

Interdisziplinäres Projekt

# Elektronische Vogelscheuche

Elektrotechnik/Informatik/MCD

WS 2024/2025

Herr Prof. Dr. Ringbeck



# Inhalte



1. Organisation
2. Komponenten
3. Schaltplan
4. Herausforderungen
5. Gehäuse
6. Modelle und Software
7. Raspberry Setup/ Installierte Software
8. Bilderkennung - Classification
9. Bilderkennung - Object Detection
10. Algorithmus Funktionsweise
11. Webcrawler -> Dataset
12. Dokumentation (Github)
13. Herausforderungen / Ausblick
14. Live Demo

# Organisation

## Ablauf:

1. Kommunikationskanal eingerichtet
  - Discord Server mit verschiedenen Kanälen
    - Allgemein
    - Informatiker
    - E-Techniker
    - Komponenten
  - Meetings nur bei Bedarf
    - Möglichst demokratisches Vorgehen (Abstimmungen/Umfragen)
    - Spontane Termine, um an der Vogelscheuche zu arbeiten
2. Zeitplan inkl. Meilensteine und Puffer erstellt
3. Teameinteilung vorgenommen
4. Entwicklung



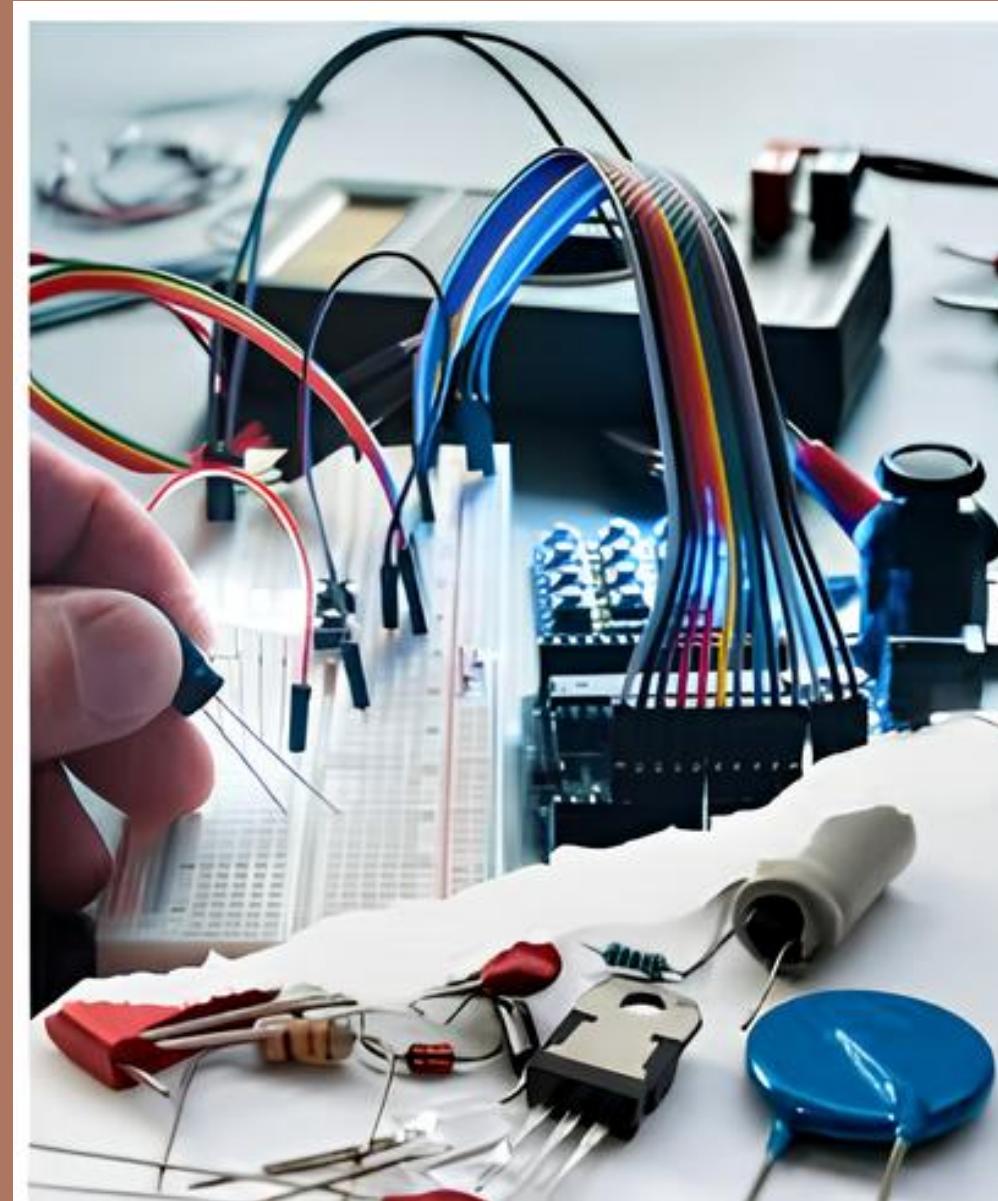
# Komponenten

- Raspberry Pi 5 mit AI-Hat (Hailo 8L Chip)
- Kamera zur Bildaufnahme (Modell 3)
- PIR-Sensor zur Bewegungserkennung
- Sirene für den Alarm
- F5305S Transistor
- Pin Verlängerung für den Raspberry pi
- Kühler zur Wärmeabfuhr
- Netzteil zur Versorgung vom Raspberry pi und Aktor.



