

Optimierung der Wiederholungsgenauigkeit von Toastern durch Einsatz von neuronalen Netzwerken

MASTERARBEIT

Zur Erlangung des akademischen Grades

„Master of Science in Engineering“

Studiengang:

„Mechatronik“

Management Center Innsbruck

Betreuer:

Ing. Ronald Stärz, BSc, MSc

Verfasser:

Kurucz Bálint

1810620039

Eidesstattliche Erklärung

„Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.“

Bad Reichenhall, 31.07.2020

Ort, Datum

Va BzD

Unterschrift

Eidesstattliche Erklärung

Danke an meine liebe Freundin, die mich und das öftere Umfunktionieren des Wohnzimmers zur Werkstatt fünf lange Jahre lang ausgehalten hat und mich während meines berufsbegleitenden Studiums so sehr unterstützt hat.

Danke an Hr. Ronald Stärz, der mich durch die Zusammenarbeit und durch seine Person auf den Weg gebracht hat wo ich mich heute befinde.

Danke an meinen Chef, Jury Löcker der mich mit seiner Leidenschaft und Ideen seit zwei Jahren kontinuierlich inspiriert.

Danke an Gernot, Dominik und Kevin die meine krummen ungaro-österreichische Sätze teilweise geradebiegen konnten.

Kurzfassung

Toaster haben seit deren Erfindung im Jahr 1893 bereits viele Optimierungsschritte hinter sich, jedoch haben sie aus der Entwicklung der Sensorik und digitale Technologie in Kombination mit Machine Learning im dem letzte Jahrzehnte wenig profitiert. In der Regel wird eine digitale Zeitschaltuhr verwendet, um den Toastvorgang abzuschalten. Dies führt dazu, dass die hintereinander gerösteten Toasts nicht den gleichen Bräunungsgrad aufweisen. Die hier präsentierte Arbeit sucht eine Lösung um die Wiederholungsgenauigkeit von Toastern signifikant zu erhöhen. Basierend auf verschiedensten Sensordaten in Kombination mit Machine Learning, kann der Toaster eine nächste Evolutionsstufe erreichen. Es wird ein Messsystem entwickelt um die Sensordaten aufzunehmen und diese während den Aufnahmen durch das HMI zu bewerten. Es wird Luftfeuchtigkeit, Temperatur, Rauch, Wasserstoff, Ethanol, VOC, Gewicht, sowie die Farbe während des Röstvorgangs gespeichert und über WLAN übertragen. Das entworfene System beinhaltet mehrere Module die durch einen differentiellen Bus verbunden sind. Da nicht alle Sensordaten zwingen notwendig sind, wird angestrebt mit reduzierter Anzahl an Sensoren ein bestmögliches Ergebnis zu erzielen. Das finale Konzept beinhaltet ein Convolutional Neural Network (CNN), welches in Echtzeit die momentane Sensordaten verarbeitet und die ideale Toastzeit bestimmt. Weiters wird eine Methode präsentiert um eine visuelle Vorschau des Brotes während dem Toastvorgang anhand den Sensordaten zu generieren.

Schlagwörter: Machine Learning, Smart Toaster, Data Acquisition, Neural Networks, Embedded

Abstract

Toasters have undergone many optimization steps since their invention in 1893, but they have benefited little from the development of sensor technology and digital technology in combination with machine learning in the past decade. A digital timer is usually used to switch off the toasting process. This means that the toasts roasted one After the other do not have the same degree of browning. The work presented here is looking for a solution to significantly increase the repetition accuracy of toasters. Based on a wide variety of sensor data in combination with machine learning, the toaster can reach a next level of evolution. A measuring system is being developed to record the sensor data and to provide human classification information during the recordings through a GUI. Humidity, temperature, smoke, hydrogen, ethanol, VOC, weight and the color during the toasting process are measured and transmitted via WLAN. The designed system includes several modules that are connected by a differential bus. Since not all sensor data are absolutely necessary, the aim is to achieve the best possible result with a reduced number of sensors. The final concept includes a convolutional neural network (CNN), which processes the current sensor data in real time and calculates the ideal toasting time. Furthermore, a method is presented to generate a visual preview of the toast during the toasting process, based on the sensor data.

