

# Von der Vogel-Kamera zum eigenen KI-Modell

Souveränes Machine Learning auf Open-Source-Hardware

Roland Imme und Adrian Imme  
[roimme@mailbox.org](mailto:roimme@mailbox.org)

## Motivation und Problemstellung

Standard-KI-Modelle wie Google Vision oder vortrainierte COCO-Datasets versagen bei der Erkennung europäischer Gartenvögel – eine Kohlmeise wird zum asiatischen Fasan, ein Rotkehlchen zur undefinierten „kleinen Vogelart“. Diese Ungenauigkeit ist symptomatisch für eine Abhängigkeit von proprietären Cloud-Services und deren Trainingsdaten, die primär auf kommerzielle Anwendungsfälle ausgerichtet sind.

**Souveräne Digitalität** bedeutet in diesem Kontext: vollständige Kontrolle über Daten, Modelle und Infrastruktur – von der Hardware-Aufnahme bis zum produktiven Inference-System. Dieser Vortrag zeigt, wie diese Souveränität praktisch umgesetzt werden kann.

## Das Projekt-Ökosystem

Was als einfaches Vogelhaus-Monitoring begann, entwickelte sich zu einem kompletten Open-Source-Ökosystem für Custom-Machine-Learning:

**vogel-kamera-linux** [github.com/kamera-linux/vogel-kamera-linux](https://github.com/kamera-linux/vogel-kamera-linux)

Automatisches Kamerasystem auf Raspberry Pi 5 mit AI Kit (Hailo-8L NPU). Aufnahme in Cinema 4K und 120fps-Zeitlupe, motion-triggered Recording, Echtzeit-Klassifizierung mit Custom-Modellen.

**vogel-model-trainer** [github.com/kamera-linux/vogel-model-trainer](https://github.com/kamera-linux/vogel-model-trainer)

YOLOv8-basierte Extraktion von Trainingsdaten aus Videoaufnahmen. Automatische Frame-Analyse, Vogel-Detection, Cropping und Dataset-Organisation für TensorFlow/PyTorch.

**vogel-video-analyzer** [github.com/kamera-linux/vogel-video-analyzer](https://github.com/kamera-linux/vogel-video-analyzer)

Automatische Qualitätskontrolle und Video-Analyse. Erkennung unscharfer Bilder, Duplikate, unvollständige Objekte – sichert Dataset-Qualität für effektives Training.

## Technische Pipeline: Von Pixeln zu Predictions

### 1. Hardware-Aufnahme (Edge-Computing)

- **Raspberry Pi 5** (8GB RAM) mit Hailo-8L AI Accelerator (13 TOPS)
- **Raspberry Pi Camera Module 3 Wide** (12MP, 120° FOV)
- **Lokale Speicherung:** Keine Cloud-Abhängigkeit, vollständige Datenkontrolle
- **SSH-Remote-Control:** Kamera-Parameter in Echtzeit anpassbar



Abbildung 1: \*  
Aktuelles Vogelhaus mit 2  
Kameras

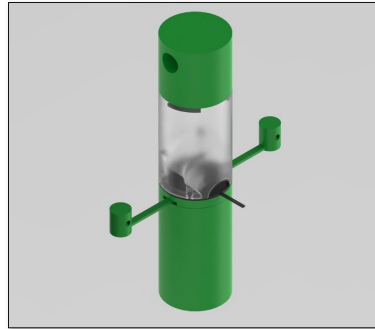


Abbildung 2: \*  
Zukünftiges Design (3D-Druck  
von Adrian Imme)



Abbildung 3: \*  
Frame aus 4K-Aufnahme

## 2. Dataset-Erstellung (Automatisiert)

- **YOLOv8-Detection:** Frame-by-Frame-Analyse der 4K-Videos
- **Intelligentes Cropping:** Vogel-Bounding-Boxes mit Kontext-Padding
- **Qualitätsfilterung:** Blur-Detection, Occlusion-Check, Duplikat-Entfernung
- **Ergebnis:** 5.000+ qualitätsgeprüfte Bilder für 8 Vogelarten

## 3. Model Training (Consumer Hardware)

- **Architecture:** EfficientNet-B0 (balanciert: Accuracy vs. Edge-Deployment)
- **Training-Setup:** TensorFlow auf Standard-Desktop-PC (keine Cloud-GPUs)
- **Data Augmentation:** Rotation, Flip, Brightness – robuste Generalisierung
- **Resultat:** 100% Validierungsgenauigkeit für deutsche Gartenvögel