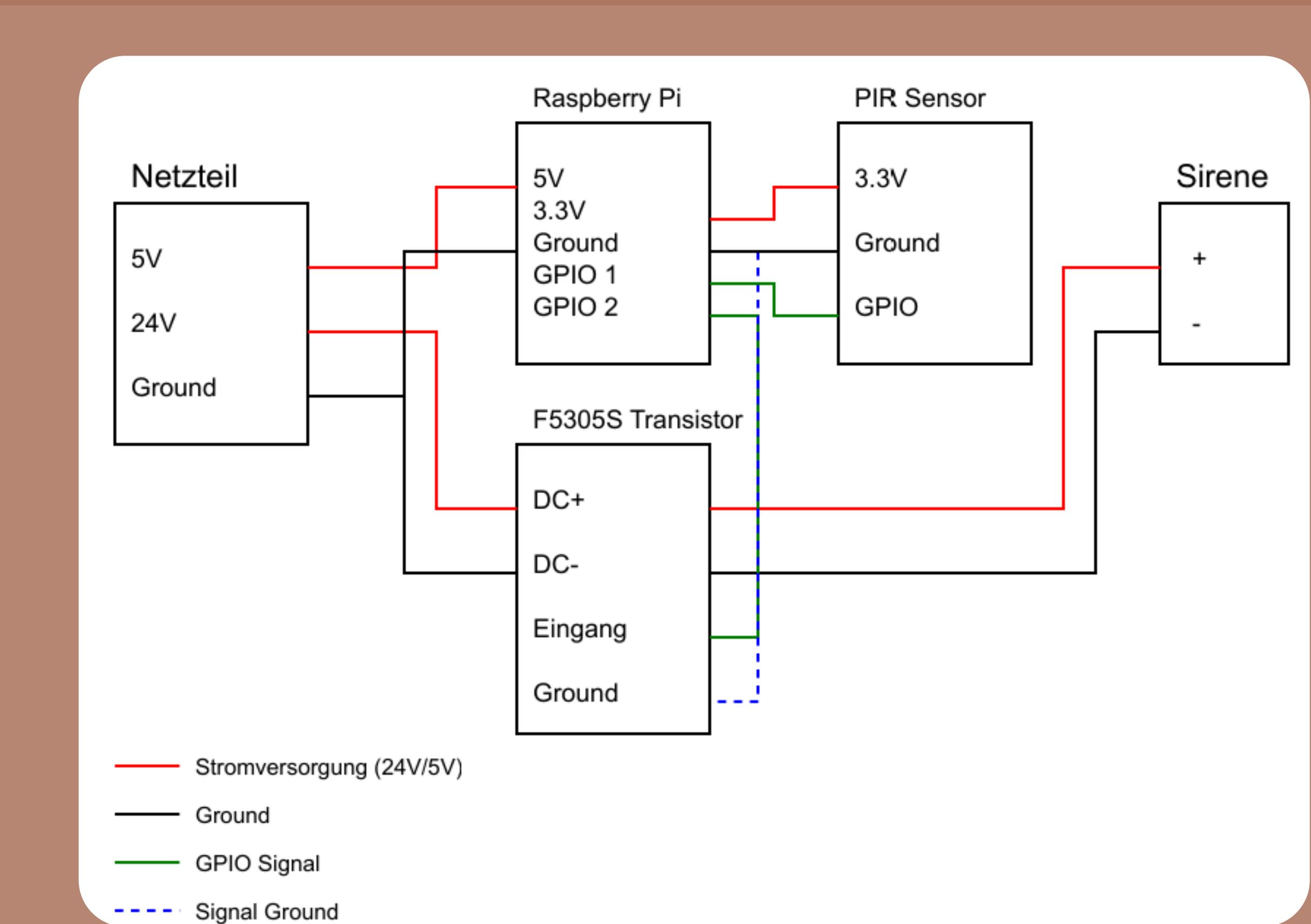
# Herausforderungen

- Die passenden Schaltungen für das Projekt finden.
- Untersuchen ob eine Versorgung mit Raspberry pi möglich ist.
- Die Raspberry pi pins können nicht genügend Strom liefern um alle Komponenten zu betreiben.
- Schaltungen direkt vom Raspberry pi zu versorgen kann die Leistung und Stabilität beeinträchtigen.
- Um eine Leistungsprobleme zu vermeiden, muss eine externe Stromquelle verwendet werden.

