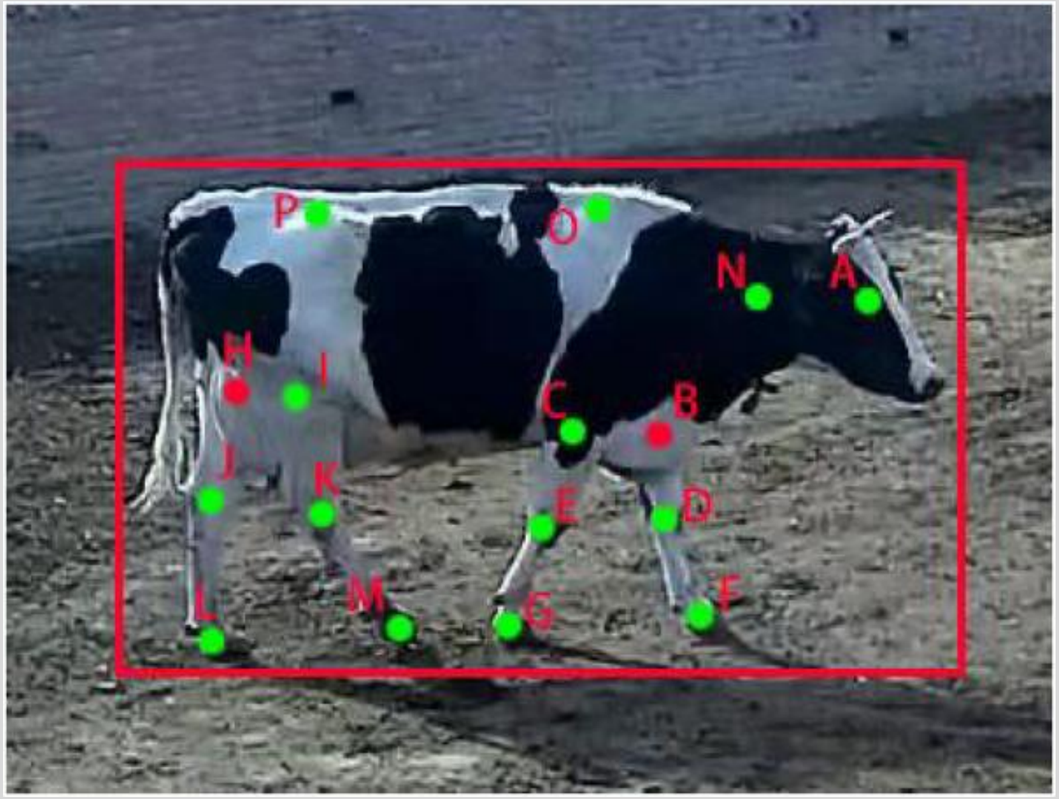
# Kuhanalyse und Früherkennung von Lahmheiten

- **Kuh-Erkennung** – Jede Kuh in den Kamerabildern erkennen und lokalisieren.
- **Individuelle Verfolgung** – Jede Kuh über aufeinanderfolgende Frames hinweg mithilfe einer eindeutigen ID verfolgen.



