# Projekt-Log

## zum Praxisseminar im Sommersemester 2023 bei Prof. Dr. Wolff

## von Stefan Röhr, Thomas Hofbauer

### Projektziel:

* Erarbeitung, welche Tools eine vollautomatischer Supermarkt Scanner haben soll.
* Herausfinden welche Probleme ein Supermarkt Scanner verursacht
* Konzeption einer Anwendung für das Projekt AGS des MIT GSL
* Der Fokus des praktischen Projekts war die Bilderkennung des Systems.

### Hintergrund:

* MIT GSL als Anstoß zur Umsetzung des Praxisseminars
* Wunsch nach vereinfachung des Ablaufs an der Supermarktkasse
* Probleme der aktuellen Supermarkt Scanner:
  + Zeitaufwändiger Prozess
  + Self Checkout ist ein fehleranfälliger Prozess
  + Self Checkout ist nicht intuitiv zu bedienen
  + Unternehmen verlieren Geld, weil Kunden abwandern
* Ziel des neuen Automatischen Supermarkt Scanner:
  + Soll das Scannen an der Kasse vereinfachen
  + Verringerung der Fehler beim einscannen
  + Zeiteinsparung für Käufer und Supermarktangestellte
* Aufteilung in drei Nutzergruppen:
  + Supermarktbetreiber
  + Kunden
  + Supermarktangestellte
* Probleme bei der Konzeption des GSL:
  + AGS muss Vorteile für alle Beteiligten bieten
  + Augenscheinlicher Mehraufwand für den Supermarkt und Kunden
  + Negative Einstellung wegen vorhandenen Self Checkout Systemen
* Lösung:
  + Kunden befragen
  + Auf geäußerte probleme ansprechen und Lösungsvorschläge einholen

### Projekt-Ablauf:

Im Rahmen des Projekts wurden wöchentliche und tägliche Gruppentreffen abgehalten, um den Fortschritt und die Entwicklung des Prototyps zu überwachen. Dieser strukturierte Ansatz ermöglichte es dem Team, kontinuierlich Informationen auszutauschen und gemeinsame Entscheidungen zu treffen.

Darüber hinaus wurden gezielte Treffen mit Stakeholdern organisiert, um wertvolle Hintergrundinformationen zu sammeln, die für die Anpassung des Prototyps von großer Bedeutung waren. Diese Interaktionen mit Stakeholdern dienten dazu, ein tiefes Verständnis für die Anforderungen und Bedürfnisse des Endbenutzers zu entwickeln.

Verwendete Hilfsmittel:

* + Discord(Meetings)
  + WhatsApp(Kurznachrichten)
  + Google Drive für kollaboratives Arbeiten

Projektplanung: In diesem Schritt wird die gesamte Projektstruktur entworfen, die Ziele und Meilensteine festgelegt sowie die benötigten Ressourcen und Zeitpläne definiert.

Auswahl der Künstlichen Intelligenz (KI): Hier erfolgt die sorgfältige Auswahl und Evaluierung der KI-Technologie oder des Algorithmus, der im Projekt angewendet werden soll. Dies umfasst die Recherche und Analyse von verfügbaren KI-Optionen.

Produktauswahl: Die Auswahl der Produkte, auf die die KI-Anwendung abzielt, ist ein zentraler Schritt. Dies erfordert eine sorgfältige Analyse der Zielmärkte und Bedürfnisse der Benutzer.

Datenbeschaffung: In diesem Stadium geht es darum, die benötigten Datenquellen zu identifizieren und die Daten zu sammeln, die für das Training und die Validierung des KI-Modells erforderlich sind.

Datenlabelung: Die erfassten Daten müssen gelabelt werden, um die richtige Zuordnung von Eingangsdaten zu Ausgaben zu ermöglichen. Dies ist ein entscheidender Schritt für das Überwachungslernen (Supervised Learning).

KI trainieren: In diesem Schritt wurde die KI auf den gelabelten Daten trainiert.

Programmentwicklung**:** Nachdem eine vorläufige Version des trainierten Modells bereit war, wurde ein das Programm geschrieben, das den Input des Nutzers und die Bilderkennungsergebnisse verarbeitet, um zu entscheiden, ob das Produkt richtig erkannt wurde.