Keywords: Machine Learning, Smart Toaster, Data Acquisition, Neural Networks, Embedded

Inhalte

1	EINLEITUNG	1
1.1	Problemstellung	1
1.2	Geschichte des Elektrischen Toasters	2
1.3	Bräunung	9
1.4	Neurale Netzwerke	10
2	THEORETISCHE GRUNDLAGEN	11
2.1	Was ist Machine Learning?.....	11
2.2	Perceptron	12
2.3	Multi-Layer Perceptron.....	14
2.4	Fehler (Loss) Funktion	14
2.5	Gradient Descent	15
2.6	Back Propagation	16
2.7	Stochastic Gradient Descent	18
2.8	Overfitting und Cross Validation	19
2.9	Konvolution	20
2.10	Aktivierungsfunktionen	22
2.11	Max Pooling	23
2.12	Dropout.....	23
2.13	Fitness	24
3	ENTWICKLUNG EINES MODULAREN MESSSYSTEMS	26
3.1	Mainboard	28
3.1.1	Spannungsversorgung.....	29
3.1.2	I2C	30
3.1.3	GPIO Expansion	32
3.1.4	Ansteuerung von Aktoren.....	32
3.1.5	ADCs	34

3.1.6	Schnittstellen.....	34
3.2	Versorgungsmodul.....	35
3.3	Analog Digital Converter Modul	37
3.4	Wägemesszellen Modul.....	38
3.5	Rohr Sensor Modul.....	39
3.6	Frequenz Konverter	41
3.6.1	Kapazität zu Frequenz Konverter.....	41
3.6.2	Licht zu Frequenz Konverter	42
3.7	IR Temperatur Sensor Modul.....	43
3.8	Farbsensor	44
3.9	Encoder	46
3.10	Diverse.....	47
3.10.1	ESP32 Programmer	47
3.10.2	Display	47
4	AUFBAU DES PROTOYPS	48
4.1	Waage in einem Toaster	51
4.2	Positionierung der Sensoren	52
4.3	Lüfterstrecke.....	56
4.4	Rauchsensor in einem Toaster.....	57
4.5	Fertiger Prototyp, Übersicht der Sensorik.....	58
4.6	Firmware	59
4.6.1	DAQ	60
4.6.2	SD-Karte.....	62
4.6.3	Wi-Fi	62
4.6.4	GUI.....	64
4.6.5	Heizer.....	65
4.7	PC-Software.....	66
5	WIE FRAGT MAN DIE MASCHINE OB DER TOAST FERTIG IST? ...	68
5.1	Aufnehmen der Daten	68

5.2	Preprocessing.....	72
5.2.1	Random Slicing	74
5.2.2	Time Slicing.....	74
5.2.3	Random Slicing With Padding.....	75
5.3	Modellauswahl.....	76
5.3.1	Binäre Klassifikation.....	77
5.4	Relevanz einzelner Sensoren.....	84
5.5	Random Forests	87
5.6	Toast Decoder	88
5.6.1	Preprocessing	88
5.6.2	Modell	90
5.7	Live Predictor.....	91
6	ZUSAMMENFASSUNG UND AUSBLICK	93
6.1	Zusammenfassung der Ergebnisse	93
6.2	Errata	96
6.3	Ausblick.....	96
7	LITERATUR- UND QUELLENVERZEICHNIS	X
8	ABKÜRZUNGSVERZEICHNIS.....	XVIII
9	ABBILDUNGSVERZEICHNIS	XX
10	TABELLENVERZEICHNIS	XXVI
ANHANG A1		XXVIII
ANHANG A2		XXXVII
ANHANG A3		XXXVIII
ANHANG A4		XXXIX
ANHANG A5		XLI

ANHANG A6 XLV

ANHANG A7 XLV

Anhang A7-1XLV

Anhang A7-2XLV

Toaster Oven Haiku

Cursed toaster oven,
Lighter setting, darker bread,
Some kind of cruel joke.