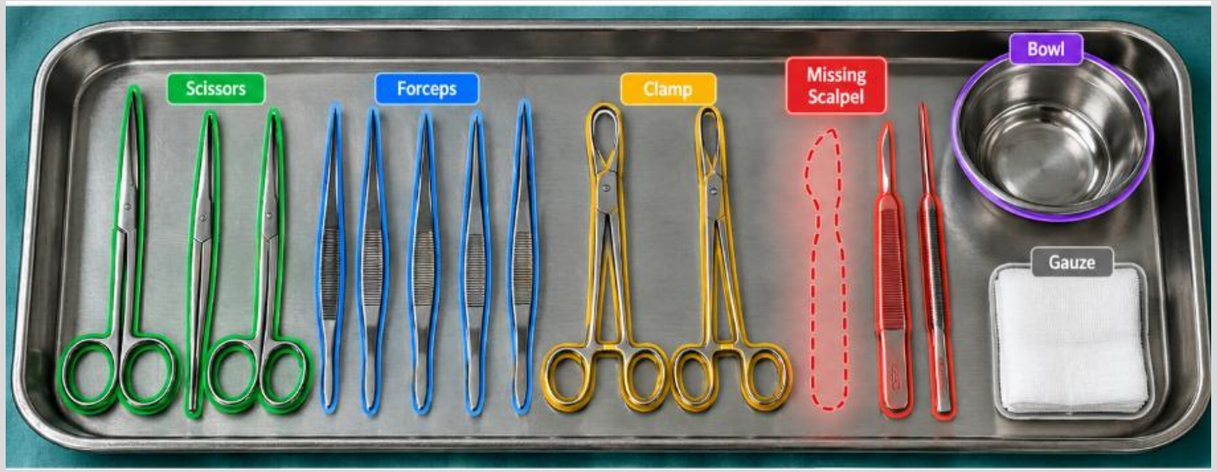
# Kuhanalyse und Früherkennung von Lahmheiten

- **Posenschätzung / Keypoint-Ermittlung** – Schlüsselpunkte (Keypoints) von Körper und Beinen schätzen, um Körperhaltung und Gangverhalten zu analysieren.
- **Ganganalyse** – Schrittlänge, Schrittfrequenz, Gehgeschwindigkeit, Standzeit sowie die Symmetrie der Links-Rechts-Bewegung messen.
- **Lahmheitsklassifizierung** – Die Kuh anhand von Gang- und Haltungsindikatoren in die Kategorien Unauffällig / Normal, Verdacht auf Lahmheit oder Lahm einstufen.



