ReSort

„Farben erkennen- Richtig trennen!
ReSort, Das System für jeden“

Übersicht

1. Zusammenfassung ReSort
2. Aktueller Stand
3. Technischer Ablauf und Implementierung
4. Praxisbeispiel Bildaufnahme
5. Praxisbeispiel Bild hochladen
6. Herausforderungen und Lösungsansätze
7. Bezug PoC (Audit 3)
8. Bezug priorisierte PoC
9. Zukunftsperspektiven und Weiterentwicklung
10. Fazit

Zusammenfassung ReSort

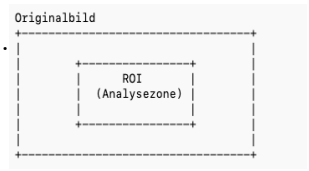
Ziel des Projekts:

Entwicklung einer mobilen Anwendung zur transparenten und barrierearmen Glasfarberkennung auf Basis regelbasierter Bildverarbeitung

Kernidee:

Analyse eines Glasbildes → Klassifikation → Konfidenzbewertung → konkrete Entsorgungsempfehlung

Zusammenfassung

1.


2.
```python
# 1. Grünes Glas
# Hue 28-90° (Grünbereich), mittlere bis hohe Sättigung
if 28 <= hue <= 90 and saturation > 20:  # Erweitert für knapp-grüne Farben
    confidence = min(1.0, (saturation / 100) * 0.6 + 0.4)  # Höhere Base-Konfidenz
    logger.info(f"→ Klassifiziert als: Glasflasche_Gruen (Konfidenz: {confidence:.2%})")
    return 'Glasflasche_Gruen', confidence
```

3.
```
"recycling_info": {
    "material": "Grünes Glas",
    "recycling_category": "Grüner Glascontainer",
    "instructions": "Deckel entfernen und in den grünen Glascontainer werfen. Pfandflaschen zum Automaten bringen.",
"message": "Flasche erkannt: Glasflasche_Gruen",
"processing_time_ms": 77.15,
"timestamp": "2026-01-15T15:49:19.009552"
```

Hier sieht man Ausschnitte der letzten Präsentation aus Audit 3 um einen kurzen Rückblick und eine Übersicht zum bisherigen Stand des Projektes zu schaffen.
Das Bild konnte zu dem Zeitpunkt nur über die Funktion “Bild hochladen“ getestet werden. Getestet wurde zu dem Zeitpunkt via Postman.
Das Bild wurde hochgeladen und geprüft. ROI war festgelegt, sodass genau dieser Abschnitt des Bildes in das Analyseverfahren gefallen ist. Dort wurden die HSV Werte abgeglichen (im Beispiel für Farbe Grün). Nach erfolgreicher Auswertung wurde via Postman dann die aufgeführte Antwort ausgegeben.

Aktueller Stand

- Entwicklung eines funktionsfähigen Prototyps als hybride Applikation
- Reduziertes und barrierearmes UI mit klarer und simpler Struktur
- Ergebnisdarstellung durch symbolische Icons zur Unterstützung der Barrierefreiheit
- Anzeige eines Konfidenzwertes zur Transparenz der Klassifikation
- Anzeige einer kontextbezogenen Entsorgungsempfehlung
- Anpassung der Region of Interest an den Flaschenboden (runde ROI-Struktur) sowohl für die Funktion der Bildaufnahme, als auch des Bildhochladens

