

# Qualitätskontrolle mittels KI/ML

Elia „KopfKrieg“ @ GPN 22 / 2024-06-01

# Hinweis bzgl. Lizenz

Dieser Talk wird aufgezeichnet, daher gab es Einschränkungen bzgl. der Bilder, die ich zeigen kann

# Inhalt

- Wer bin ich, was erzähle ich, wie begann das ganze?
- Das erste Projekt – Proof of Concept
- Das zweite Projekt – Produktivbetrieb \o/
- Gründung des Digi-Teams
- Hardware
- Software
- Projekte & Instandhaltung
- Fakten & Fazit

# Wer bin ich?

- Elia „KopfKrieg“ (es/ihr • sie/ihr)
- Blog - <https://kopfkrieg.dev>
- Seit der Kindheit was mit Elektrotechnik/IT
- ~2008/2009 angefangen mit neuronalen Netzen
- Erster Vortrag auf einer Chaos-Veranstaltung, bitte lieb sein 🙏🙏
- Es wird sehr viel Info 😅

# Was erzähle ich heute?

Die Geschichte rund um „Qualitätssicherung mittels KI/ML“

Wie es dazu kam...

...und warum das auch noch funktioniert?!

# Wie es begann...

Irgendwann 2018/2019: „Hey, irgendwas mit Machine Learning wäre doch mal wieder nett“

\*schaut sich Tensorflow an, ist begeistert\*

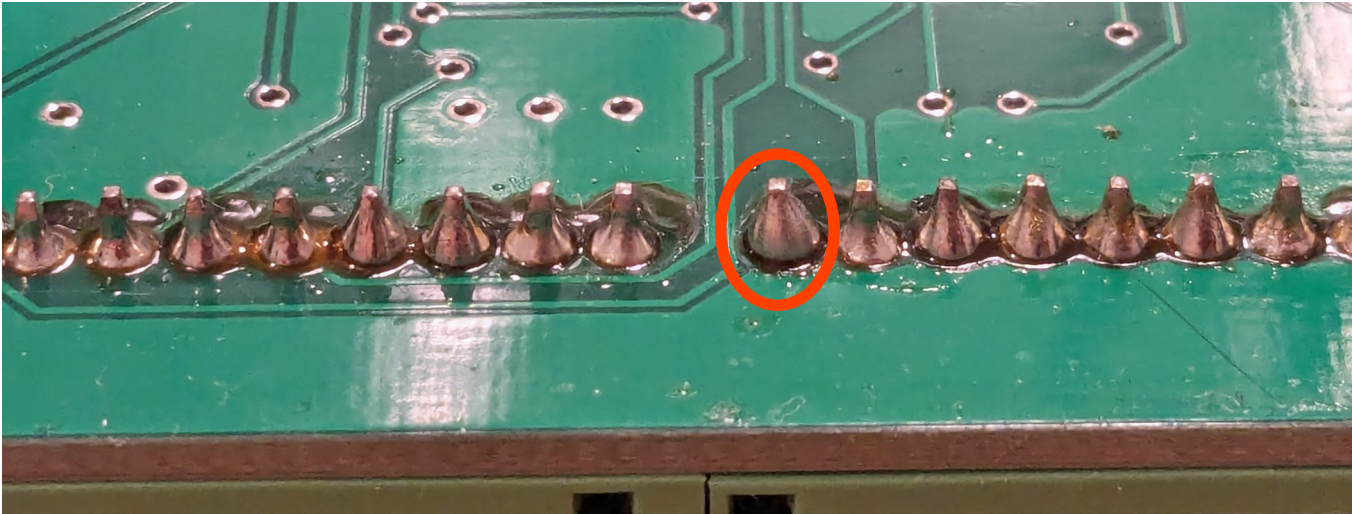
\*erzählt das auch noch rum\*

# Das erste Projekt... (Proof of Concept)



# Das erste Projekt... (Proof of Concept)

## Automatisierte Prüfung von Lötpins





# Das erste Projekt... (Proof of Concept)

## Automatisierte Prüfung von Lötpins

- Umsetzungszeit ca. 3 Wochen
- Gebrauchtes Laptop, Intel Core i5
- Logitech C525 (720p, günstig)