# Instrumentenkontrolle im Krankenhaus

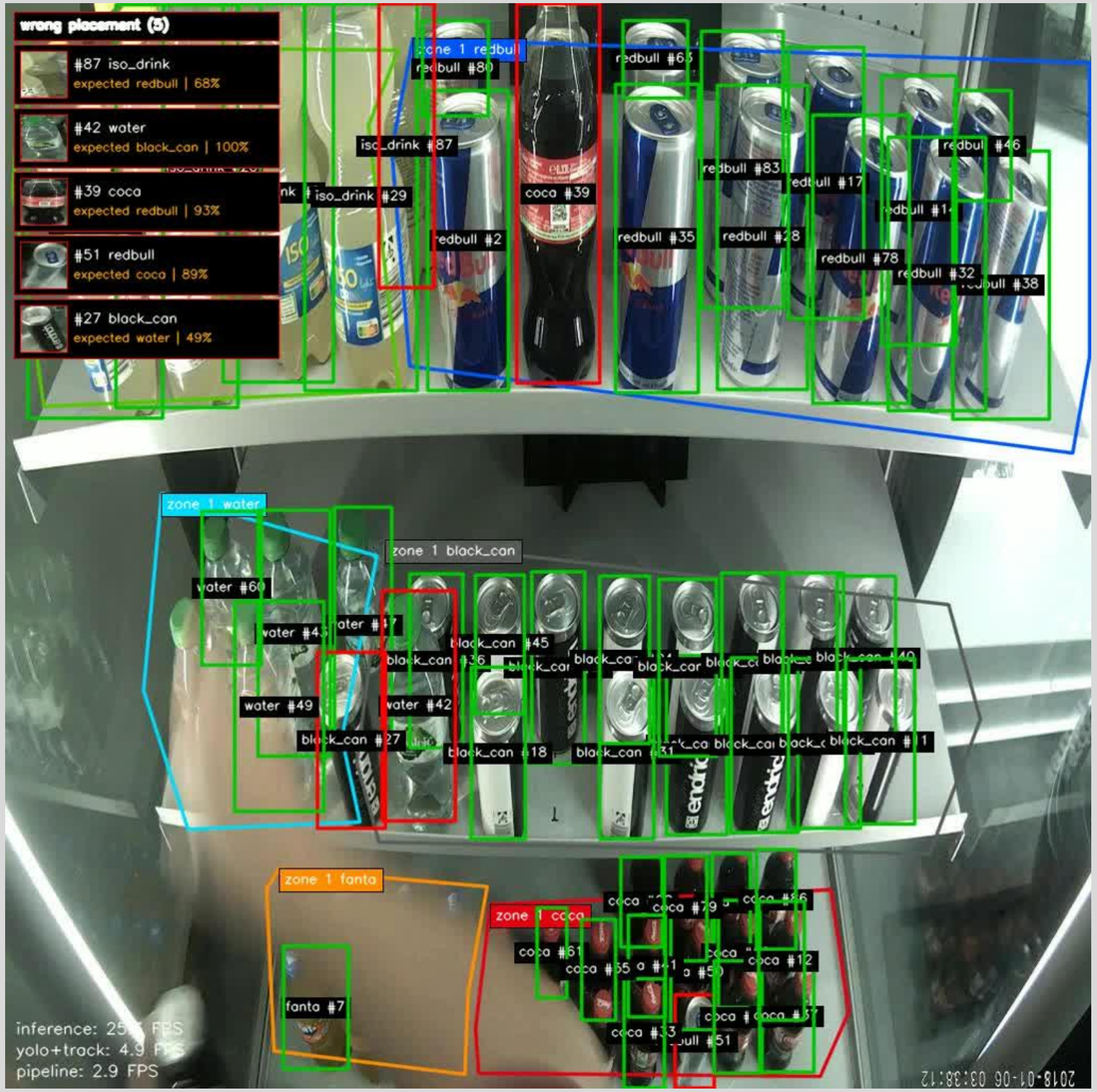
- **Bilderfassung** – Erfassen des OP-Siebes mit einer fest installierten Kamera.
- **Instrumentensegmentierung** – Erkennen und einzelnes Segmentieren jedes medizinischen Instruments.
- **Zählung** – Zählen der Anzahl der Instrumente pro Klasse.
- **Referenzvergleich** – Vergleichen des erkannten Siebinhalts mit der erwarteten Siebkonfiguration.
- **Erkennung fehlender/zusätzlicher Elemente** – Identifizieren fehlender, zusätzlicher oder falsch platzierter Instrumente.



# Planogram Analyse

**Ablauf:**

1. **Bild-/Videoaufnahme** – Aufnahme von Regalbildern mit einer fest montierten Kamera oder durch periodische Inspektionsaufnahmen.
2. **Regal-/ROI-Erkennung** – Festlegen der Regalbereiche, die überwacht werden sollen.
3. **Produkterkennung und -klassifizierung** – Erkennen jedes sichtbaren Produkts und Bestimmen seiner Kategorie bzw. SKU.
4. **Positionszuordnung** – Bestimmen, wo sich jedes Produkt innerhalb des Regals oder einer vordefinierten Zone befindet.
5. **Planogramm-Vergleich** – Abgleich der erkannten Regalanordnung mit dem Soll-Planogramm.
6. **Abweichungserkennung** – Identifizieren fehlender Produkte, falsch platzierter Produkte, Leerstellen, falscher Mengen oder falscher Regalpositionen.



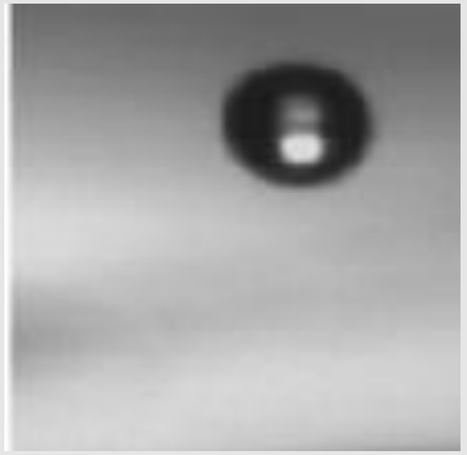
# Projektbeispiel

# Tropfenerkennung

“Warum ein Tropfen zählt”