Im Fokus bei der Frontend Entwicklung stand, dass die Oberfläche simple und leicht nachvollziehbar strukturiert wird. Dies war uns wichtig, um möglichst alle Altersgruppen für die Nutzung mit abzudecken. Das Gefühl und Bewusstsein zur korrekten Glastrennung soll gestärkt werden. Fokus liegt hierbei auf der Gesellschaftlichen, sowie wirtschaftlichen Relevanz. Ebenfalls sollen Menschen mit Farbsehschwäche bei ihrem Vorhaben der korrekten Glastrennung bestärkt und unterstützt werden. Bei der Ergebnisausgabe wurde daher gezielt mir Symbolen gearbeitet. Somit wird beispielsweise bei dem Ergebnis für Grünglas ein Blattsymbol oder bei braunem Glas eine Kaffeetasse angezeigt. Auch ist der Hintergrund in der jeweiligen Farbe leicht strukturiert. Beim Entwurf haben wir uns mit der Frage beschäftigt, wie sinnvoll die Verwendung von Icons in diesem Kontext wäre. Wir sind schnell zu dem Entschluss gekommen, dass sie aus unserer Sicht ein essenzielles visuelles Mittel sind. Nach weiterer Recherche wurde klar, dass sich auch das Spektrum der Farbsehschwäche in verschiedene Kategorien unterteilen lässt. Einige betroffene leiden bspw. Unter Rot- grün Schwäche und andere sehen keine Farben. Da viele jedoch Farben wahrnehmen oder sie einfach „anders“ sehen, können sie diese visuell mit Gegenständen oder Objekten assoziieren. So sehen die Betroffenen bspw. Nicht die Farbe Grün im herkömmlichen Sinne, wissen jedoch dass ein Blatt als grün gilt.

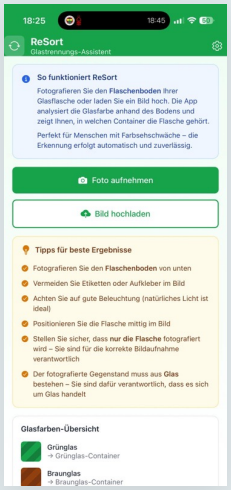
Der Konfidenzwert hatte sich im Zuge der Programmierung des Backends ergeben und bildet aus unserer Sicht einen relevanten Faktor. Dadurch, dass er bei der Ergebnisausgabe mit ausgegeben wird, wird dem Nutzer ein Gefühl im Hinblick auf Zuverlässigkeit und Korrektheit vermittelt. Bei unsicheren Ergebnissen wird dies demnach auch gekennzeichnet.
Die Entsorgungsempfehlungen sind derzeit grundsätzlich auf die Entsorgungsrichtlinien von Gummersbach ausgerichtet.
Da die meisten Flaschenböden rund sind, wurde die ROI rund ausgerichtet. Problem bei eckigem ROI ist, dass durch die Kanten zuviel der Hintergrundfarben mit aufgenommen wurden und daher die Ergebnisse stark beeinträchtigt wurden. Sollte man eine eckigere Flasche haben; dann ist auch in diesem Fall eine runde Ausrichtung sinnvoller und effektiver, da diese konkret in den mittleren Bereich des Flaschenbodens gezogen werden kann.

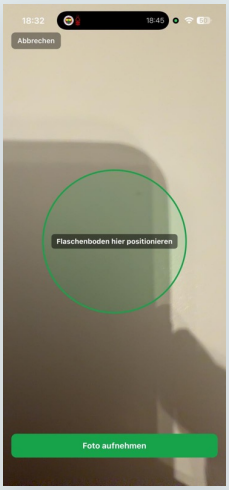
Technischer Ablauf und Implementierung

1. Backend-Implementierung mit Python (FastAPI) inklusive Analyse-Logik und REST-Endpunkten
2. Containerisierung mittels Docker
3. Frontend-Backend-Kommunikation über REST-API (Client-Server-Architektur)
4. UI-Konzeption mit Figma, Umsetzung mit React Native & Expo (JavaScript)
4. Mit Expo .apk Datei erstellt, um die App auf einem Android Smartphone ausführen zu können
5. Auf Smartphone übertragen

Expo; um durch Bibliothek den Code auf dem Handy durchzuführen.
Durch Docker war die Nachvollziehbarkeit der einzelnen Schritte gegeben und hat sich für die Testung und Durchführung des Projekts als hilfreich erwiesen.
Die Mockups wurden zunächst mit Figma erstellt. Fokus lag hier zunächst auf der Startseite. Wie bereits vorher angekündigt wollten wir vieles zunächst rein Kommunikativ regeln und weitergeben, sodass bspw. Auf der Startseite Kriterien und Hinweise für den Nutzer zu sehen sind. Wichtig war hier, dass der Nutzer darauf hingewiesen wird, dass es selbst dafür verantwortlich ist, dass es sich bei dem fotografierten Gegenstand wirklich um Glas handelt. Das System ist derzeit nicht auf dem Stand, Objekte zu erkennen und zu selektieren. Dies wäre jedoch ein Punkt der in der Zukunft noch angepasst werden könnte.