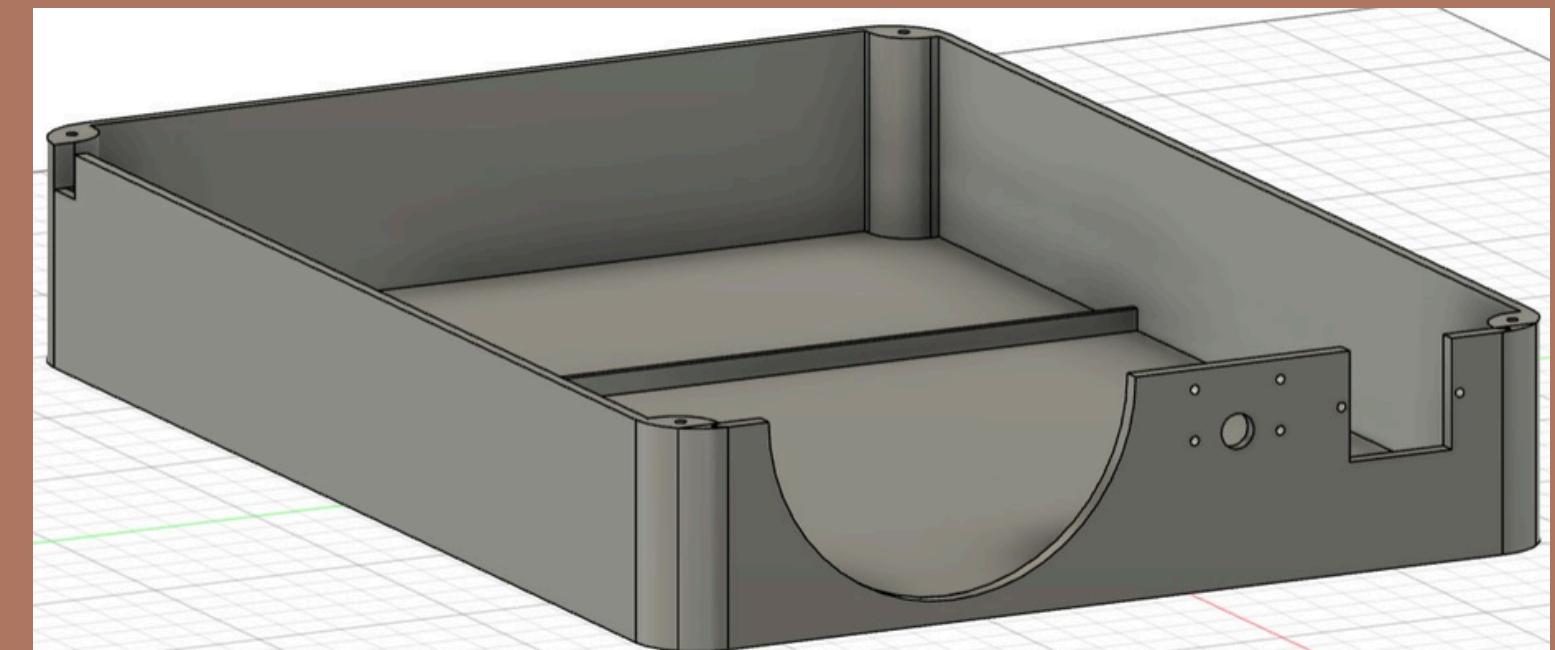
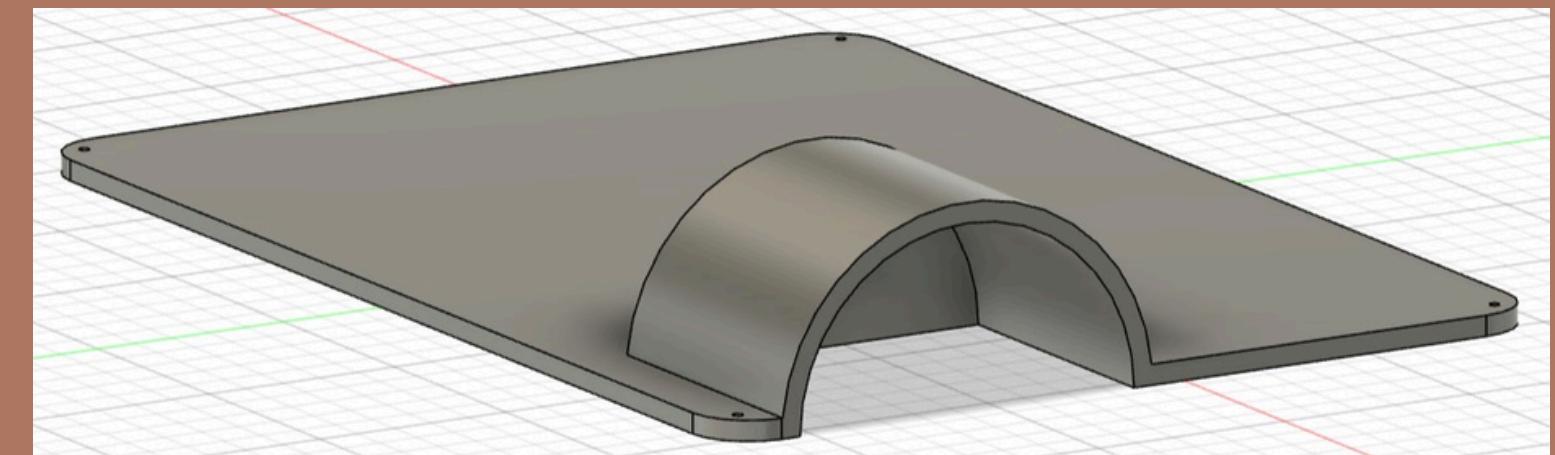