# Das erste Projekt... (Proof of Concept)

Proof of Concept hat funktioniert, aber es war nicht die beste Idee um damit anzufangen

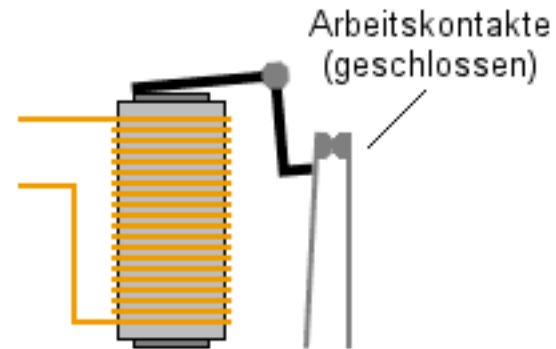
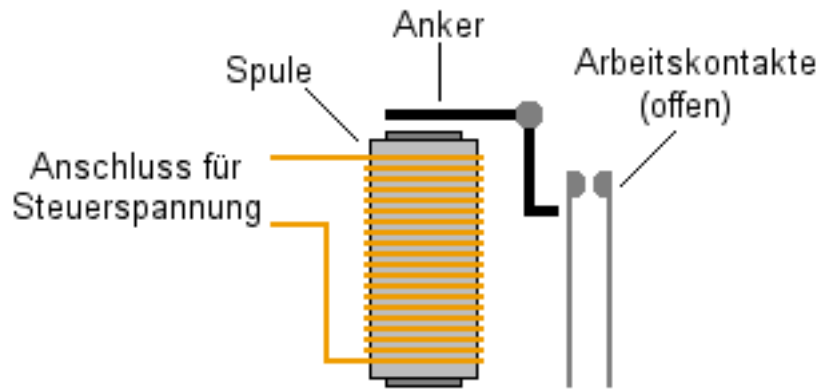
Umsetzung ca. Mitte 2019

# Das zweite Projekt...



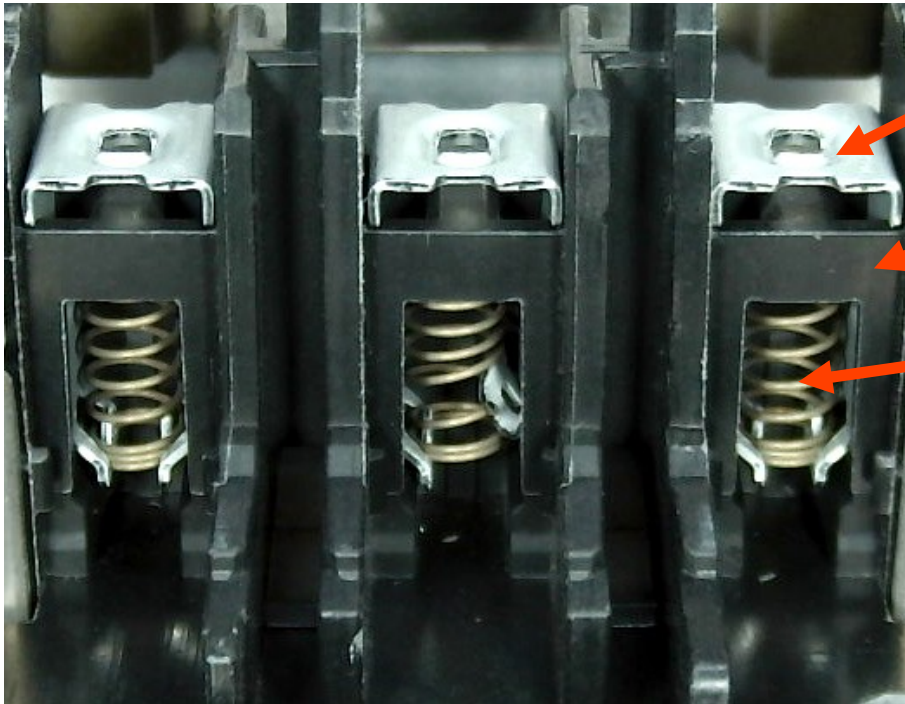
# Das zweite Projekt...

