C:\Users\haller\Desktop\Logo_HFU.tif

Projektdokumentation

in

AIN / ITP

Convolutional Neural Networks

Qualitätsanalyse und Vorhersage von Komponenten des Weinschorle-Automaten

|  |  |
| --- | --- |
| Referent: | Prof. Dr. Christoph Reich |
| Korreferent: | Mathias Lermer |
| Vorgelegt am: | [Datum] |
| Vorgelegt von: | Arulsothy, Sugirthan  Hirt, Marius  Mattes, Oliver  Türkmen, Yasemin |
|  |  |
|  |  |

# Vorwort

# Abstract

[Englisch, 100 -120 Worte]

[Deutsch, 100 – 120 Worte]

# Inhaltsverzeichnis

[Vorwort I](#_Toc514078746)

[Abstract III](#_Toc514078747)

[Inhaltsverzeichnis V](#_Toc514078748)

[Abbildungsverzeichnis VII](#_Toc514078749)

[Tabellenverzeichnis IX](#_Toc514078750)

[Abkürzungsverzeichnis XI](#_Toc514078751)

[1 Einleitung 1](#_Toc514078752)

[2 Grundlagen 3](#_Toc514078753)

[2.1 Test 3](#_Toc514078754)

[2.2 Test 2 3](#_Toc514078755)

[2.2.1 Test 3 3](#_Toc514078756)

[3 Gläser 5](#_Toc514078757)

[3.1 Anforderungen und Ziele 5](#_Toc514078758)

[3.2 Ausgangssituation 5](#_Toc514078759)

[3.3 Vorgehen 6](#_Toc514078760)

[3.3.1 Trainingsdaten Sammeln 6](#_Toc514078761)

[3.3.2 Trainingsdaten und Script anpassen 6](#_Toc514078762)

[4 Volumenberechnung mit der Kinect 9](#_Toc514078763)

[4.1 Anforderungen und Ziele 9](#_Toc514078764)

[4.2 Vorgehen 9](#_Toc514078765)

[4.2.1 Kapazitäten der Kinect 9](#_Toc514078766)

[4.2.2 Vogelperspektive 9](#_Toc514078767)

[4.2.3 Frontalperspektive 11](#_Toc514078768)

[4.2.4 Fazit zur Kinect 12](#_Toc514078769)

[5 Verschmutzung des Drehtellers 13](#_Toc514078770)

[6 Fazit 15](#_Toc514078771)

[Literaturverzeichnis 17](#_Toc514078772)

[Stichwortverzeichnis 19](#_Toc514078773)

[Eidesstattliche Erklärung 21](#_Toc514078774)

[A. [Anhang] 23](#_Toc514078775)

# Abbildungsverzeichnis

[Abbildung 1: Position der Kamera 5](file:///C:\Users\Oliver\Documents\Olis%20Shit\Forschung%20und%20Lernen\cnn\cnnproject\documentation\Projektdokumentation.docx#_Toc514078776)

[Abbildung 2: Ursprünglich bekannte Gläser 5](file:///C:\Users\Oliver\Documents\Olis%20Shit\Forschung%20und%20Lernen\cnn\cnnproject\documentation\Projektdokumentation.docx#_Toc514078777)

[Abbildung 3: Neue Gläser 6](file:///C:\Users\Oliver\Documents\Olis%20Shit\Forschung%20und%20Lernen\cnn\cnnproject\documentation\Projektdokumentation.docx#_Toc514078778)

[Abbildung 4: (a) Foto vor und (b) Foto nach dem Zuschneiden 7](file:///C:\Users\Oliver\Documents\Olis%20Shit\Forschung%20und%20Lernen\cnn\cnnproject\documentation\Projektdokumentation.docx#_Toc514078779)

[Abbildung 5: Platzierung der Kinect für Vogelperspektive 10](file:///C:\Users\Oliver\Documents\Olis%20Shit\Forschung%20und%20Lernen\cnn\cnnproject\documentation\Projektdokumentation.docx#_Toc514078780)

[Abbildung 6:Kinect Aufnahme aus Vogelperspektive (a) RGB-Modus (b) Tiefen-Modus 11](file:///C:\Users\Oliver\Documents\Olis%20Shit\Forschung%20und%20Lernen\cnn\cnnproject\documentation\Projektdokumentation.docx#_Toc514078781)

[Abbildung 7: Platzierung der Kinect für Frontalperspektive 12](file:///C:\Users\Oliver\Documents\Olis%20Shit\Forschung%20und%20Lernen\cnn\cnnproject\documentation\Projektdokumentation.docx#_Toc514078782)