## 4. Produktiv-Deployment (Echtzeitfähig)

- **TensorFlow Lite:** Modell-Konvertierung für Edge-Inference
- **Hailo-Optimierung:** Hardware-Acceleration für 30fps-Klassifizierung
- **Im Einsatz seit Oktober 2025** mit Live-Monitoring

## Souveränität durch Open Source

Der Vortrag demonstriert praktisch, wie **souveräne Digitalität** in AI-Projekten umsetzbar ist:

**Daten-Souveränität** Alle Aufnahmen lokal gespeichert und verarbeitet. Videos werden optional auf YouTube geteilt – aber die Rohdaten bleiben unter eigener Kontrolle

**Modell-Souveränität** Custom-Training für regionale Use-Cases statt generischer APIs

**Infrastruktur-Souveränität** Edge-Computing auf Open-Hardware (Raspberry Pi)

**Lizenz-Souveränität** Vollständig MIT-lizenziert – keine Vendor-Lock-ins

**Warum ist das relevant?** Die großen Cloud-AI-Anbieter trainieren primär für ihre Märkte (z.B. nordamerikanische oder asiatische Vogelarten). Wer europäische Spezies erkennen will, braucht entweder teure Custom-API-Calls – oder baut sein eigenes System. Letzteres ist mit Open-Source-Tools nicht nur möglich, sondern auch wirtschaftlich sinnvoll.

# Herausforderungen & Lessons Learned

## Technische Hürden

- **SSH-Stabilität:** Raspberry Pi in Outdoor-Umgebung – Lösung: Watchdog-Scripts, VPN-Fallbacks
- **Hintergrundentfernung:** Bewegte Äste/Blätter verwirren YOLOv8 – Lösung: Temporal Filtering
- **Class Imbalance:** Manche Vögel selten am Futterhaus – Lösung: Targeted Recording Sessions

## Iterativer Trainingsprozess

1. **Version 1:** Generisches COCO-Modell → 40% Accuracy
2. **Version 2:** Transfer Learning von iNaturalist → 75% Accuracy
3. **Version 3:** Custom-Dataset (1.000 Bilder) → 90% Accuracy
4. **Version 4:** Erweitertes Dataset (5.000 Bilder) + Augmentation → **100% Validation Accuracy**

## Praktische Vorführung

Der Vortrag beinhaltet:

- **Live-Videos** von echten Vogelhaus-Aufnahmen (4K-Qualität, ausgewählte Videos auch auf YouTube)
- **Code-Walkthrough** durch die YOLOv8-Extraction-Pipeline
- **Training-Metriken** und Confusion-Matrix-Analyse
- **3D-Druck-Hardware** vorgestellt von Adrian Imme (Konstruktion und Fertigung)
- **Real-Time-Demo** des produktiven Systems (falls Netzwerk verfügbar)
- **QR-Codes** zu allen drei GitHub-Repositories für direkten Zugriff

## Zielgruppe & Vorkenntnisse

**Geeignet für:**

- ML/AI-Einsteiger, die praktische Edge-AI-Projekte kennenlernen wollen
- Erfahrene Entwickler, die souveräne Alternativen zu Cloud-AI suchen
- Maker und Hardware-Enthusiasten (Raspberry Pi, Kamera-Technik)
- Open-Source-Advocates interessiert an vollständigen Projekt-Ökosystemen

**Vorkenntnisse:** Grundlegendes Verständnis von Python und Programmierkonzepten hilfreich, aber nicht zwingend. Alle ML-Technologien werden im Kontext erklärt.

## Webseiten & Literatur

- **Projekt-Repositories:** [github.com/kamera-linux](https://github.com/kamera-linux) (alle drei Projekte)
- **Raspberry Pi AI Kit:** [raspberrypi.com/products/ai-kit](https://raspberrypi.com/products/ai-kit)
- **YOLOv8 Documentation:** [docs.ultralytics.com](https://docs.ultralytics.com)
- **TensorFlow Lite:** [tensorflow.org/lite](https://tensorflow.org/lite)
- **EfficientNet Paper:** Tan & Le (2019): „EfficientNet: Rethinking Model Scaling for CNNs”

## Format & Dauer

- **Standard:** 30-45 Minuten Vortrag + 10 Min. Q&A
- **Technik-Bedarf:** HDMI-Beamer, Internet für Live-Demo (optional)

---

Alle Projekte sind **Open Source** (MIT License) und seit **Oktober 2025** produktiv im Einsatz.

**Referenten:** Roland Imme (Software/KI) und Adrian Imme (Hardware/3D-Druck)

**Kontakt:** [roimme@mailbox.org](mailto:roimme@mailbox.org)