Automatisierte Prüfung von Kontaktträgern in Schützen (aka großen Relais) – ca. 500 verschiedene Varianten



# Das zweite Projekt...

## Automatisierte Prüfung von Kontakträgern



Federbügel

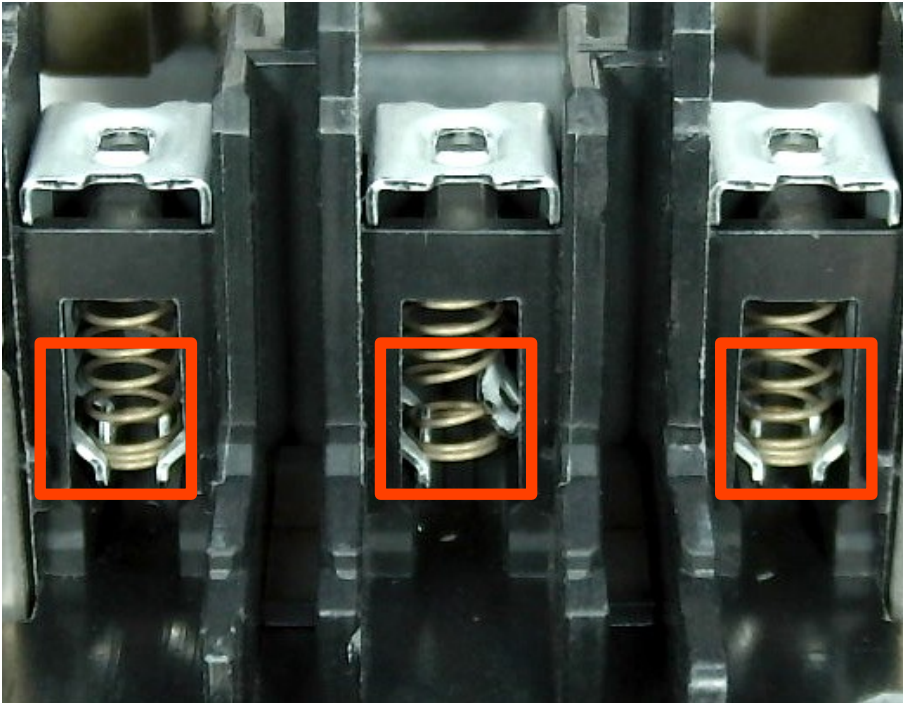
Kontakträger (KTR)

Feder

Von allem gibt es mehrere  
Varianten

# Das zweite Projekt...

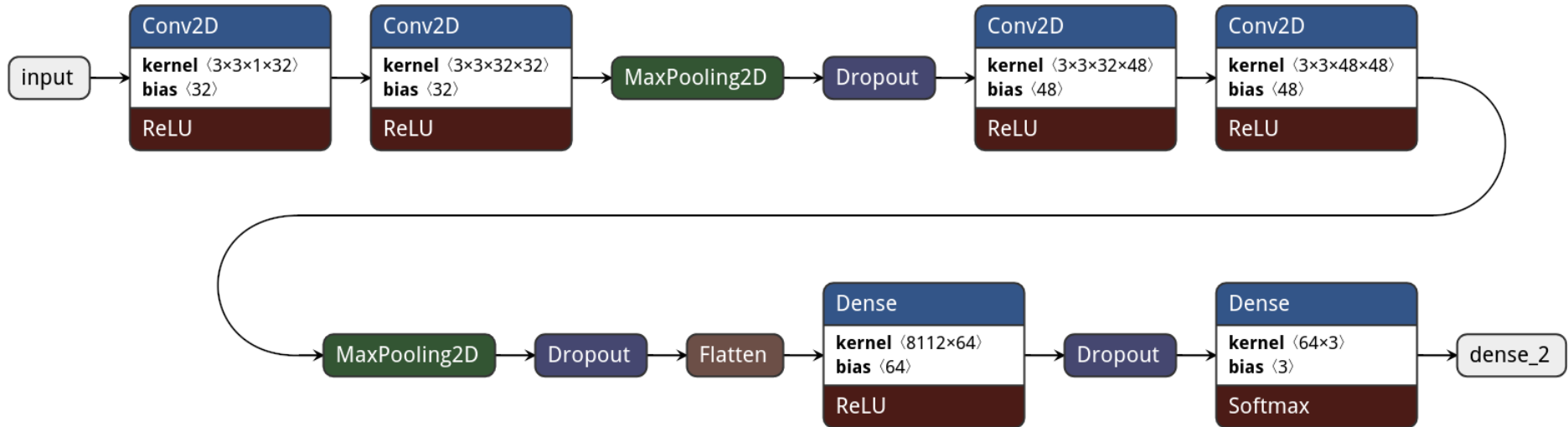
## Automatisierte Prüfung von Kontaktträgern