### Programme sind für Linux ausgelegt:

Der Computer, auf dem die Programme für das Projekt erstellt wurden, sowie die KI trainiert wurde, war ein Linux System. Daher sind die Programme auf Linux ausgerichtet. Dazu kommt noch, dass wir empfanden, dass YOLOv8 einfacher für Linux zu installieren ist. Das liegt nicht an YOLOv8 selbst, sondern die dazugehörigen Dependencies, wie dem Python Toolkit, PIP, und einer Shell Umgebung. Da sich Linux und Windows Geräte in ihrer Pfadstruktur unterscheiden, funktioniert das Programm nur für Linux.

### Bedienungsanleitung für das Programm:

Die Bedienungsanleitung, wie man das Programm startet und benutzt, befindet sich in der README.md Datei.

### Entscheidungen Begründen:

* Warum Yolov8: In unserer Recherche haben wir herausgefunden, dass bereits viele Projekte mit YOLO gemacht wurden. Es verfügt auch über eine gute primäre und sekundäre Dokumentation und ist einfach zu installieren(Wenn die Requirements installiert sind, mit einem Befehl in der Linux Command-Line) die trainierten Modelle sind genau und können über das ONNX Format zu anderen Frameworks, wie Tensorflow exportiert werden. Dass YOLO auf Pytorch und somit Python basiert, hat den Vorteil, dass es einfach ist in Python zu integrieren.
* Produktauswahl: Damit wir trotz der limitierten Anzahl an Produkten, die praktischen Möglichkeiten der Bilderkennung austesten, haben wir darauf geachtet, dass wir eine weite Spanne an Produkten abdecken, aber dass die Produkte immer auch in zueinander ähnlichen Paaren auftreten, wie z.B. Rote und Grüne Äpfel oder die Tomaten-Dosen und Wok-Dosen.
* Warum die Daten nicht augmentieren?: Es gab die Überlegung, mehr Trainingsdaten zu erzeugen, die existierenden Trainingsdaten zu augmentieren, indem die Bilder gedreht und gespiegelt werden. Es hat sich herausgestellt, dass dieser Schritt unnötig ist, da YOLOv8 dies bereits automatisch macht.
* Modellgröße: YOLOv8 verfügt über unterschiedliche Modellgrößen, n s m l x. N ist die kleinste, schnellste und am wenigsten genaueste, während X langsamer, genauer und größer ist. Das trainierte Modell, das bei der Abschlusspräsentation vorgelegt wurde, war von der Größe N, doch für die finale Abgabe wurde eines mit der Größe M trainiert. Dadurch befand sich die Trainingszeit nicht mehr bei ca. 7 Stunden, sondern ca. 30 Stunden.
* Warum nur Bilderkennung gemacht wird, wenn kein Barcode angegeben wurde?: Während der Abschlusspräsentation gab es den berechtigten Einwand, dass es schwer wäre, die Bilderkennung für die tausenden Produkte in einem Laden zu trainieren und dass dies gar nicht notwendig ist. Stattdessen wäre die Bilderkennung nur für jene Produkte notwendig, die keinen Barcode haben, wie Früchte. Daher wurde das Programm angepasst. Wird ein Barcode angegeben, wird keine Bilderkennung durchgeführt und nur geschaut, ob das angegebene Gewicht zum angegebenen Barcode passt. Wird aber kein Barcode angegeben, wird aus zwei Blickwinkeln eine Bilderkennung durchgeführt und basierend darauf und dem angegebenen Gewicht entschieden, ob das Produkt erkannt wurde.

Leider hatten wir nicht genügend Zeit noch einmal von 8 neue Früchte jeweils 800 Fotos zu machen, diese zu labeln und darauf basierend die KI noch einmal zu trainieren. Daher tut das Programm, wenn kein Barcode angegeben wird, einfach so, als handele es sich bei all den Fotos der Produkte, z.B. der Vollmilch, um Früchte, die keine Barcodes haben.