**Wo berührungslos dosiert wird:**

| | |
|---|---|
| Elektronikfertigung | Klebstoff, Underfill, Lotpaste |
| Biotech & Pharma | Proben und Wirkstoffe exakt dosieren |
| Chemie | berührungsloses Dosieren und Mischen |



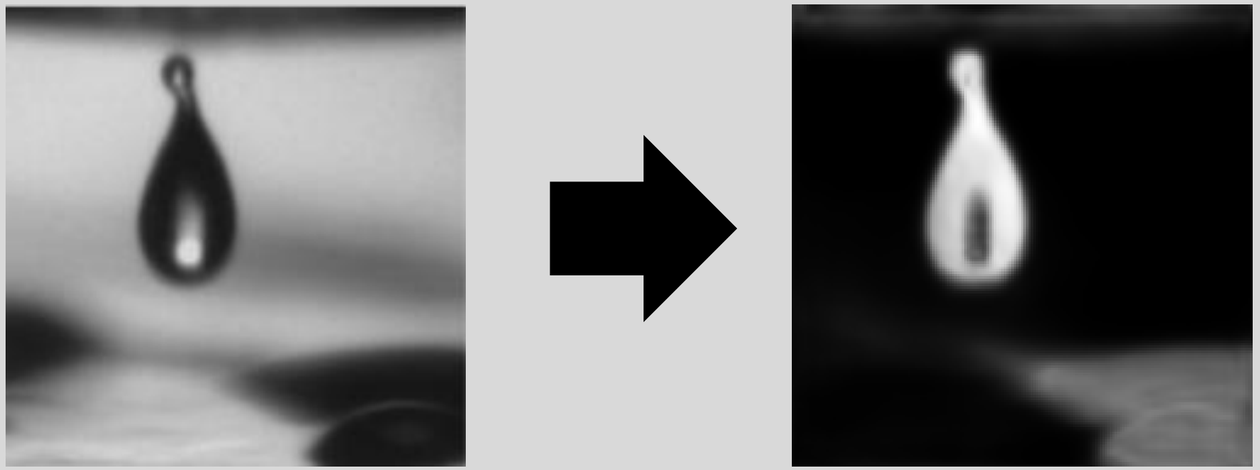
**Tropfen von ca. 0,06–0,5 mm · zu klein und zu schnell fürs Auge**