[Abbildung 8: Perspektive der Kinect bei Frontalpositionierung 12](file:///C:\Users\Oliver\Documents\Olis%20Shit\Forschung%20und%20Lernen\cnn\cnnproject\documentation\Projektdokumentation.docx#_Toc514078783)

# Tabellenverzeichnis

[Tabelle 1: Umsetzung der Industrie 4.0 Konzepte im WISH 4.0 2](#_Toc513471880)

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| CNN | Convolutional Neural Network |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Einleitung

Im Rahmen des Wettbewerbs "CLOUD-MEISTER 2017: DER WAGO-IDEENWETTBEWERB" hat ein Team von Studierenden um Herrn Reich den Weinschorle-Automaten „WISH 4.0“ entwickelt. Am Smartphone, Tablet oder Computer lässt sich dabei das gewünschte Mischverhältnis des Schorles per Webseite angeben, das eingestellte Glas wird per Webcam bemessen, das Volumen bestimmt und anschließend über mehrere Ventile das Schorle gemischt.

Außerdem kann der Automat auch Barcodes erkennen, um früher festgelegte Mischungen wieder zu erkennen. Im Zentrum der Umsetzung standen dabei die in Tabelle 1 dargestellten Industrie 4.0 Konzepte.

Tabelle : Umsetzung der Industrie 4.0 Konzepte im WISH 4.0

|  |  |
| --- | --- |
| Industrie 4.0 Konzept | WISH 4.0 Umsetzung |
| Individualisierte Produktion | Individuelles Glas, Mischverhältnis, Füllmenge und Glas-Label |
| Prozessoptimierung in der Cloud | Nutzungsabhängige Auswahl des Befüllers |
| Zustandsüberwachung (Condition Monitoring) in der Cloud | Parameter der Befüller wird überwacht |
| Optimierte Wartung | Je nach Nutzung wird der Bedarf einer Wartung bestimmt |
| Cloud Steuerung | Meta-Steuerbefehle ( z.B. fülle Glas) werden von der Cloud an die Lokale Steuerung geschickt  Lokale Steuerung (WAGO-Controller) führt die Geräte-Befehle aus |
| Visual Quality Management | Visuelle Erkennung von verschiedenen Gläsern in der Cloud |
| Smart Logistics | Erkennung des Barcodes, Kupplung zu Glas, und speicherung für weitere Verarbeitung |
| Supply Chain Compliance | Einhaltung vordefinierter Compliance Vorgaben bezüglich des zu druckenden Labels |

# Grundlagen

## Test

## Test 2

### Test 3

# Gläser

## Anforderungen und Ziele

Der Weinschorle-Automat soll eine Vielfalt verschiedener Glasformen erkennen um vor dem Befüllen dem Glas ein vordefiniertes Volumen zuordnen können. Ziel dieser Aufgabe ist das Vertraut machen mit Python an sich und insbesondere mit dem Skript, welches bereits in der Cloud für die Zuordnung zuständig ist.

## Ausgangssituation

Der Automat kann zu Beginn des Projekts bereits zwischen zwei unterschiedlichen Glastypen unterschieden (siehe Abbildung 2). Dazu nimmt eine vertikal in der Fotokammer angebrachte Kamera (siehe Abbildung 1) ein Bild des eingestellten Glases auf, welches zunächst zur Performance-Verbesserung auf ein kleineres Format zugeschnitten wird. Dieses Bild wird als Eingabe für das CNN (Convolutional Neural Network) übergeben. Als Ausgabe wird das Glas im Automaten mit einer gewissen Wahrscheinlichkeit einem der beiden bekannten Glasformen zugeordnet. Basierend auf diesem Ergebnis wird die befüll Routine durchgeführt.

## Vorgehen

### Trainingsdaten sammeln

Zunächst wurde die Kamera von dem Weinschorleautomat getrennt und direkt an einen Laptop angeschlossen. Über diesen haben wir automatisch eine Serie von 100 Bildern für insgesamt vier neue Glasformen (siehe Abbildung 3) aufgenommen. Zwischen den Aufnahmen war jeweils eine verzögerung von drei Sekunden, so dass wir das Glas zwischen den Aufnahmen drehen konnten. Dadurch haben wir ein diverseres Trainingsdatenset erzeugt, was die Trainingseffektivität des CNN verbessert. (ZITAT)

### Trainingsdaten und Script anpassen

Bevor wir das CNN mit diesen Daten trainieren konnten mussten wir die Bilder zuschneiden (siehe Abbildung 4). Dies geschah mit dem selben Pythonscript, welches bereits in der Routine des Weinschorleautomaten verwendet wird. Desweiteren mussten wir das Trainingsscript dahingehend anpassen, dass wir die neuen Trainingsdaten als solche deklarieren und jedem Trainingsset ein entsprechendes Label zuordnen.