* Das Programm-Interface: Das Interface funktioniert über die Command-Line, da es in dem theoretischen Endgerät ganz wegfallen würde. Die Daten, wie Gewicht, Barcode ID, oder die Quellen für die beiden Bilder müssten dann nicht mehr per Hand eingegeben werden, sondern würden von dem Gerät automatisch übermittelt. Im momentanen Interface können Quellen für die Bilder über einen direkten Datenpfad angegeben oder einen relativen Dateipfad angegeben werden. Befindet sich ein Bild im “pictures” Ordner des Programms und es handelt sich um ein jpg, kann auch nur deren Name geschrieben werden. Handelt es sich nicht um ein jpg, muss stattdessen deren Name und Datentyp angegeben werden( Achtung, JPG wird als anderer Datentyp, als jpg behandelt). Es ist zu beachten, dass, da das Programm auf Linux programmiert wurde, es nur auf Linux Systemen funktioniert.
* Wie das Programm die Bilderkennung-Ergebnisse bewertet: Nachdem das Programm die Bilderkennung der beiden Bilder durchgeführt hat, führt es noch einige weitere Schritte durch. Die Zuversicht Schwelle wurde auf 60% gesetzt, was bedeutet, dass das Ergebnis als unzulänglich zurückgewiesen wird, falls keines der beiden Ergebnisse mehr als 0,6 Zuversicht hat. Sind sich die Ergebnisse nicht einig, wird jenes mit höherer Zuversicht genommen, wobei eine Übereinstimmung des Ergebnisses mit dem angegebenen Gewicht einen Zuversicht-Bonus von 0.2 gibt.

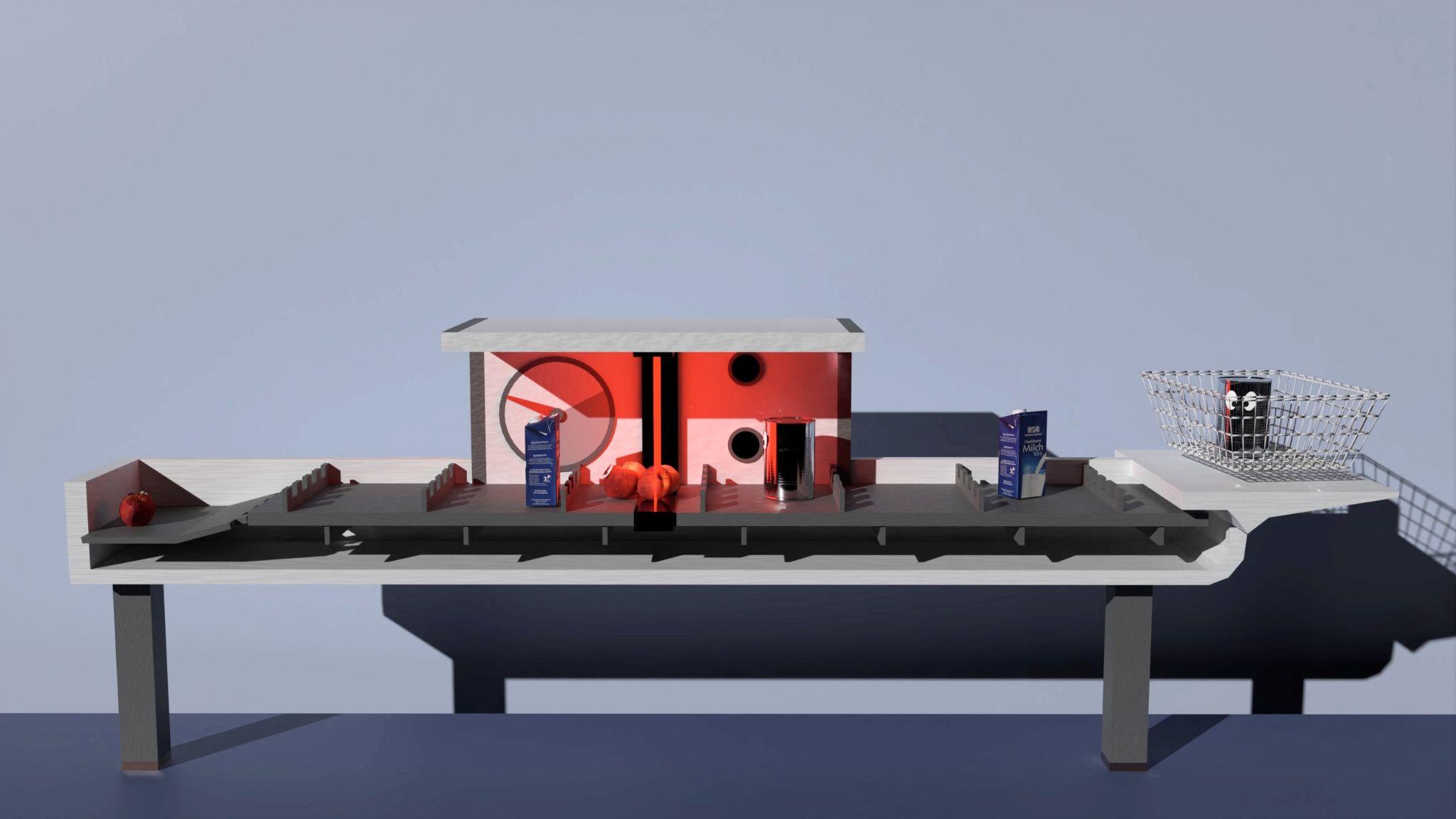
Vorteil durch den Entscheidungsprozess: Dass dem System für ein Objekt mehr als nur ein Bild für die Bilderkennung zur Verfügung steht, hat mehrere Vorteile. Der erste ist, dass es Redundanz ins System bringt. Es kann vorkommen, dass die Bilderkennung sich bei einem Bild unsicher ist, oder sich irrt, aber es ist weniger wahrscheinlich, dass es gleich zweimal passiert. Falsche Identifizierungen tendieren zu passieren, wenn das System sich unsicher ist. Das bedeutet, dass das richtige Ergebnis mit hoher Wahrscheinlichkeit eine höhere Zuversicht hat und daher gewählt wird. Zusätzlich hat das richtige Ergebnis noch den Vorteil, da das Gewicht von dem real gemessenen Objekt mit denen im Bild erkannten Objekten verglichen wird.