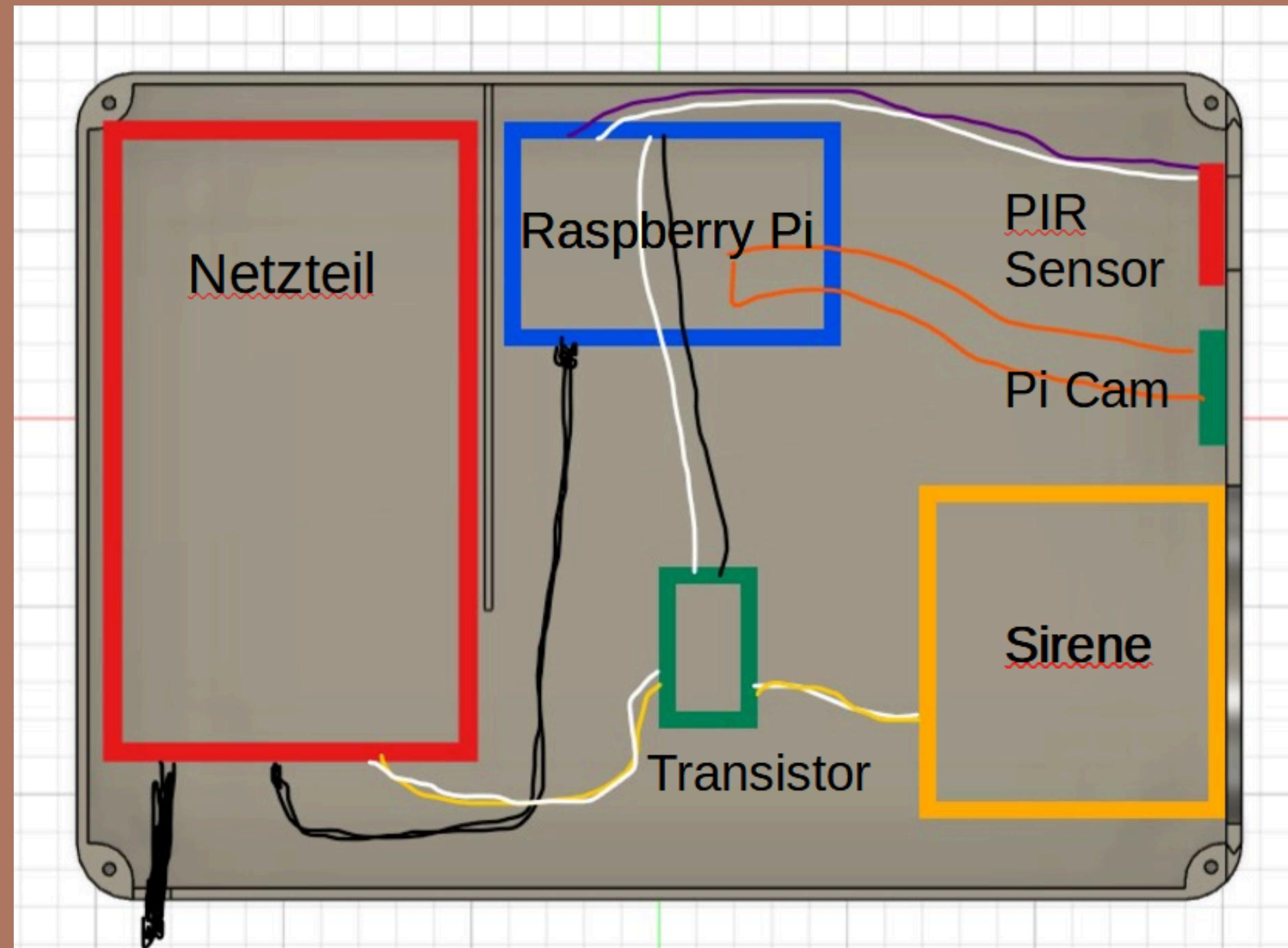