- 64x64 Pixel
- Graustufen

# Das zweite Projekt...

## Automatisierte Prüfung von Kontaktträgern



# Das zweite Projekt...

## Automatisierte Prüfung von Kontaktträgern

- Raspberry Pi 3 + MicroSD-Karte
- Logitech C525
- 3D-gedrucktes Gehäuse
- Datenübertragung zur SPS via GPIO-Pins
- Zykluszeit ~1 Sekunde
- Prüft täglich tausende von Kontaktträgern



# Das zweite Projekt...

System ist seit Ende 2019 produktiv

Nach zwei Jahren versagte der Autofokus der Kamera

# Gründung des Digi-Teams



# Gründung des Digi-Teams

- Team aus 4,5 Personen
- „Technologietreiber\*in“ (später: „Data Analyst“)
- Bereiche Künstliche Intelligenz, 3D-Druck, Robotik, Automatisierung („PC Bot“)

# Gründung des Digi-Teams

- Übergangsweise Projektleitung des Bereichs KI
- Vernetzung mit anderen Entwickler\*innen & Hochschulen
- Full Stack. Und zwar so richtig
  - Hardware (SoC/Raspberry Pi, Kamera, Objektiv, Beleuchtung, etc.)
    - Am liebsten Consumer-Hardware
  - Software (Betriebssystem, KI, Schnittstellen zur SPS)
    - Am besten kostenlos
  - Weiterentwicklung des ganzen, plus Instandhaltung, plus neue Projekte umsetzen

# Hardware



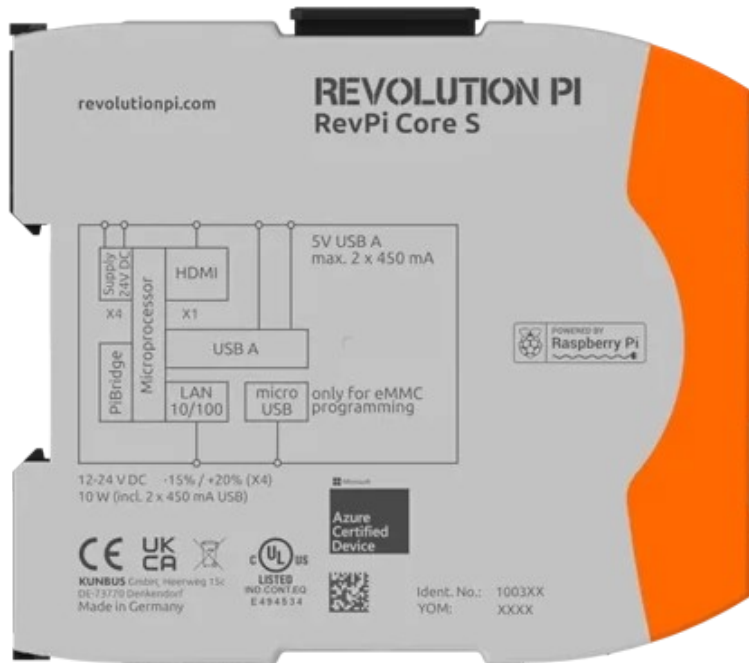
# Hardware

Was kann man anstelle eines Raspberry Pi 3 nutzen?