PBrigitte

1 Einleitung

1.1 Problemstellung

Die meisten die ihren Toast am Morgen genießen wollen, kennen das Problem. Die Toastzeiten variieren, speziell nach mehreren Toastvorgängen, aufgrund der im Toaster gespeicherten Wärme. Die Einstellung am Toaster, welche beim zweiten oder dritten Toast ein erwünschtes Ergebnis liefert, gibt in der Regel das erste Röstbrot, viel zu früh aus. Man muss den Röstvorgang ein zweites Mal starten um das gewünschte Ergebnis zu erhalten. Sollte man die richtige Einstellung für einen perfekten ersten Toast gefunden haben, verkohlen dafür die nachfolgenden Brötchen. Um beim Toasten kein verkohltes Ergebnis zu erhalten, stoppt man entweder die Zeit oder man beobachtet den Toaster genau.

Die Zielsetzung dieser Arbeit ist es, die Lösung dieses Problems durch Einsatz von neuronalen Netzwerken zu finden. Es soll ein Prototyp entstehen, welcher bei aufeinanderfolgenden Toasts die gleichen Ergebnisse im Sinne vom Bräunungsgrad liefert. Unabhängig von aktueller Temperatur, Luftfeuchtigkeit im Geräteinneren und Anzahl von vorher getoasteten Brotscheiben.

Ziel dieser Arbeit ist es nicht, den Toaster als Konsument Produkt zu überarbeiten oder die angeführten Probleme mit mechanischen oder thermodynamischen Analysen zu eruieren:

- Toastoberfläche unregelmäßig gebräunt
- Die Seiten des Toasts werden unterschiedlich gebräunt
- Langsames Toasten
- Konstruktionsprobleme von Toaster
 - Gerät wird zu heiß, schwer zu reinigen, erwartete Lebensdauer zu gering, schmelzende Plastikteile im Geräteinneren

1.2 Geschichte des Elektrischen Toasters

Er hat sich in jedes Zuhause eingeschlichen und glänzt oder staubt auf jeder Küchenplatte, unscheinbar und selbstverständlich. Ein wahres Denkmal für die Elektrifizierung der Menschheit, der elektrische Toaster. Seit 1903 hat sich an den wesentlichsten Aufbaukomponenten nichts geändert, es steckt jedoch über ein Jahrhundert an Geschichte und Entwicklung drin.

Das Wort Toast stammt aus dem lateinischen torrere – trocknen / brennen. Nachdem die Ägypter das Brot erfunden [5] haben, wurde es gebräuchlich Brot zu rösten, um dessen Haltbarkeit zu verlängern. 1893 gelang es Alan McMaster den elektrischen Toaster zu erfinden. Damals war der Toaster jedoch noch kein kommerzieller Erfolg. Dafür fehlte das hitzebeständige Strahlerelement aus NickelChrome-Draht, welches durch Albert Marsh [48] 1905 erfunden wurde.

Die ersten Jahrzehnte des elektrischen Toasters boten viel Platz für den Erfindergeist. Hunderte von verschiedenen Modellen stammten aus kleinen Werkstätten und zahlreiche Patente wurden angemeldet. Das Hauptproblem war vorerst das beidseitige Toasten, hierzu wurden viele Konzepte ins Leben gerufen.

Abbildung 1-1 zeigt einen Percher (Modell D-12 von General Electric). Dieser Toaster war der erste kommerzielle Erfolg und verwendete schon den innovativen Nickelchrome- Draht als Heizelement. Er wurde 1908 unter US951287A patentiert. Das Brot wird an das Gitter vor dem Heizelement gelegt und von Hand gedreht. In Abbildung 1-1 (rechts) ist ein Pincher ersichtlich, vom ungarischen Hersteller Elekthermax aus den 1930er Jahren. Das Brot wird bei Pinchern mit einem von Federn gespannten Seitenpanel zum Gitter vor dem Heizelement geklemmt. Das Umdrehen musste ebenfalls noch per Hand erfolgen.



Abbildung 1-1: General Electric D-12 - Percher und Elekthermax VP-3 – Pincher [56]

Abbildung 1-2 zeigt das Patent US1108552A eines Turn Overs von Copeman von 1913. Copeman vereinfachte die Bedienung indem beim Wegschwenken des Seitenpanels das Brot umkippt, entlang des Gitters rutscht und somit um die horizontale Achse umgedreht wird.