# Schaltplan

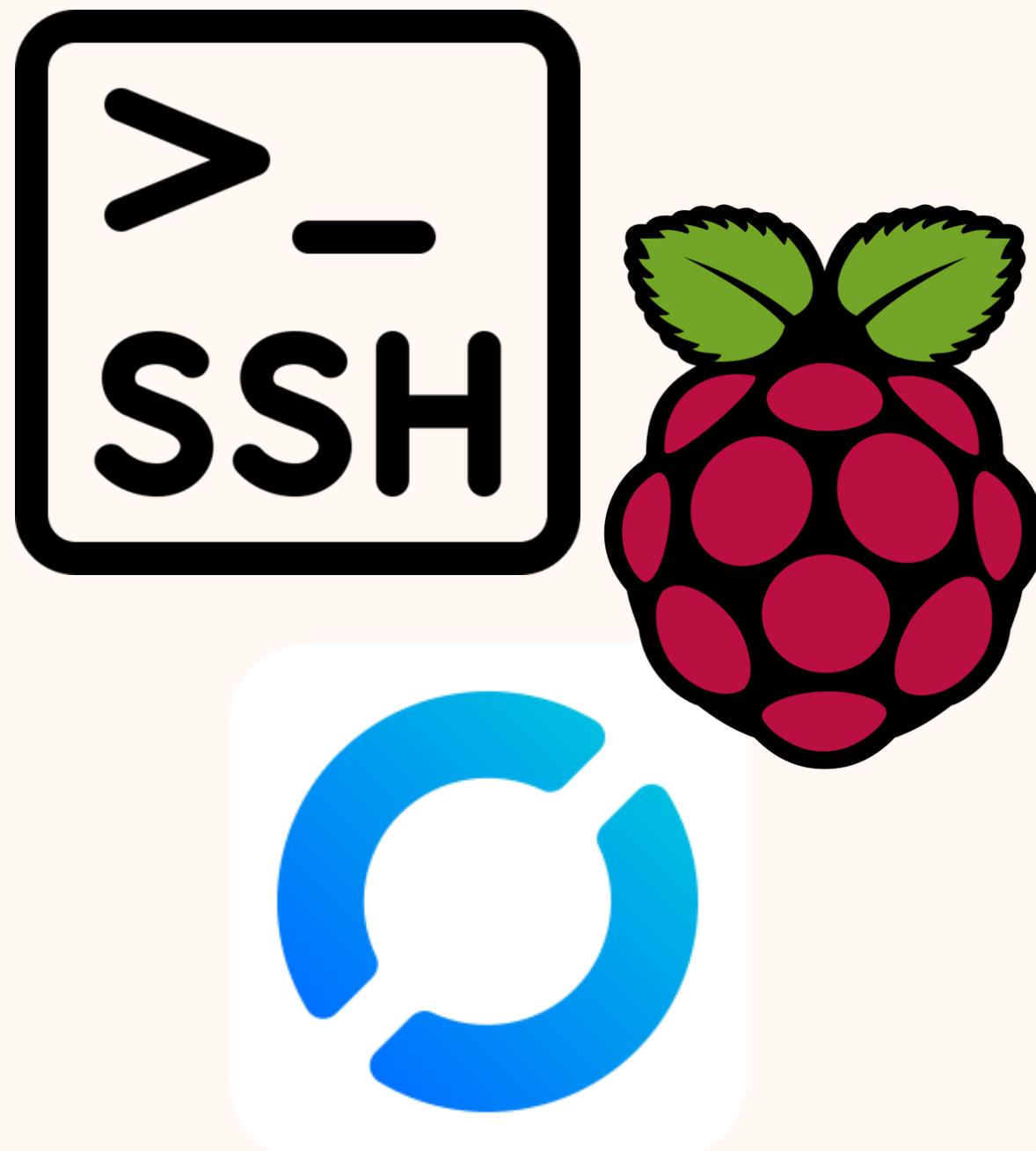
- Duo Netzteil zur Versorgung vom Raspberry sowie der Sirene
- PIR Sensor um den Stromverbrauch niedrig zu halten, damit später ein Akku-Betrieb möglich ist
- Sirene um den Reiher effektiv zu verscheuchen
- F5305S Transistor zur Ausgangssteuering



# Gehäuse mit Fusion 360



# Raspberry Setup/ Installierte Software



## Installation Raspberry Pi & AI-Hat

- Installation von Raspberry Pi Os
- Zugriff über SSH, da kein Monitor vorhanden
- Anfangs Nutzung eines Handy-Hotspots
- Erfolgreiche Installation der Treiber
- Erster Test des KI-Hats

## Netzwerklösungen

- Wechsel von Hotspot zu Eduroam
  - Herausforderung
    - Eduroam-Einrichtung unter Linux
    - Wpa Key installation