Einen Raspberry Pi! In anderer Verpackung...

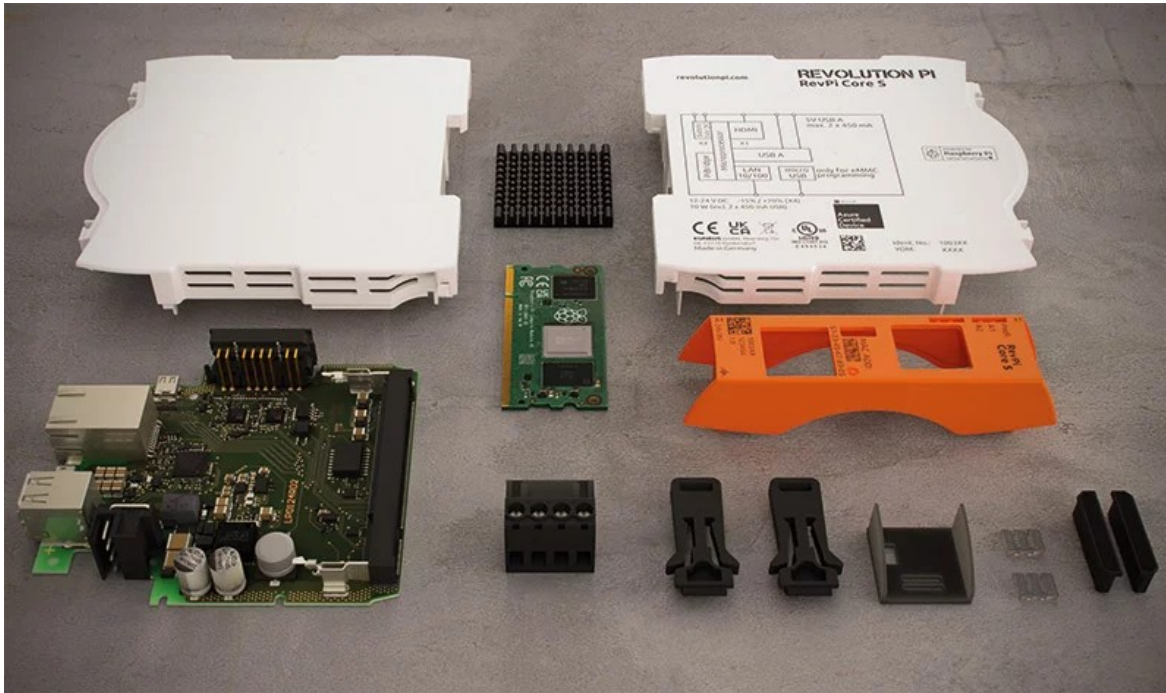
# Hardware

## Revolution Pi, Kunbus GmbH



# Hardware

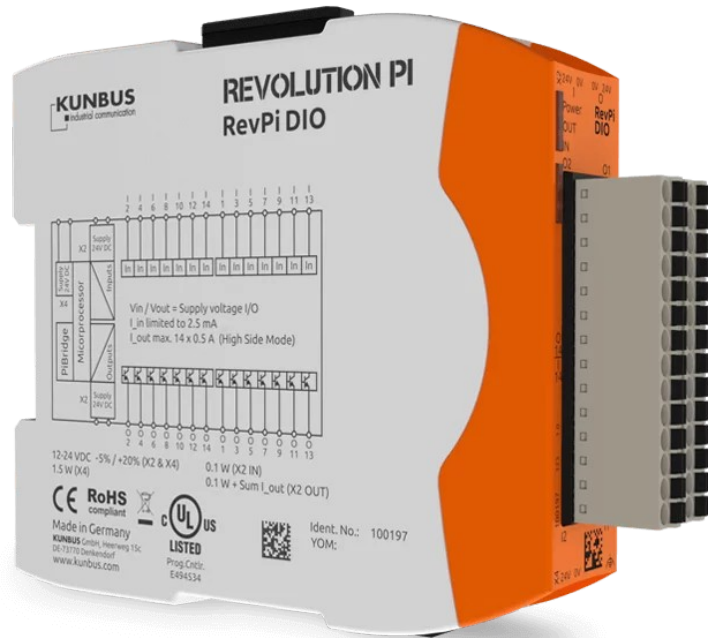
## Revolution Pi, Kunbus GmbH





# Hardware

## Revolution Pi, Kunbus GmbH



# Hardware

Was kann man anstelle eine Logitech C525 nutzen?