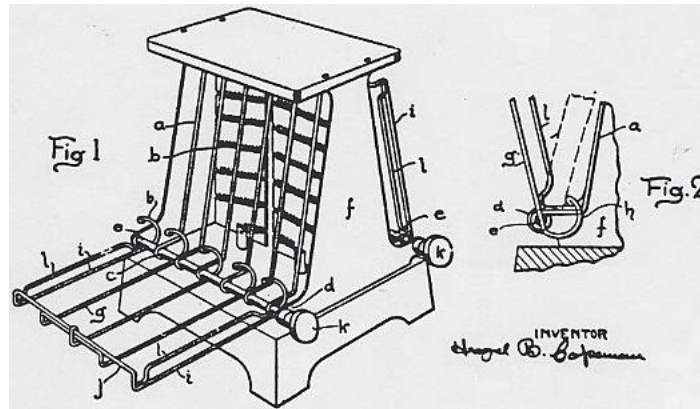


Abbildung 1-2: Copeman Turnover [US1108552A]

Bei Droppern wie dem Heatmaster No.692 in Abbildung 1-3, wird das Brot nach unten durch eine Falltür ausgegeben. Beim Flatbed liegt der Toast auf dem Gitter, wie bei einem Grill. Abbildung 1-3 (rechts) zeigt ein Modell von Rowenta.

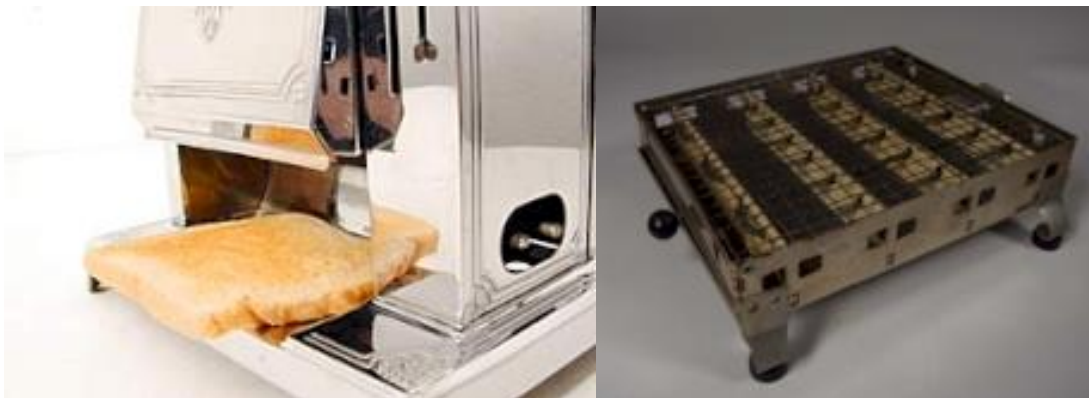


Abbildung 1-3: Heatmaster No.692 Dropdown [56] und Flatbed [57]

Turners und Swingers (Abbildung 1-4) – Das Brot wird in einen Käfig geklemmt und um die horizontale bzw. vertikale Achse gedreht. Bei dieser Bauart verringerte sich die Arbeitszeit des Bedieners um die Hälfte, um das Brot durch händische Betätigung eines Hebels zu drehen.



Abbildung 1-4: Universal E 947 [56] Turner

Beim Slider wird das Brot horizontal in den Toaster geschoben – wie in der Abbildung 1-5 beim E9422 von Universal. Beim Tipper wird das Brot in einem Winkel in den Toaster eingeführt. (Abbildung 1-5).



Abbildung 1-5: Universal E 9422 - Slider[56], Landers, Frary & Clark, circa 1930s. – Tipper[70]

Der Drive Thru (Model G) von Toast-O-Lator in Abbildung 1-6 beförderte in den 1940er das Brot motorisiert und horizontal durch den Toaster. Wenn es auf der Gegenseite rausfällt, ist das Brot fertig (Abbildung 1-6).

Der heute meist verbreitete Toaster Typ ist der Popup Toaster, erfunden in 1920 von Charles Strite (Abbildung 1-6 - rechts). US1394450A.



Abbildung 1-6: Toast-O-Lator Model G - Drive Through, Strites Pop Up [56]

Ein Turn Up Toaster wird in dieser Arbeit eine zentrale Rolle spielen. Viele Modelle von Dualit und Rowlett verwenden diese Bauart. Ein weiterer Typ von Toastern sind Conveyor Toaster, die sich in der Gastronomie verbreitet haben.