Damit hat sich die Anzahl der klassifizierbaren Gläser auf insgesamt sechs erhöht. Deshalb mussten wir im Predictscript die Dimension des Output-Tensors an die neue Anzahl der Klassen entsprechend anpassen.



# Volumenberechnung mit der Kinect

## Anforderungen und Ziele

Um sich nicht mehr auf die Klassifizierung verschiedener Vordefinierter Gläsertypen verlassen zu müssen, soll der Weinschorle-Automat in der Lage sein das Volumen eines Glases mithilfe des Tiefensensors einer Microsoft Kinect zu berechnen.

## Vorgehen

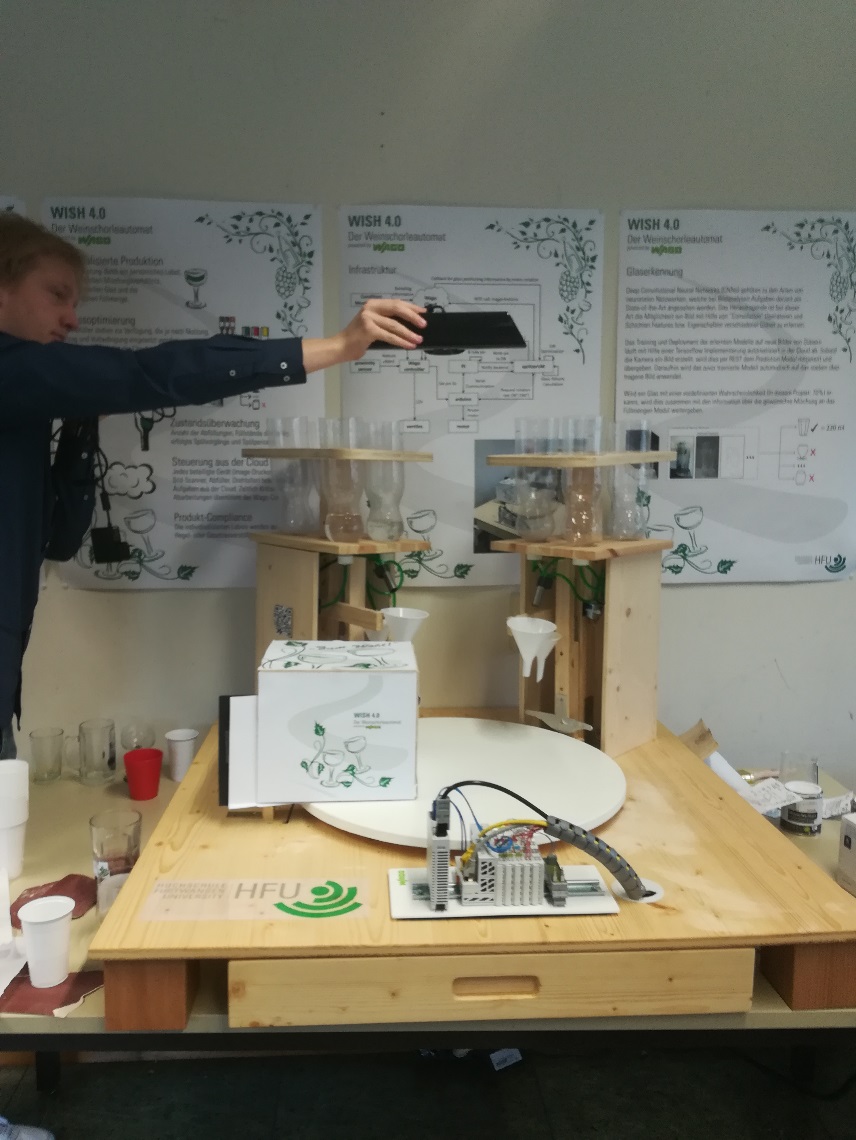
### Kapazitäten der Kinect

Bevor wir die Kinect in den Weinschorle Automaten einbezogen haben, testeten wir deren Möglichkeiten. Nach der Installation der notwendigen Treiber-Software begannen wir mit den Verschiedenen Modi der Kinect zu Experimentieren.   
Nach ersten Tests und Recherchen stellen wir fest, dass der Tiefensensor der Kinect bei Objekten näher als ca. 50cm keine zuverlässigen Werte liefern kann. Diese Tatsache schränkte die Positionierungsmöglichkeiten am Weinschorle-Automaten deutlich ein. Übrig blieben die beiden Möglichkeiten:

1. Vogelperspektive
2. Frontalperspektive

### Vogelperspektive

Unsere erste Idee war es die Gläser von direkt über ihnen aufzunehmen. Dazu müssten wir allerdings einen Überbau konstruieren, um die Kinect mindestens 50cm über den Glasrändern anbringen zu können. (Abbildung 5)

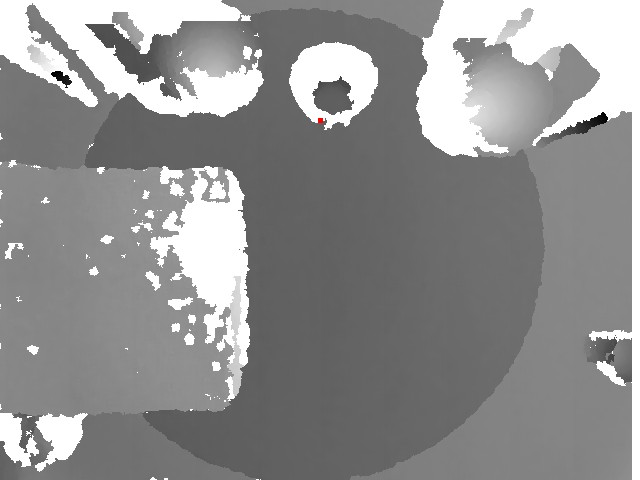


Vorteile dieser Positionierung sind zum einen die einfachere Messung des Volumens, da wir nur zwei Messpunkte im selben Bild benötigen. Einen für die Entfernung des Randes und einen für die Entfernung des Glasbodens zur Kinect. Daraus lässt sich dann einfach die Füllhöhe als Differenz der beiden werte berechnen. Mithilfe des Radius könnten wir das Volumen des Glases abschätzen.

Zum andern könnten wir bei dieser Positionierung die RGB-Kamera der Kinect einsetzen, um die Sauberkeit des Drehtellers zu überprüfen. (siehe Verschmutzung des Drehtellers)

Ein Nachteil wäre der zusätzliche Aufwand einen Überbau zu konstruieren, an dem wir die Kinect befestigen können. Außerdem muss bei diesem Ansatz das Glas zwischen den beiden Trichtern des Weinschorle-Automaten vermessen werden (siehe Abbildung 6a). Dies würde uns dazu zwingen den Drehteller nach der Vermessung gegen den Uhrzeigersinn zu drehen. Da zu diesem Zeitpunkt bereits ein neues Glas auf dem Drehteller stehen könnte, würde dessen Füllroutine behindert.

Zum Scheitern dieses Ansatzes führte schließlich die Untauglichkeit der Kinect für die Messung des Glasrandes, wie man in Abbildung 6b erkennen kann. Im besten Fall sollte man an der Stelle des Glasrandes einen dunkelgrauen Ring erkennen können. Jedoch nimmt die Kinect nur undefinierte Werte auf (repräsentiert durch weiß im Bild) Die Kinect gibt Tiefenwerte von 0 (schwarz) bis 2048(weiß) für jeden Pixel in ihrem Tiefenbild an.



### Frontalperspektive

Alternativ bliebe noch die Möglichkeit die Gläser frontal aufzunehmen, während sie unter dem ersten Trichter auf das Befüllen warten (siehe Abbildung 8). Dazu positionieren wir die Kinect in der vorderen rechten Ecke des Tisches (siehe Abbildung 7).

Aus dieser Positionierung ergeben sich folgende Vorteile:

* Kein Überbau benötigt
* Keine Rotation des Drehtellers gegen den Uhrzeigersinn

Die Untauglichkeit der Kinect zuverlässige Werte zu liefern führt auch hier, ähnlich wie bei der Vogelperspektive, zum Scheitern des Ansatzes.

### C:\Users\Oliver\Downloads\IMG_20180514_153607.jpgFazit zur Kinect

Die Kinect eignet sich nicht für Distanzen unter 50cm Entfernung, was schwer mit den Dimensionen des Weinschorle-Automaten zu vereinbaren ist. (z.B. mit einem Überbau).

Weiterhin reicht die Auflösung der Tiefenkamera nicht aus um schmale Flächen wie einen Glas- oder Becherrand zu erfassen, was zum Scheitern unserer ersten Idee führte.