Puh, schwierig...

# Hardware

Was kann man anstelle eine Logitech C525 nutzen?

- Logitech Brio – 4K Auflösung, besseres Objektiv
- Das günstigste, was man auf Amazon findet (ELP)
- Industriehardware
  - Baumer GmbH
  - Keyence Deutschland GmbH
- Protokolle: USB, GigE Vision

# Hardware

## Logitech Brio

- Zu „intelligent“
- Autofokus, Weißabgleich, Kontrast, Schärfe, etc.
- Halterung schwer möglich

Wir wollen aber eine möglichst einfache Kamera

# Hardware

Das günstigste, was man auf Amazon findet (ELP)

- Full HD Auflösung
- Nicht sonderlich intelligent
- Lässt sich hervorragend via libuvc ansteuern
  - OpenCV ist glücklich, und ich auch
  - Support hängt von der Plattform und der exakten Kamera ab

# Hardware

## Industriehardware - Baumer GmbH, Keyence Deutschland

- Baumer GmbH liefert Kameras, Objektive, sowie Treiber
  - Keyence Deutschland GmbH fiel wg. Treiber raus
- Ansteuerung via GigE Vision, d.h. zweiter Netzwerkport nötig
  - Abstraktion der Ansteuerung weg von OpenCV, hin zu nativem Treiber
  - Bandbreite?! Angeblich ein 10G Port, es reicht aber auch 1G

# Hardware

## Beleuchtung?!

- Das erste umgesetzte Projekt nutzte einen WS2801 LED-Streifen, angesteuert via GPIO-Pins
- Blitzauslöser sind zu komplex, schwierig abzustimmen
- Einfache Beleuchtung via fertiger LED-Module, plus Treiber (mit integrierter Helligkeitsregelung)
  - Einfach anschließen, fertig

# Software





# Software

Betriebssystem

# Software

## Betriebssystem

- Aufgrund der Hardware vorgegeben
- Stark modifiziertes Raspbian, mit Treiber des Revolution Pi
- CI-artige Pipeline für schnellere Deployments
- „Ordentliches“ Event-basiertes System, mit systemd-Units die Geräte erkennen, den Service starten, etc.

# Software

KI-Plattform – Kontinuierliche Weiterentwicklung über ~3,5 Jahre

# Software

## KI-Plattform (1/5)

- Konnte anfangs nur IO/NIO → mehrere Varianten
- Concurrency! So viel wie möglich, und alles auf einmal, inklusive IOs
- Projektverwaltung
- REST API (Node-RED)
- Bessere Modelle & Trainingsroutinen

# Software

## KI-Plattform (2/5)

- Finetuning!
- EXIF-Metadaten in Bildern
  - Bildbetrachter (auch wg. Datensätze generieren)
- Erkennung von Kamera- bzw. Bildfehlern (Helligkeit, Kontrast, Schärfe)
- Datenaugmentierung

# Software

## KI-Plattform (3/5)

- Tensorflow 1.x → 2.x Migration
- Wechsel von H5 auf TFLite Format
- Bessere Ansteuerung der Kamera, höhere Bildqualität
- So viel wegautomatisiert wie möglich
- So viel Tests wie möglich

# Software

## KI-Plattform (4/5)

- Mehr Abstraktion → läuft auf jedem OS mit Python 3.x, OpenCV und Tensorflow, mit jeder Kamera/Eingabequelle (bspw. Wärmebildkamera)
- Absurde Grenzen: Bilder bis 2 Gigapixel möglich, über 500 Million Classifier (Bildbereiche)

# Software

## KI-Plattform (5/5)

- Integration von allem, was irgendwie praktisch war
  - Tesseract OCR für Texterkennung
  - Data Matrix und QR-Code Leser
  - Alles, was OpenCV sonst noch erlaubt...
    - ...und auch Scikit-learn, SciPy, was man halt so kennt