US2459170A – 1942, US2667828A – 1948, der Sunbeam Radiant Control Toaster bietet Funktionen die heute kaum noch vorhanden sind, ausschließlich basierend auf mechanischen Mitteln. Nach Einlegen des Brotes, wird der Toast automatisch abgesenkt und nach der automatischen Abschaltung angehoben. Das Einlegen des Brotes betätigt einen Schalter und beginnt den Toastvorgang. Der auf einen federnden Isolator gewickelte Nichrome Draht verlängert sich durch die Erwärmung kann sich die Feder (In Abbildung 1-7 mit 112 markiert) in eine Ruhelage positionieren. Dadurch wird eine vertikale Bewegung hervorgerufen, welche über drei Gelenke 175-fach übersetzt wird und zum Absenken und Anheben des Toastes genutzt wird. Die automatische Abschaltung erfolgt durch einen bi-metallischen Schalter, welcher zwischen den seitlichen Heizelementen positioniert ist. Dieser Schalter wird von allen Seiten isoliert und dadurch wird der Thermostat von der reflektierten Hitze der Toastoberfläche erwärmt. Daher stammt der Name Radiant Control des Toasters.

Wie Artman [7] beschreibt – die großen Jahre des Toasters waren mit Ende der 60er vorbei und das Streben nach Perfektion, wurde gegen Streben nach günstiger umsatzorientierter Massenproduktion getauscht.

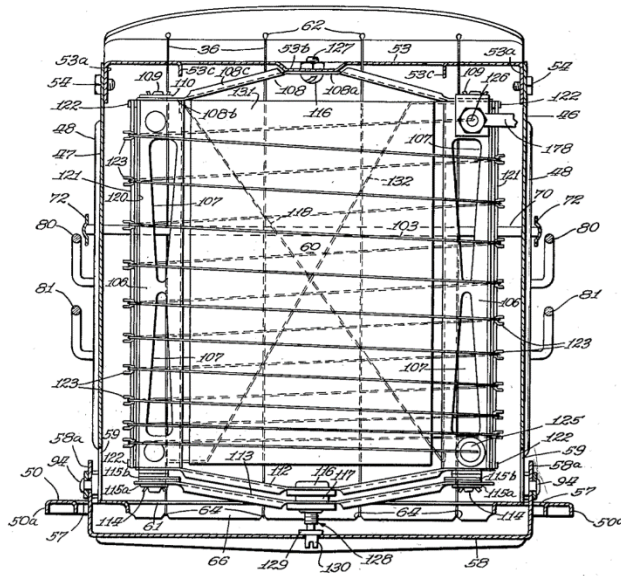


Abbildung 1-7: Sunbeam Radiant Toaster [US2459170A]

GB1342833A - 1970 Rowenta baut einen NTC Sensor an die Toastraumwand, welcher die verwendete Zeitschaltlogik, bestehend aus einem RC Glied und einem Programmable Unijunction Transistor erweitern soll. Wird der Threshold des PUTs durch die erhöhte Temperatur und verminderten Widerstand des NTCs früher erreicht, verkürzt sich die Toastzeit.

Da Bimetall-Schalter, als Zeitregler verwendet hauptsächlich von konvektiver Hitze und Strahlungswärme beeinflusst werden, bieten sie bei einem bereits vorgewärmten System keine zuverlässige Lösung. Toaster, die Solid State Temperatur Sensoren verwenden, beachten vorangegangene Toastvorgänge nicht und bieten auch keine geeignete Lösung, laut US5128521 - 1986. Dieses schlägt vor, vorprogrammierte Lookup Tables zu verwenden. Die Toastzeit soll, basierend auf der letzten Toastdauer und der Anzahl von Toast Zyklen seit Einschalten, anhand von Zeitmessung ideal gewählt werden können. Die Lösung von US4510376A-1982 scheint einige Jahre früher die gleiche Logik bei einem Drop Down Toaster zu verwenden. GB2318502B - 1994 erweitert diese Art von Toastzeitbestimmung indem durch den „Toast Color Selection“ Drehschalter ein zusätzlicher Toastzeitverlängerungsfaktor eingeführt wird. Weiters wird ein Temperatursensor im Gehäuse des Toasters (außerhalb des Toastraums) platziert, welcher die Toastzeit um eine definierte Dauer verringert, wenn die Umgebungstemperatur 30 °C überschreitet. Dies soll die Sicherheit bieten, dass auch ein Toaster, welcher schon warm ist und auch kurzzeitig die Verbindung zur Stromversorgung verloren hat, die richtige Zeitkompensation aus den gespeicherten Tabellen laden kann.