**ein falscher Tropfen = Ausschuss**

# Ausgangslage und Ziel

Tropfensegmentierung & Edge-KI NPU-Deployment

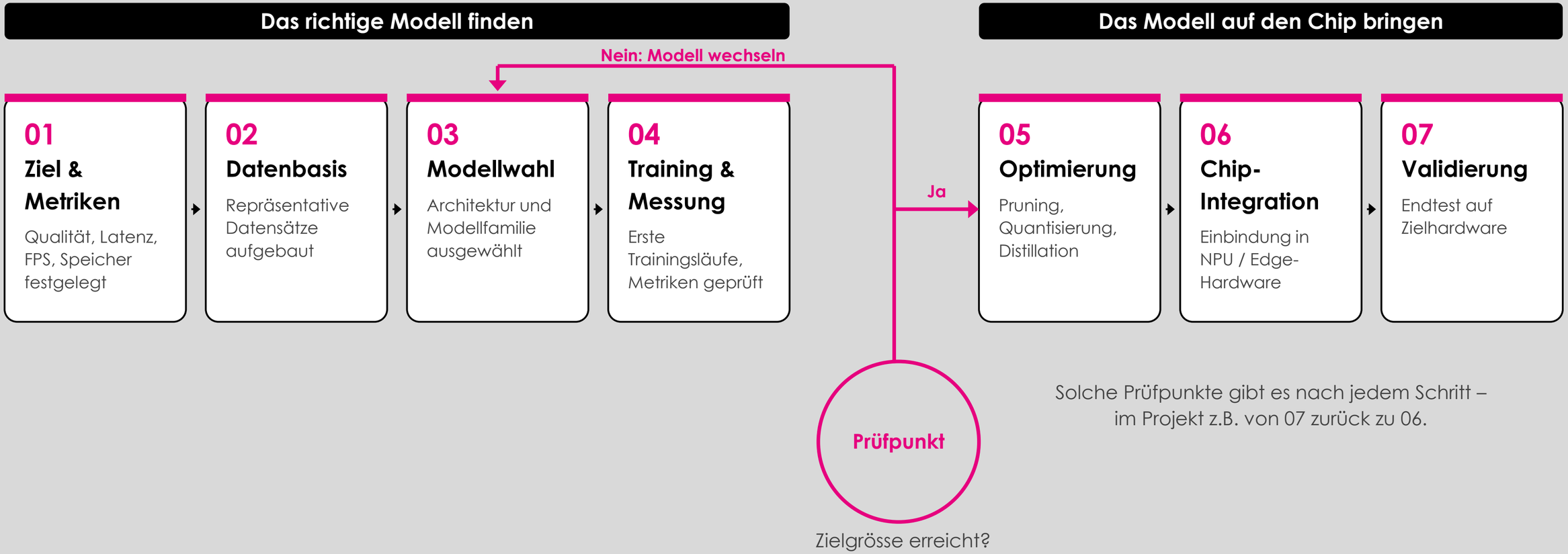
- **Automatische Tröpfchensegmentierung** aus aufgenommenen Bildern
- **Optimierung** für Echtzeit-Edge-Inferenz
- **Migration auf CPU+NPU von Edge GPU**: Jetson → DX-M1 NPU + Host CPU



**Ziel ist es, die Segmentierungsqualität beizubehalten und gleichzeitig Rechenaufwand, Energieverbrauch und Gesamtkosten deutlich zu senken.**

# Vorgehen: Jeder Schritt muss sich beweisen

Engineering mit gemessenen Zwischenzielen – nicht Trial and Error




Solche Prüfpunkte gibt es nach jedem Schritt – im Projekt z.B. von 07 zurück zu 06.

**Zielgrössen vor dem ersten Trainingslauf festgelegt: Qualität · Latenz · FPS · Speicher**

# Zielmetriken & KI-Modellauswahl

## Projektanforderung

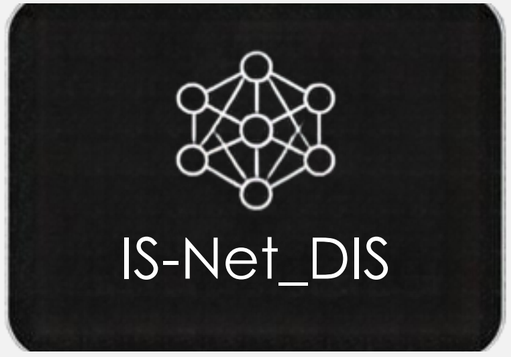
| Anforderung | Ziel |
|---|---|
| Eingabeauflösung | 192 x 192 |
| Inferenzleistung | ≥ 100 FPS |
| Zielhardware | DeepX NPU |
| Segmentierungsverhalten | Nur Tropfen im Vordergrund |
| Verhalten ohne Tropfen | Schwarze Ausgabemaske |

## Basismodell

| Kenngrösse | Aktuelles Basismodell @192x192 | Bedeutung |
|---|---|---|
| Parameter | 44.00 M | Grosses Modell (Speicher) |
| MACs | 5.61 G | Hohe Rechenauslastung für ein Edge-Ziel |
| FLOPs | 11.2 G | Bestätigt den Bedarf an Architektur-Optimierung |

### Hardwarekonfiguration

| | |
|---|---|
| **Board** | Grinn AstraOneSBC-1680 |
| **Architektur** | ARM64 / AArch64 |
| **KI-Beschleuniger** | DEEPX-M1 NPU |
| **Systemspeicher** | ~4 GB LPDDR5 (nur NPU) |
| **Speichergeschwindigkeit** | 5600 MT/s |
| **Betriebssystem** | Embedded Linux (Yocto) |