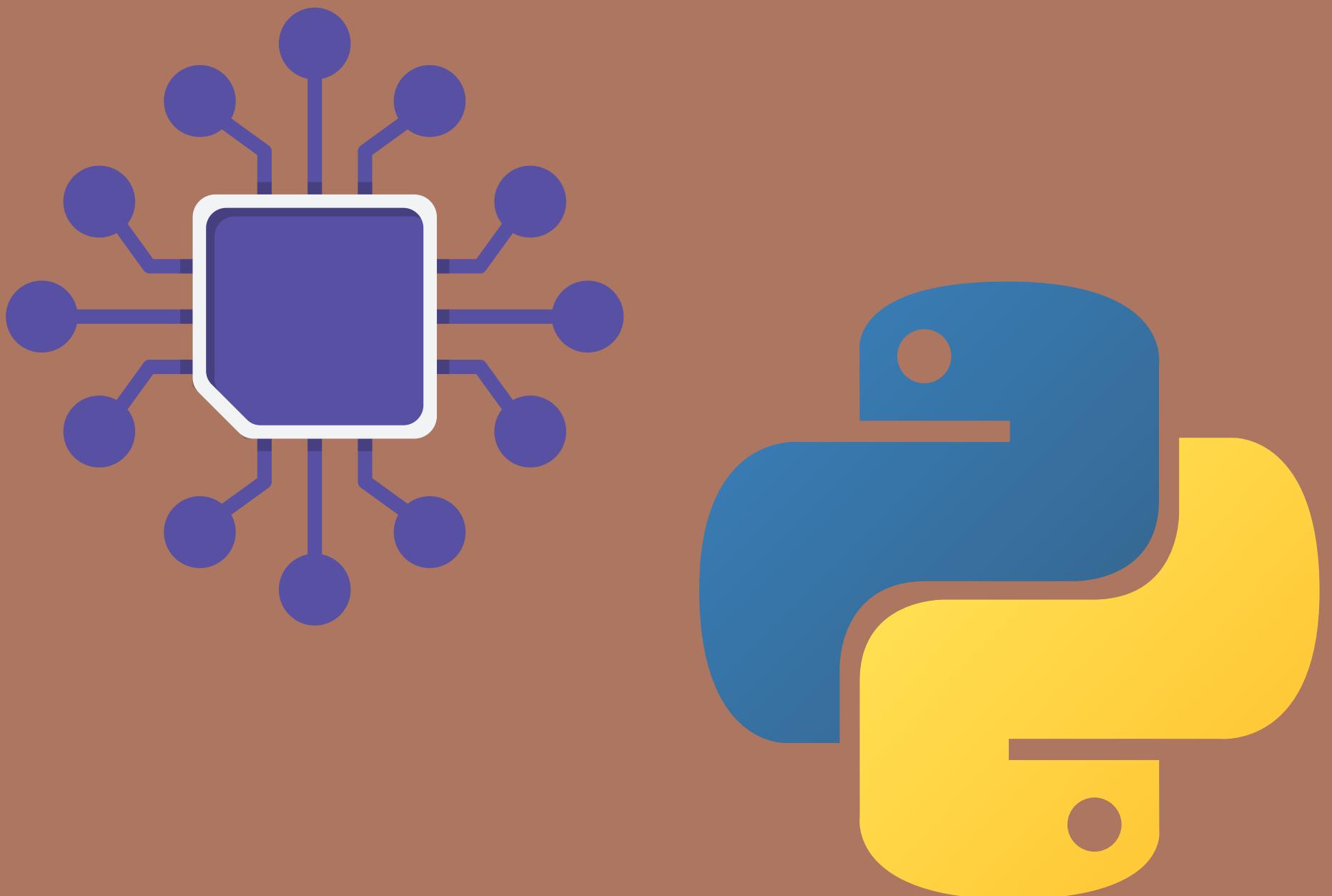
## Remote-Zugriff

- Einführung von Rustdesk
  - ermöglicht TCP - Port weitergaben
  - Remote Desktop Lösung
- Erleichterter Zugriff und Programmierstart

## Modelle und Software

- YOLO-Modell zur Objekterkennung
- ResNet-Modell zur Klassifizierung
- Modelle für Hailo-Prozessor optimiert
- Python Programm zum verbinden von Kamera, KI-Modellen und Sirene

H A I L O



# Bilderkennung - Classification



## Erste Schritte

- Intensive Recherche
- Fund eines Bird Classification Modells
  - erkannte 450 Vogelarten
  - Auf dem Laptop funktionierte es
    - langsam (30 Sek/Bild)

## Modellkonvertierung für den KI-Hat

- Konvertierung von Tensorflow-Modell ins .hef-Format
- Hailo Dataflow Compiler benötigt
  - NVIDIA-Grafikkarte, Ubuntu Linux, CUDA/CUDNN
  - Lief auf einem Laborrechner von Reke
- Erfolgreiche Konvertierung nach mehreren Versuchen
  - **Herausforderung:**
    - Modell funktionierte nicht auf dem KI-Hat
    - keine sinnvollen Ergebnisse

## Wechsel zum Hailo-Tutorial-Modell

- Test des Hailo-Tutorial Modells
- Funktionierte auf dem KI-Hat
- erkannte Fischreiher
- **Entscheidung:** Nutzung des Tutorial-Modells

# Bilderkennung - Object Detection

## Modell

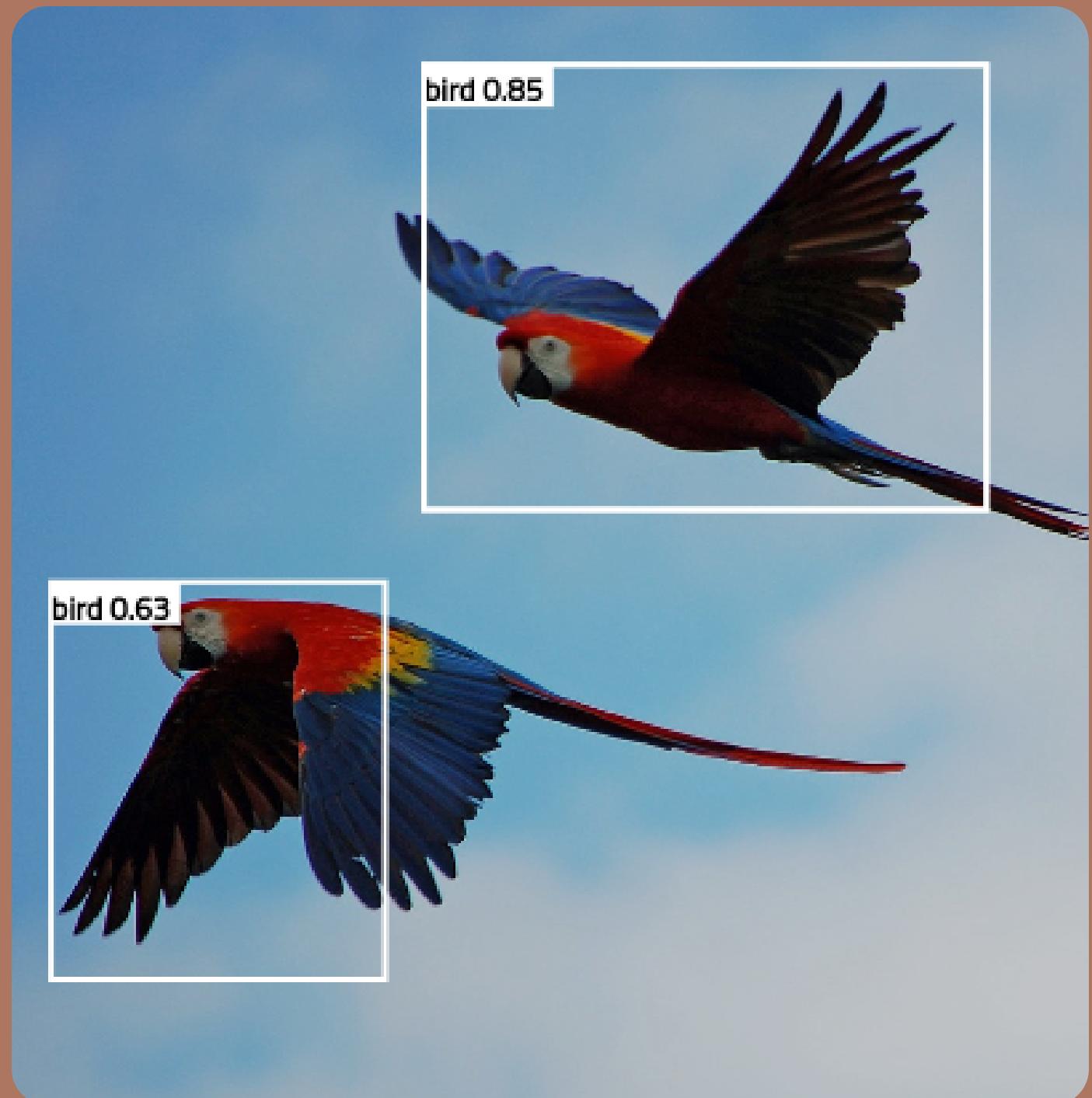
- Yolo v8
- bereitgestellt von Hailo
- Identifiziert alle Objekte im Bild
- Liefert Bounding Boxes und Wahrscheinlichkeiten