Praxisbeispiel Bildaufnahme



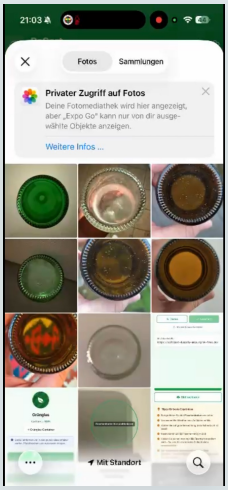


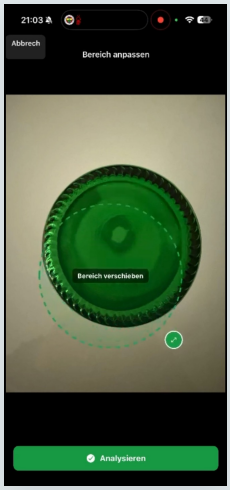
Hier sieht man die Startseite, sowie Frame der Bildaufnahme und die Ergebnisausgabe. Die Startseite weist eine kurze Anleitung, sowie Tipps und Anmerkungen für den Nutzer auf. Der Nutzer wählt zwischen der Bildaufnahme und alternativ dem Bild hochladen. Tippt man auf das erste, so kann man den Flaschenboden Problemlos abfotografieren. Der vorgegebene Frame ist rund. Ergebnisausgabe ist bewusst minimalistisch und kompakt gestaltet. Kurze und klare Aussagen sowie Anweisungen. Auch die Entsorgung bzw. das Zurückbringen von Pfandflaschen ist ein relevanter Bestandteil der Glasentsorgung und dessen Weiterverarbeitung. Daher erscheint bei jeder Ausgabe (bei einem Ergebnis) der Hinweis, dass Pfandflaschen zum Automaten gebracht werden sollen.

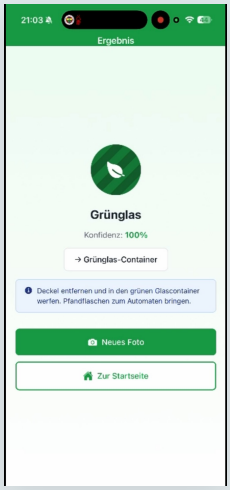
Praxisbeispiel Bild hochladen

Hier ein kurzes Video zum Bild hochladen.