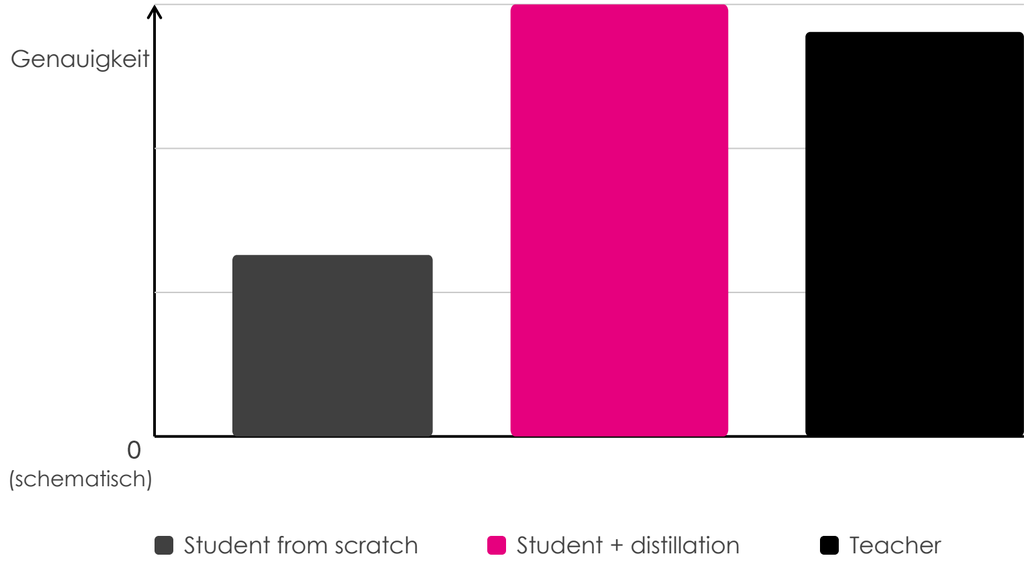

# Wissenstransfer und Pruning

Das grosse Modell lehrt das kleine – dann entfernen wir Rechenaufwand für die Hardware

01

## Knowledge distillation

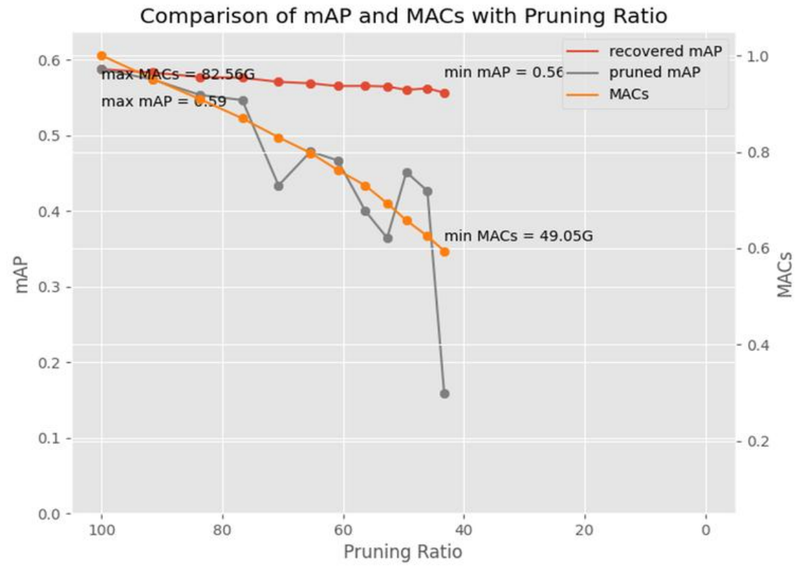
Genauigkeit von Student und Teacher – Genauigkeit Metriken F1/Dice/IoU




02

## Pruning and fine-tuning

Genauigkeit und FPS über die Pruning-Schritte, mit Erholung nach dem Fine-Tuning. Beispiel:

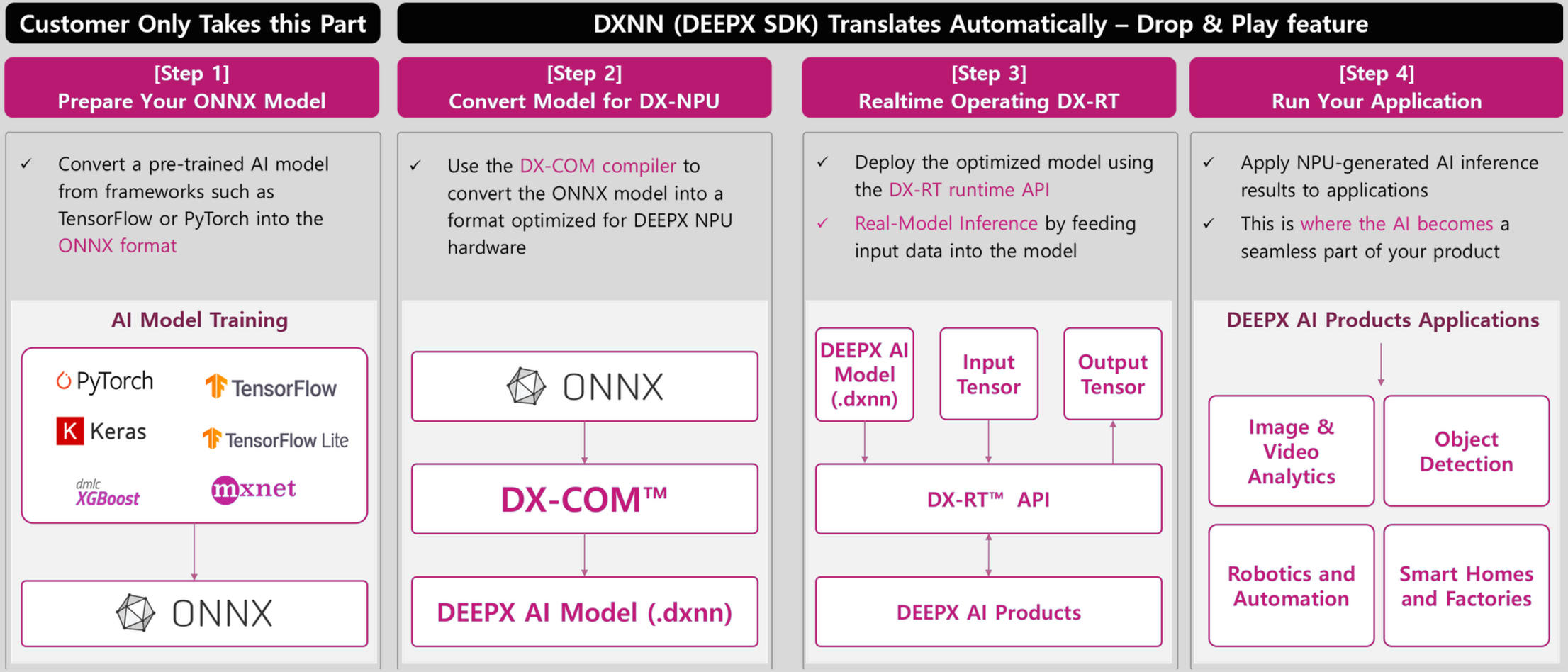



**Ein effizientes und trotzdem genaues Studentmodell**

Idealerweise lernt der Student vom Teacher und übertrifft ihn trotz geringerem Rechenaufwand sogar.