## Herausforderungen bei der Verknüpfung

- Unterschiedliche Anforderungen der Modelle
  - Bildgröße und RGB Struktur
- 2 Modelle auf einem Ki-Hat laufen lassen

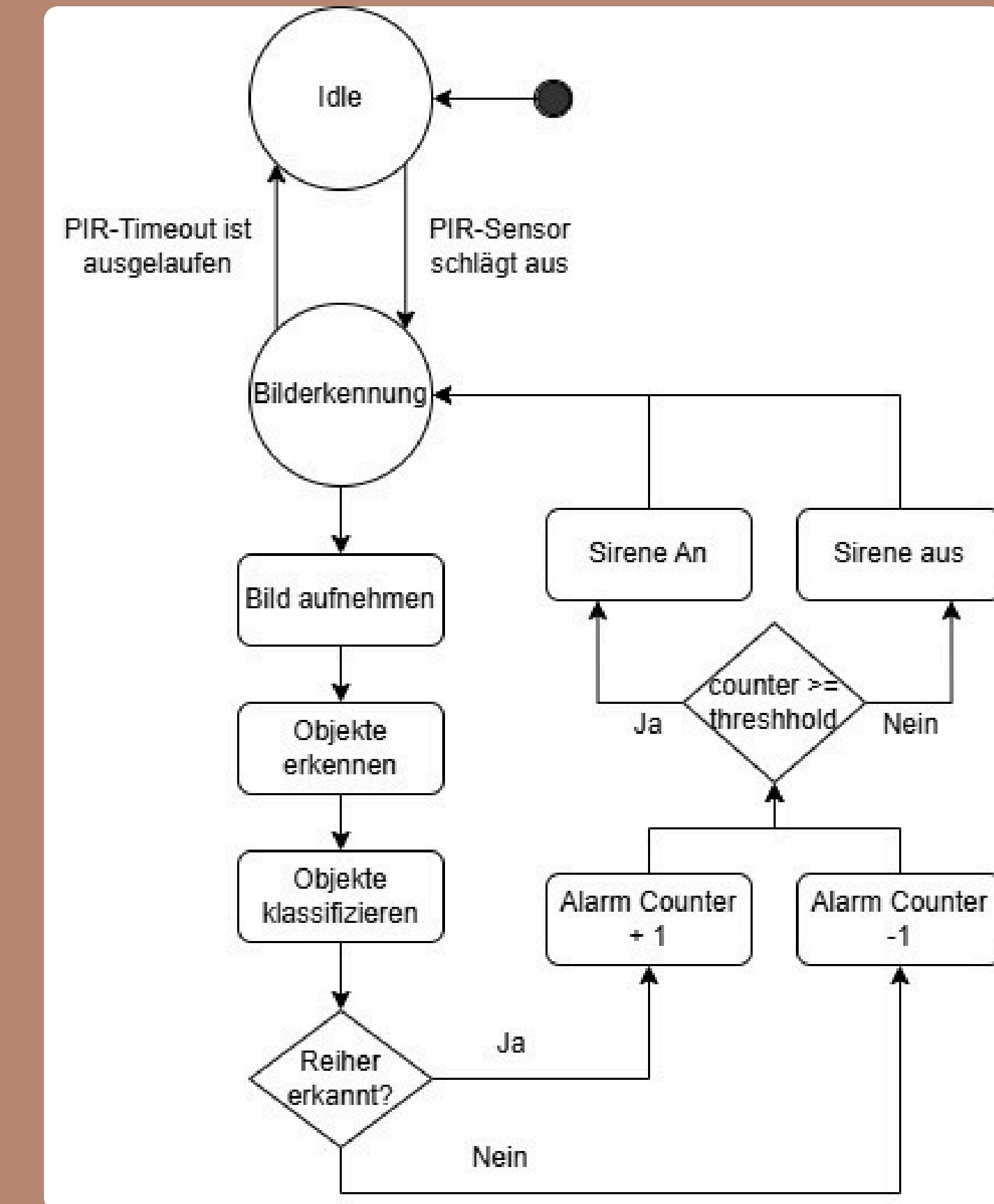
## Verknüpfung

- Python Klasse
  - Foto aufnehmen
  - Objekte erkennen
  - Objekte klassifizieren
  - Entscheiden ob Reiher erkannt wurde