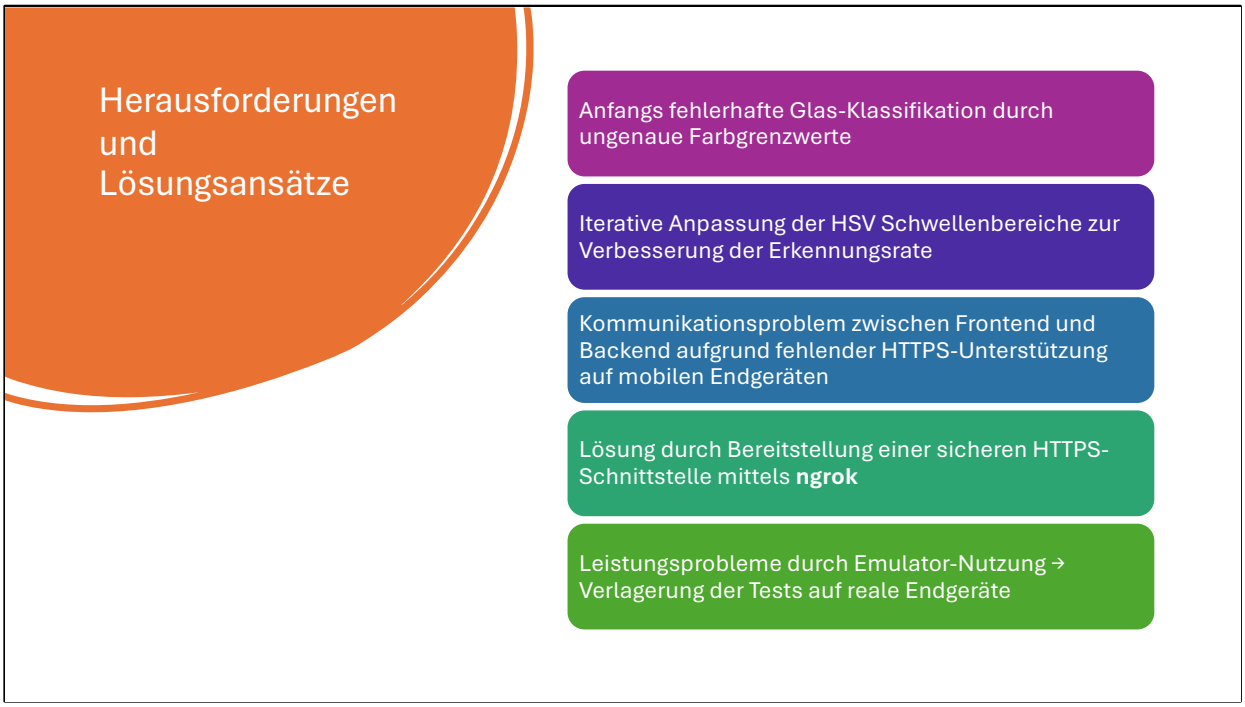
Praxisbeispiel Bild hochladen







Fallback; nochmals in Screenshot Form



Ein Problem mit dem wir uns beschäftigen mussten war die Testung bzw. Simulation von Front+Backend auf einem Endgerät. Die Kapazität der Laptops hat nicht ideal ausgereicht, sodass durch das Testen die Reaktionszeit der Geräte stark beeinträchtigt wurde. Ebenfalls war unser Ziel zu Beginn, den Prototypen lauffähig auf einem Smartphone präsentieren zu können. Nachdem verschiedene Ansätze getestet wurden, sind wir zu der, auf der Folien aufgeführten, Lösung gekommen.

Bezug PoC (Audit 3)

PoC	Status	Einschätzung
PoC 1 – Kamera, Bildaufnahme, ROI	**Abgedeckt**	vollständig umgesetzt
PoC 2 – Regelbasierte Farberkennung im HSV-Farbraum	**Abgedeckt**	vollständig umgesetzt
PoC 3 – Ergebnisausgabe, Transparenz und Barrierefreiheit	**Teils abgedeckt**	Im Backend vorhanden
PoC 4 – Regionale Entsorgung (Gummersbach)	**Teils Abgedeckt**	technisch umgesetzt
PoC 5 – Confidencewert & Ergebnisverwerfung	**Neu**	Aus Code ergeben
PoC 6 – API-Stabilität & Umgang mit Fehlern	**Neu**	Aus Code ergeben

Hinzugekommene PoCs
PoC 5 – Confidence & Qualitätsfilter
Vermeidung falscher Empfehlungen
Regelbasierte Ergebnissicherung

PoC 6 – API-Stabilität & Fehlerhandling
Dateityp-, Größen- und Formatprüfung
Saubere HTTP-Fehlercodes

Bezug priorisierte PoC

PoC	Status	Einschätzung
PoC 1 – Kamera, Bildaufnahme, ROI	**Abgedeckt**	vollständig umgesetzt
PoC 2 – Regelbasierte Farberkennung im HSV-Farbraum	**Abgedeckt**	vollständig umgesetzt
PoC 3 – Ergebnisausgabe, Transparenz und Barrierefreiheit	**Abgedeckt**	vollständig umgesetzt

Die aufgeführten PoC sind die PoC mit der höchsten Priorität. Diese bilden die Kernfunktionen des Systems ab und sind daher der Reihe nach Ausschlaggebend für die Erfolgreiche Durchführung der App.
Diese wurden um Prototypen nun vollständig umgesetzt. Im Punkt Barrierefreiheit wurde zunehmend Rücksicht genommen und versucht, die Barrierefreiheit auf verschiedenen Ebenen zu stützen und zu gewährleisten.
Weiteres wird in der nächsten Folie aufgeführt und könnte in der Zukunft optimiert werden.