Dass die beiden Bilder radikal unterschiedliche Blickwinkel haben, vermeiden, dass nur ein schwer zu identifizierender Teil des Objektes zu sehen ist, wie z.B. nur der Deckel einer Dose.

Zugegebener weise würde es die Redundanz und das Kompensieren für schlecht erkennbare Blickwinkel weiter steigern, wenn das System noch mehr Bilder mit variierenden Blickwinkeln verwenden würde, aber wir glauben, dass die zwei Blickwinkel weitestgehend ausreichen sollten, da sie sich zu 90Grad unterscheiden. Außerdem würden zusätzliche Kameras das Gerät teurer machen.

### Prototypen:

* Prototyping:
  + Allgemeine Grundsätze:
    - Die Prototypen sind in ihrem Konzept als einheitliches System einer Supermarktkasse gedacht, anstelle drei unzusammenhängender Systeme.



* Virtueller Prototyp:

Beschreibung des Prototypen:

Der virtuelle Prototyp, den wir entwickeln, ist ein innovatives System, das dazu dient, den Einkaufsprozess in Supermärkten zu revolutionieren. Dieser Prototyp verfügt über eine integrierte Bilderkennung, einen Barcode-Scanner und eine Waage, um Produkte in einem Supermarkt automatisch zu identifizieren, zu verifizieren und zu wiegen. Der Hauptzweck dieses Prototyps besteht darin, den Einkauf für Kunden zeitsparender, bequemer und effizienter zu gestalten.

Funktionalitäten:

-Bilderkennungssystem: Das Bilderkennungssystem ermöglicht die Identifizierung von Produkten anhand visueller Merkmale.

-Barcode-Scanner: Der Barcode-Scanner bietet eine zusätzliche Möglichkeit zur Produktidentifikation

-Waage: Die integrierte Waage ermöglicht es dem Prototypen, das Gewicht von Produkten zu kontrollieren.

* Physischer Prototyp:

Beschreibung des Prototypen: Unser physischer Prototyp repräsentiert einen entscheidenden Meilenstein auf dem Weg zur Entwicklung eines vollautomatischen Supermarkt Scanners. Dieses innovative System ist mit zwei hochauflösenden Kameras ausgestattet, die es ermöglichen, Produkte aus unterschiedlichen Perspektiven zu erfassen und detaillierte Bilder zu generieren. Diese Bilder werden anschließend verwendet, um das Bilderkennungssystem zu trainieren.