# Auf dem Chip: Quantisierung und Deployment

Das Modell für den NPU-Chip kompilieren



# Auf dem Chip: Benchmark

## Gleiche NPU, 3,4× unterschiedlicher Durchsatz

DeepX DX-M1, gleiches Segmentierungsmodell, gleiche Runtime – nur das Host-System ist anders

x86-Mini-PC · Intel Core Ultra 9

398 FPS

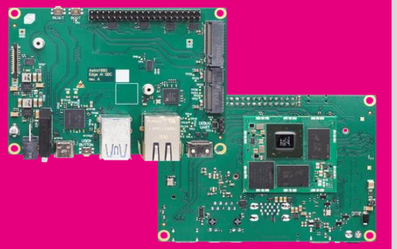
NPU-Auslastung ≈ 98 %



Arm-SBC · 4× Cortex-A73

116 FPS

NPU-Auslastung ≈ 20 %



44,0M

5,61 GMAC

IS-Net_DIS

→

1,16M

2,12 GMAC

“lite” IS-Net_DIS

x86-Host: **+43 % FPS**



Arm-Host: **+0,6 % FPS**

**Gepruntes Modell: 38× weniger Parameter, 2,7× weniger Rechenaufwand, −30 % NPU-Zeit**

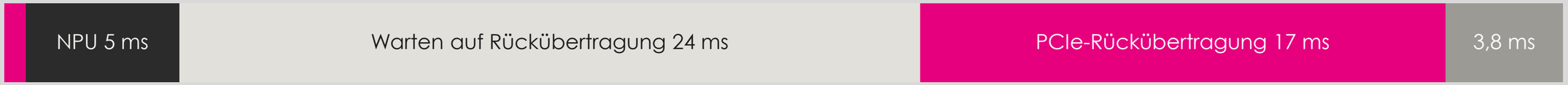
Eigene Messungen · DeepX DX-M1 (M.2) · DX-RT Benchmark und Profiler · Aug. 2026

# Auf dem Chip: Benchmark

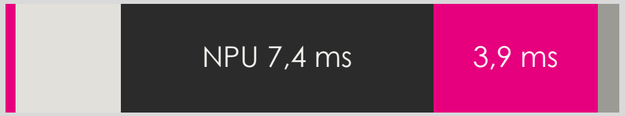
## Wohin die Zeit pro Bild geht

Ursache: 7,1 MB Ausgabedaten pro Bild über PCIe – für eine Maske von 0,15 MB

Arm-SBC: 51 ms pro Bild (116 FPS)




x86-Host: 15 ms pro Bild (398 FPS), gleicher Massstab, rund 3,7ms für Warten und CPU-Nachverarbeitung




NPU-Inferenz · PCIe-Transfer · Warten · CPU-Nachverarbeitung

**Ausgabetensoren**

Grösse, Datentyp und Layout bestimmen die PCIe-Last

**PCIe-Anbindung**

Generation und Lanes des M.2-Steckplatzes

**Host-CPU**

Nachverarbeitung, DMA, Speicherzugriffe

**Runtime**

Queues, Puffer, Core-Scheduling

Eigene Messungen · DeepX DX-M1 (M.2) · DX-RT Benchmark und Profiler · Aug. 2026

# Zusammenfassung

endrich

# Zusammenfassung

1. **Edge-KI lohnt sich, wenn das Problem es verlangt**
   Latenz, Ausfallsicherheit, Datensouveränität, Kosten im Dauerbetrieb, aber mit höherem Entwicklungsaufwand, der sich über Stückzahlen rechnet.

2. **Aufgabe und Modell bestimmen die Hardware, nicht die TOPS**
   Früh mit dem eigenen Modell auf echter Hardware messen.

3. **Wenn möglich – von Anfang an für die Zielhardware entwickeln**
   Mit Erfahrung geht es direkt auf die NPU – und jedes Projekt wird schneller.

4. **Ziele messbar machen – vor dem ersten Training**
   Qualität, Latenz, FPS und Speicher festlegen, dann jeden Schritt daran messen.



**Die Technologie entwickelt sich rasant**
Wahrnehmung ist heute die Grundlage, neuere Modelle kommen bereits auf die Edge

# Fragen?

**Andi Purnomo**
*Field Application Engineer*

+49 (0) 151 27646064
a.purnomo@endrich.com