Zukunftsperspektiven und Weiterentwicklung

- Weiterentwicklung der Farberkennung durch robustere Analyseverfahren (z. B. Differenzmodell mit erweitertem ROI)
- Implementierung einer audiobasierten Ausgabe zur Unterstützung stark sehbeeinträchtigter Nutzer
- Integration von Ortungsdiensten zur Anzeige nahegelegener Glascontainer
- Anbindung an lokale Entsorgungsrichtlinien mit automatisierter Aktualisierung
- Evtl. Objekterkennung um sicherzustellen, dass es sich wirklich um Glas handelt

Fazit

- Einfache und intuitive Bedienung für verschiedene Alters- und Nutzergruppen
- Transparente und nachvollziehbare Funktionsweise
- Gute Grundlage für eine mögliche Integration in bestehende Systeme
- Potenzial als Lern- und Sensibilisierungsinstrument für korrekte Glastrennung
- Weiterentwicklungsbedarf vorhanden, jedoch vielversprechende Zukunftsperspektive
- Bewusste funktionale Begrenzung im Rahmen des Moduls zur Sicherstellung der Umsetzbarkeit

Vielen Dank für Ihre Aufmerksamkeit

ReSort Farbanalyse

- **Hue (28–90°)** → Grünbereich im HSV-Farbraum
- **Saturation > 20** heißt die Farbe ist ausreichend kräftig.
- leicht grünliche Töne werdenebenfalls berücksichtigt
- **Basiswert (0.4)** → Regel grundsätzlich
- **Zusatzwert** → je höher die Sättigung, desto höhere Sicherheit
- **Begrenzung auf max. 1.0** (also = 100 %)
- Rückgabe des Flaschentypen **„Glasflasche_Gruen“**
- Confidence zeigt, **wie sicher** die Entscheidung ist
- Ausgabe im Log als Prozentwert (nur Darstellung)

```
# 1. Grünes Glas
# Hue 28-90° (Grünbereich), mittlere bis hohe Sättigung
if 28 <= hue <= 90 and saturation > 20:  # Erweitert für knapp-grüne Farben
    confidence = min(1.0, (saturation / 100) * 0.6 + 0.4)  # Höhere Base-Konfidenz
    logger.info(f"→ Klassifiziert als: Glasflasche_Gruen (Konfidenz: {confidence:.2%})")
    return 'Glasflasche_Gruen', confidence
```

Hier am Beispiel von Grünglas. Die Werte können variieren und wurden durch das Testen angepasst.
Grün wird überwiegend zuverlässig erkannt.
-HSV ist weniger anfällig für wechselnde Lichtverhältnisse;

Confidence Wert

- **Confidence = Basiswert + gewichtete Farbmerkmale**

```
Confidence = (saturation / 100) * 0.6 + 0.4
```

- Confidence wird **regelbasiert pro Glasart** berechnet
- Jede Regel hat:
- einen **Basiswert**
- eine **gewichtete Bewertung von Farbmerkmalen**
- Wertebereich: **0.0 – 1.0**
- Zu niedrige Confidence- Ergebnis wird verworfen bzw. Fehlermeldung

Wurde hinzugefügt, um Testergebnisse möglichst zuverlässig zu generieren und Fehlerquote besser auszuschließen.
Ebenfalls ware der Cinfidencewert beim Testen sehr wichtig und hat bei der Anpassung von Fehlermeldungen geholfen.
Confidence ist zusammengesetzt und je nach Glasfarbe anders berechnet.
0.0-1.0 liest sich in %. Also von 0 bis 100%.
Relevanter Bestandteil für unser Projekt, da Zuverlässigkeit vorausgesetzt werden muss.

Grobe Übersicht zum derzeitigen System.
Viele Punkte bereits implementiert. Einige müssen noch optimiert werden.
Weitere Schritte werden derzeit geplant.

Bezug PoC

- **PoC 1 – Kamera, Bildaufnahme & ROI**
- Bild-Upload über API
- Standard-ROI + optionaler Custom-ROI
- Performante und deterministische ROI-Berechnung

- **PoC 2 – Regelbasierte Farbanalyse (HSV)**
- Umwandlung BGR → HSV
- Mittelwertbildung im ROI
- Feste Schwellenwerte für Weiß-, Braun- und Grünglas

Bezug PoC

- **PoC 3 – Ergebnisausgabe & Barrierefreiheit**
- Klare textuelle Ergebnisse
- Confidence-Wert zur Transparenz
- Unsichere Ergebnisse werden verworfen

- **PoC 4 – Regionale Entsorgung (Gummersbach)**
- Kontextbezogene Entsorgungsempfehlungen