Funktionalitäten:

-Doppelte Kameraperspektiven: Der Prototyp verfügt über zwei Kameras, die Produkte aus verschiedenen Winkeln und Perspektiven aufnehmen. Dies ermöglicht die Erfassung von Produkten aus der Seitenansicht und der Draufsicht, um eine bessere Erkennung und Identifikation sicherzustellen.

-Bilderkennung und Trainingsdaten: Die aufgenommenen Bilder dienen als Grundlage für das Bilderkennungssystem. Das System erfasst und analysiert die visuellen Merkmale der Produkte, um sie später im Supermarkt automatisch zu identifizieren.

* Problemstellung und Ziel: Die Entwicklung dieses physischen Prototyps ist eine Reaktion auf die Herausforderungen, die mit dem Einkaufsprozess in herkömmlichen Supermärkten verbunden sind. Kunden verbringen oft viel Zeit damit, Produkte manuell zu identifizieren und zu scannen, was zu langen Warteschlangen und Unannehmlichkeiten führt.

Unser Ziel ist es, diese Probleme zu lösen, indem wir ein System entwickeln, das die Identifikation und Erfassung von Produkten automatisiert. Die Verwendung von zwei Kameras ermöglicht eine präzise Produktidentifikation und -erfassung, unabhängig von Größe und Form.

Dieser Prototyp dient nicht nur als funktionierendes Modell, sondern auch als entscheidende Datengrundlage, um das Bilderkennungssystem zu trainieren und zu optimieren. Unser langfristiges Ziel ist es, ein vollautomatisches Einkaufssystem zu schaffen, das Kunden ein bequemeres und effizienteres Einkaufserlebnis bietet.

Die Erstellung von Trainingsdaten mithilfe physischer Prototypen ist ein wichtiger Schritt auf dem Weg zur Entwicklung eines zuverlässigen und genauen Bilderkennungssystems für den Supermarktscanner der Zukunft.

### KI trainieren:

* Die KI wurde auf insgesamt ca. 8000 Fotos trainiert. Diese wurden mit den Labelimg-Master und YoloLabel ge labeled. Dadurch wurde für jedes Bild eine gleichnamige txt Datei erstellt, die fünf nummern enthielt, die erste war die Klasse und die letzten vier waren die Position und Dimensionen der Bounding-Box um das Objekt. Es gab insgesamt zehn Klassen(AppleR; AppleG; DoseTomate; DoseWok; Vollmilch; Hafermilch; CornflakesJa; CornflakesDr; NudelnBarilla; NudelnBernbacher). Nachdem alle Fotos gelabeled wurden, mussten die Daten in mehrere Sets geteilt werden. Dazu wurde split-folders 0.5.1 benutzt(<https://pypi.org/project/split-folders/>). Die Daten wurden dadurch zufällig in ein großes Trainings-Set und zwei kleine Test- und Validation-Sets aufgeteilt. Eine yaml datei wurde erstellt, die Yolo beim Trainieren sagt, wo die Daten-Sets sind, sowie wie viele Klassen es gibt und welche diese sind. Die KI wurde dann auf einem Linux Mint Rechner mit einer GeForce RTX 3060 trainiert. Die Learningrate(Wie schnell die KI lernen soll) war auf 0.01 gesetzt und die Epochen zahl auf 300 wobei Yolo automatisch stoppt, wenn sich das Modell über die letzten 50 Epochen nicht verbessert hat. Bei der Modellgröße wurde, wie bereits erwähnt, die mittlere Größe M benutzt.