# Software

Schnittstellen zur SPS

# Software

## Schnittstellen zur SPS

- Entweder via Input/Outputs...
- ...oder via HTTP/Websocket-Verbindung

Alternativ auch direkt auf dem Gerät via Terminal-Interface

# Projekte & Instandhaltung



# Projekte & Instandhaltung

Weit mehr als 30 Projekte evaluiert, jedes Projekt mit dutzenden Anforderungen

- KI/ML ist immer noch der primäre Faktor
- IO/NIO, Varianten, Kerbenerkennung, Federn, Farbabweichungen, Orientierung von Bauteilen, Erkennung von losen Teilen, Druckfehlern (Masterarbeit über Autoencoder)
- Es muss sich nicht einmal mehr um Bilder handeln

# Projekte & Instandhaltung

Weit mehr als 30 Projekte evaluiert, jedes Projekt mit dutzenden Anforderungen

- So ziemlich alle davon wären umsetzbar
- In der Nähe von Schweißgeräten gibt's lustige Fehler
- Je mehr Projekte, desto aufwendiger die Instandhaltung
  - Instandhaltung („Elektriker“) dürfen deswegen mittlerweile auch mitmachen

# Jetzt gib doch mal Fakten & Fazit



# Jetzt gibt doch mal Fakten & Fazit Preis

## Eigenentwicklung

- Revolution Pi Core ~300 €
  - Revolution Pi DIO ~160 €
  - Kamera ~70€ - 2500 €
  - Beleuchtung (optional) ~400 €
  - USB-Stick, USB-Kabel ~40 €
- 570 €, mit Beleuchtung 970 €, mit teurer Kamera ~3400 €

# Jetzt gibt doch mal Fakten & Fazit Preis

Vergleichbares System namhafter Hersteller

- Schwarz/Weiß ~3.000 €
- Farbe ~10.000 €, ~15.000 € wenn man's unbedingt wissen will, 50.000 € für das teuerste, was ich bisher gesehen habe

Zuzüglich Wartungsverträge, Instandhaltung (extern), etc.



# Jetzt gibt doch mal Fakten & Fazit

## Technische Daten

- Zykluszeit ~350ms
- F1-Score beim ersten Projekt ~99,9948%
  - Cutoff für Umsetzung ist ~99,5%
  - Mensch liegt bei ~95%
- Bachelorarbeit über das Thema sagt ca. Faktor 100 genauer als herkömmliche Systeme
- Alle Modelle klein genug, dass man sie in Minuten bis wenigen Stunden auf einem Core i7 (8. Gen) neu berechnen kann, Finetuning in wenigen Minuten

# Jetzt gibt doch mal Fakten & Fazit

## Genauigkeit / F1-Score

- Modelle & Trainingsverfahren durchprobieren
- Gewichtungen nutzen – IO als NIO ist ok, NIO als IO nicht!
- Daten, Daten, und noch mehr Daten
  - Im Zweifel hilft Datenaugmentierung (Position, Rotation, Helligkeit, Kontrast)

# Jetzt gibt doch mal Fakten & Fazit Performance

- So einfach wie möglich
  - Profiler sind praktisch, bspw. um unnötige Konvertiervorgänge zu finden (BGR → RGB → BGR)
- Alles im RAM. Zu wenig RAM? Zram!
- Alles an Puffern deaktivieren, die man findet (Camera buffer)
- Einfache Modelle! Kein MobileNet, keine Vision Transformer (MobileViT)
  - Wir müssen keine 500 Tiere unterscheiden, am Ende fallen nur 2-3 relevante Bits raus

# Jetzt gibt doch mal Fakten & Fazit Und wofür nun?

Für Fehler, die nicht kritisch sind. Wenn Menschenleben davon abhängen, wird's unangenehm!

- Beschriftungsfehler
- Falsche Teile, die nicht auf Qualität gehen
- Lebensdauer
- Kratzer, Beschädigungen von außen

Vielen Dank!  
Fragen gerne jetzt :)