# Funktionsweise

- Warten auf PIR-Sensor Signal
- Bilderkennung
  - Objekterkennung
  - Objektklassifizierung
  - Auswerten
- Alarm aktivieren



# Python-Bing-Fotos- Webcrawler: Erstellung eines Vogel-Datensatzes

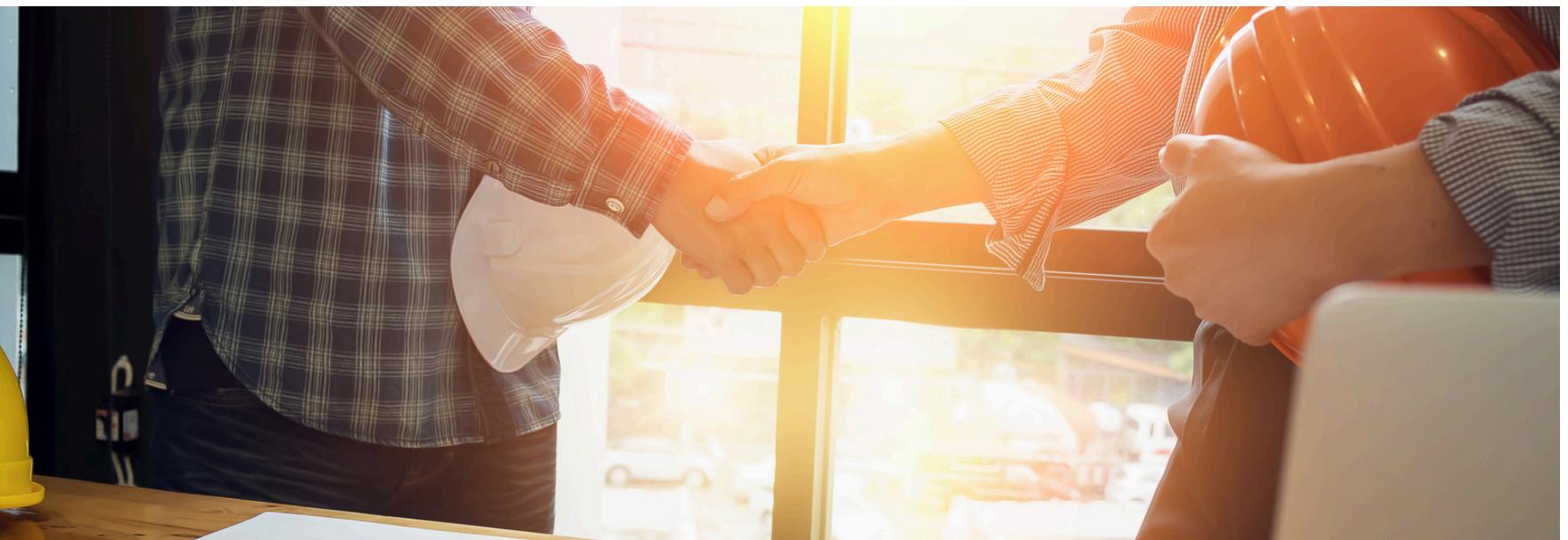
Für unser Projekt haben wir einen Python-Webcrawler benutzt, der mithilfe der Bing-Suche automatisch Bilder sammelt. Ziel war es, einen Datensatz mit 2250 Bildern zu erstellen, der 450 Vogelarten umfasst (jeweils 5 Bilder pro Art). Ursprünglich war geplant, diesen Datensatz zur Optimierung eines ResNet-Modells zu verwenden, um die Bildklassifikation zu verbessern. Obwohl dieser Ansatz letztlich nicht weiterverfolgt wurde, steht der erstellte Datensatz auf unserem GitHub-Wiki zur Verfügung. Weitere Details dazu finden Sie auf der nächsten Folie.



# Dokumentation (Github)

Eine Anleitung, sowie eine umfangreiche Dokumentation zur Handhabung & Entwicklung der elektronischen Vogelscheuche ist auf Github unter diesem Link zu finden:

<https://github.com/tijavo/VogelscheuchelP>



# Probleme / Ausblick



## Implementierung

- Bird Classification Resnet ans laufen bekommen
- Unterschätzung der Komplexität der Bilderkennung

## Reflektion

- gleich verteilte Arbeitsverteilung innerhalb der Teams (Parallelisierung)
- mehr Gedanken bei der Hardwarewahl (Raspberry Pi < NVIDIA Jetson)
- Discord als Kommunikationsmedium
- Projekteitung

## Ausblick

- Interface mit Liveview der Vogelscheuche
- Verbesserung der Bilderkennung
- Bird Classification Resnet ans laufen bekommen
- KI Model selbst Trainieren

Elektronische Vogelscheuche

# Live Demo

Elektronische Vogelscheuche

# Dankeschön !