### Erkenntnisse:

* Validierung der Bilderkennung: Am Ende des Trainings eines YOLOv8 Modells wird automatisch eine Validierung des neu trainierten Modells generiert. Die vollständige Validierung des trainierten Modells haben wir in dem ImageDetectionValidationData-Ordner beigefügt.
  + Der Graph metrics/Precision zeigt an, dass die Präzision bei ca. 82% liegt und der Metrics/recall zeigt an, dass der Recall bei ca. 99,8% liegt.
  + Die mAP50 (mean Average Precision with the Threshold 50) ist ungefähr 84% und die mAP50-95 ist ungefähr 68%. Der “Threshold” ist zu wie viel Prozent die Begrenzung, die die KI zieht, wo sie denkt, dass das Objekt in einem Bild ist, mit der Box übereinstimmt, die beim Labeln der Daten gezeichnet wurde.
  + Wie man anhand der Loss Grafen sehen kann, hat die Loss Rate während dem Training stark abgenommen. Die Rate ist nicht in Prozent und ein niedriger Wert ist besser. Die verschiedenen Loss werte sind. in box\_loss( Wie treffsicher die KI ist die Koordinaten des Objekts im Bild zu finden.), cls\_los( Die Wahrscheinlichkeit, zu der die KI das Objekt richtig klassifiziert. ), dfl\_loss (Hilft der KI sich auf schwer oder seltene Objekte zu konzentrieren)
  + Die Precision-Confidence Kurve zeigt, dass die Bilderkennung für die meisten Produkte, selbst bei niedriger Zuversicht, eine gute Präzision aufweist. Ausnahmen sind die Nudeln von Barilla und die Nudeln von Bernbacher. Das deutet an, dass diese beiden Produkte noch mehr Bilddaten benötigen würden, auch wenn, wie in der folgenden Sektion gezeigt wird, es bei unserem Programm zu keinen Problemen geführt hat.
* Validierung des gesamten Programms:  
   Zur Validierung unseres Programms haben wir es 100-mal per Hand getestet, zehnmal für jedes Produkt. Die Bilder wurden dazu aus dem Test-Set genommen, da dieses nicht im Trainieren des Bilderkennungs-Modells benutzt wird. Für jeden Test wurden Bilder genommen, die tatsächlich dasselbe Produkt zeigten, und die richtige Ansicht hatten(Bilder mit Draufsicht werden für die Bilderkennung der Draufsicht benutzt). Ansonsten wurden die Bilder einfach in der Reihenfolge genommen, in der sie in dem Ordner auftauchen. Die Gewichtsangaben wurden ebenfalls im Vorhinein festgelegt, wobei die Hälfte der Gewichtsangaben für das jeweilige Produkt richtig waren und die Hälfte falsch. Die Reihenfolge von richtigen und falschen Gewichtsangaben war auch für alle Produkte gleich.  
    
  Bei den einhundert Testdurchläufen hat sich das Programm zu 100% richtig entschieden. Es hat das richtige Produkt erkannt und falls das Gewicht falsch war, es auch gemerkt. Da jeder Testdurchlauf zwei Bilder hat, wurde 200-mal eine Bilderkennung durchgeführt. Von diesen hatten 96% eine Zuversicht über 80%. Zehn hatten eine Zuversicht, zwischen 70%-80% und zwei hatten eine Zuversicht zwischen 60%-70%. Doch in keinem der Fälle hatte das Programm zwei Bilderkennung Ergebnisse mit weniger als 80% Zuversicht, und die Bilderkennung lag auch nie falsch. Alle Ergebnisse mit weniger als 80% Zuversicht ereigneten sich bei der Draufsicht. Daher würden wir raten, dass, falls das Projekt weitergeführt wird, die Anzahl der Bilddaten mit Draufsicht auf das Produkt erhöht werden sollte.

### Probleme:

* Label-Klassen vom Programm verschoben: Eine der beiden Labeling-Software hat in den Labeldateien die Klassen der Produkte falsch angegeben. Eigentlich sollten diese Nummern von 0 bis 9 sein, aber das Programm hat Klassen von 6 bis teilweise 26 angegeben. Zum Glück hatten dennoch alle Bilder innerhalb eines Produktes die gleiche Klasse, weshalb ein einfaches Programm geschrieben werden konnte, das alle Klassen in einem Ordner änderte.  
   **!!!Nicht in einem Ordner mit anderen TXT Dateien anwenden!!!**, da jeglicher Text bis zum ersten Leerzeichen in allen TXT Dateien in dem Ordner durch die neue Klassennummer ersetzt wird.
* Groß, Kleinschreibung der Dateitypen ist ein Problem: Wenn nur der Name einer Bilddatei in dem “pictures” Ordner genannt wird, erwartet das Programm eine jpg Datei. Eine JPG-Datei wird aber nicht erkannt. Daher wurde ein Programm geschrieben, das alle JPG Dateien in einem Ordner zu jpg ändert.
* Daten in Sets verteilen: Die Daten in die Trainings-, Validation- und Test-Sets, aufzuteilen stellte sich als schwierig heraus, da sich mehrere Online Anleitungen dazu uneinig waren, welche Ordnerstruktur die Daten haben sollten, bevor sie aufgeteilt werden und erst festgestellt werden konnte, ob die Aufteilung richtig war, wenn der KI-Trainingsprozess gestartet wurde.