Zuletzt scheitert diese Idee an der Tatsache, dass die Tiefenkamera der Kinect durch Projektionen von Infrarotlicht funktioniert, wobei aus den Reflektionen der Objekte ein Tiefenbild generiert wird. Durch fehlende Fähigkeit von Glas Licht zu reflektieren, ist es nahezu unmöglich dieses zu vermessen.

Zusammenfassend genügen die Fähigkeiten der Kinect nicht den Anforderungen des Projektes.

# Verschmutzung des Drehtellers

## Anforderungen und Ziele

In Regelmäßigen Abständen soll der Verschmutzungsgrad des Drehtellers von einer Kamera erfasst und ausgewertet werden. Dabei sollen alle anstehenden und laufenden Befüll-Routinen angehalten werden, bis eine entsprechende Säuberung durchgeführt wurde

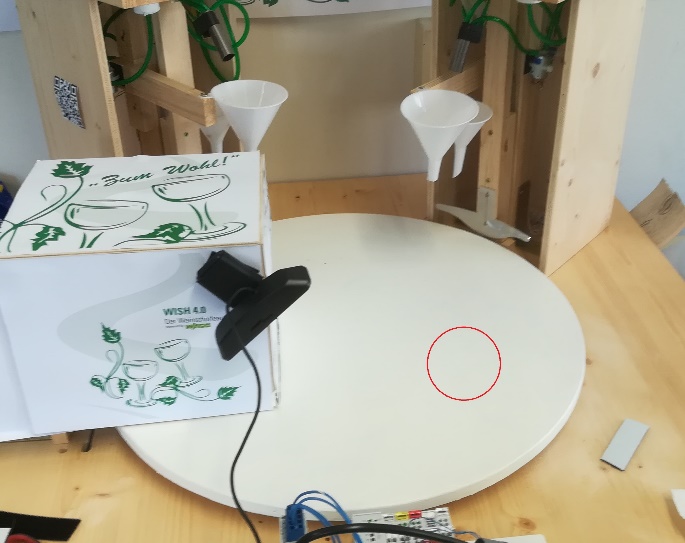
## Vorgehen

Um mögliche Komplikationen zu minimieren führen wir die Schmutzerkennung direkt vor jeder Befüll-Routine durch. Dabei ist die Kamera auf die Endposition der Gläser ausgerichtet.

Grund dafür ist die erhöhte Wahrscheinlichkeit, dass man den Inhalt des Glases beim Entnehmen verschüttet, weshalb die größte Verschmutzungsdichte eben dort zu erwarten ist.

Vorteil davon ist, dass wir ebenfalls im Blick haben, ob das fertig befüllte Glas der vorherigen Routine beim Start einer neuen eventuell noch nicht entnommen wurde. Dies hätte zur Folge, dass ein volles Glas mit der Fotokammer kollidiert, was natürlich unerwünscht ist.

### Positionierung der Kamera

Um den Endbereich der Routine (Abbildung 9, rechts), optimal zu Erfassen bringen wir die Kamera an der Seite der Fotokammer an (Abbildung 9, links). Durch diese Positionierung haben wir alle relevanten Bereiche im Blick, während wir den Hintergrund minimieren.

### Trainingsdaten sammeln

Für diesen Anwendungsfall definieren wir 3 Klassen:

* Sauber
* Verschmutzt
* Besetzt

Dazu haben wir zunächst für 6 unterschiedliche Gläser jeweils 50 Bilder aufgenommen, auf welchen die Gläser an der erwarteten Endposition stehen. Diese klassifizieren wir als „besetzt“ (Abbildung 10c).

Als zweites haben wir insgesamt 50 Fotos des sauberen Drehtellers in verschiedenen Positionen aufgenommen und diese als „sauber“ klassifiziert (Abbildung 10a).

Zum Schluss haben wir mit Weiß-, Rosé-, und Rotwein jeweils 50 Bilder aufgenommen, indem wir ein Glas befüllten und den Inhalt beim Aufnehmen verschütteten, wie es auch in Anwendung sehr wahrscheinlich passieren wird. Dieses Set klassifizieren wir als „verschmutzt“ (Abbildung 10b, Verschmutzung oben zu erkennen).



Abbildung : Auszug aus den zugeschnittenen Trainingsdaten zur Verschmutzungserkennung (a) sauber, (b) verschmutzt und (c) besetzt

# Fazit

# Literaturverzeichnis

# Stichwortverzeichnis

#### **Keine Indexeinträge gefunden.**

# Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbständig verfasst und hierzu keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Stellen der Arbeit die wörtlich oder sinngemäß aus fremden Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt oder an anderer Stelle veröffentlicht.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[Ort, Datum Name]

# [Anhang]