Verlust des Rechners: In der Nacht vom 30.10 auf den 31.10 um ca. 2 Uhr trat auf dem Linux Rechner, mit dem die Programme für das Projekt entwickelt und die KI trainiert wurden, ein kritischer Fehler auf. Die Festplatte wurde voraussichtlich unwiderruflich korrumpiert. Auslesen der Bytes, dank Herr Wimmer, zeigten mehrere kritische read-write errors an mehreren Stellen der Festplatte auf. Daher musste der Rechner mit einer neuen Festplatte neu aufgesetzt werden und die YOLOv8 neu installiert werden, damit das Training des Yolo Modell mit der Größe M erneut gestartet werden konnte und die Validierung durchgeführt werden konnte. Bei der Installation von YOLOv8 ergab sich wieder ein Fehler, der dafür sorgte, dass nur die CPU für die KI benutzt werden konnte, was zu langsam fürs Training gewesen wäre (voraussichtliche Trainingszeiten von 10 Tagen). Erst wurden die Programme und deren Requirements mehrfach neu installiert, was nichts half. Das System noch ein drittes Mal aufzusetzen, löste das Problem allerdings.

### Ausblick:

Der Ausblick für dieses Projekt ist äußerst vielversprechend, da es das Potenzial hat, den Einkaufsprozess und das Einkaufserlebnis im Supermarkt grundlegend zu transformieren.

Die nächsten Schritte umfassen die nahtlose Integration von Hardware Komponenten, wie Kameras, Sensoren und Aktoren, um die physische Erfassung und Identifizierung von Produkten zu ermöglichen. Darüber hinaus werden Fortschritte in der Echtzeit-Bildverarbeitung erwartet, um die Genauigkeit bei der Produktidentifikation weiter zu erhöhen. Ein weiterer wichtiger Schritt ist die effiziente Erfassung und Verwaltung von Daten, einschließlich Produktdaten, Preisen und Gewichten. Dies erfordert die Implementierung einer stabilen Datenbank und eines zuverlässigen Back-End-Systems. Sicherheitsaspekte, wie der Schutz der Kundendaten und die Prävention von Betrug, müssen im weiteren Verlauf des Projekts berücksichtigt werden.

Die Skalierbarkeit des Systems auf mehrere Filialen oder Supermärkte, die kontinuierliche Wartung und Aktualisierung der Software und Hardware sowie zusätzliche Funktionen und Erweiterungen sind weitere Schritte auf dem Weg zur erfolgreichen Umsetzung des Projekts. Schließlich müssen Partnerschaften mit Supermärkten oder Einzelhandelsketten in Betracht gezogen werden, um das System in der Praxis einzuführen und ein geeignetes Geschäftsmodell zu entwickeln. In der Testphase werden umfassende Tests durchgeführt, und das Feedback von Testnutzern wird genutzt, um das System weiter zu verbessern. Die Markteinführung erfordert eine sorgfältige Planung und Marketing Bemühungen, um Interesse bei Supermärkten und Endverbrauchern zu wecken. Insgesamt hat dieses Projekt das Potenzial, die Art und Weise, wie wir einkaufen, erheblich zu verändern und ein bequemeres und effizienteres Einkaufserlebnis zu bieten. Flexibilität und Anpassungsfähigkeit sind in dieser aufregenden Reise von entscheidender Bedeutung, um den sich wandelnden Bedürfnissen und Anforderungen des Marktes gerecht zu werden.

Links:

Link zu GSL Präsentation:

[Pitch - Google Präsentationen](https://docs.google.com/presentation/d/1eAiHUsKTzvMojEh329_8xAUiRYFycQQwQFrfJ2twUhM/edit#slide=id.g25d7303f645_2_0)

Link zu Praxisseminar Abschlusspräsentation:

[Abschlusspräsentation\_praxisseminar - Google Präsentationen](https://docs.google.com/presentation/d/1VwGy3_atxeGsNy6e9apjmEKKYayqKBGuXvR66OmxFHM/edit#slide=id.p14)