US4913046A-1987 Braun patentierte eine Lösung mit „radiation guide passageways“. Damit wird durch ein labyrinthförmiges Reflektionssystem die IR-Strahlung der Toasteroberfläche zum Sensor mehrmals abgelenkt, um die Wärmebelastung des Sensors zu verringern.

US4976194A 1988 verwendet einen IR Temperatursensor und eine OP Amp Schaltung um temperaturgerechte Kompensationen auszuüben.

US5692432A - 1995 Philips verwendet einen Temperatursensor um ein Steuerungssignal bei vorprogrammierten Schwellwerttemperaturen auszulösen. Die vom User vorgegebene Zeit beginnt erst dann zurückzuzählen, wenn die erste Schwellwerttemperatur erreicht wurde (100°C). Bei einem anderen Operationsmodus wird bei dem zweiten, höheren Threshold 200°C die Heizleistung zurück geregelt. Das Patent diskutiert mehrere Möglichkeiten von Temperatursensoren. Unter anderem auch, durch Anlegen eines mit Temperatursensor versehenen Metallgitters direkt auf eine Oberfläche, die Temperatur bestimmen zu können.

US6198078B1 – 2001 Cusinart schlägt ebenfalls einen Temperatursensor vor, welcher beim Start des Toastzyklus ausgelesen wird und die Toastdauer beim Überschreiten eines Temperaturschwellwertes um 30 - 40% reduziert.

GB2515075A - 2013 Improved Electric Toaster schlägt einen Aufbau vor, ähnlich wie ein Sandwich Ofen. Sie verwenden eine Kombination aus Infrarot und Mikrowellen als Heizelemente.

Eine Technologie die sich in Serienprodukten befindet wird in GB2516476A 2015 von Dualit erwähnt. der Toaster verfügt über einen im Toastraum platzierten Temperatursensor. Durch (1) wird die Temperatur $T1$ zur benötigten Verkürzung der vorgegebenen Zeit Δt verknüpft, um gleichmäßige Ergebnisse zu erreichen.

$$\Delta t = \frac{T1 - a}{b} \quad (1)$$

Als ideale Richtwerte gibt das Patent für a den Bereich von 20-22°C an und für b den Bereich zwischen -1.2 und -1.5.

CN209315631U -2019 Intelligent Control Toaster sieht die Lösung in der Verwendung eines Bildsensors. Breville besitzt zahlreiche Patente und verkauft momentan den Smart Toast, US2014352549A1 - 2014, US2012167777A1 – 2012. Diese Toaster verfügen über einen motorisierten Toastlift.

Eine wichtige mechanische Verbesserung von Pop Up Toastern, ist die Brotzentrierung, welche das Toastbrot in einer geraden Stellung in mittlerer Position der beiden Heizelemente fixiert.

US2016287014A1 2016 schlägt als mechanische Verbesserung die Verwendung von keramischen Heizelementen vor.

Toaster mit durchsichtiger Wand gab es schon während des Toasterbooms in den 1930er Jahren. Die Idee die Toasterwand als Heizstrahler zu verwenden, wurde 1946 patentiert. Bei der Erfindung US2564706A war eine halbdurchsichtige Leistungswiderstandplatte gleichzeitig Heizelement und Außenwand. Die Weiterentwicklung in dem Halbleiterzeitalter scheint GB2470292A 2010,

GB2527939 2014, von MorphyRichards zu sein. Dort werden Glaswände mit einer Metalloxiddünnschicht als Heizelement verwendet. Weitere kommerzielle Beispiele von durchsichtigen Toastern sind der Magimix Vision oder der Cuisiart ViewPro, welche beide mit Quarz Heizelementen arbeiten.

Als Infrarotstrahlungsquellen im Toaster, werden entweder direkt ausgesetzte hitzebeständige Metalllegierungen (wie Ni-Cr, Fe-Cr-Al) oder Rohrheizer aus Quarz- und Keramikglas verwendet. [50] Ein Vorteil des Quarz-Glas gegenüber Keramik-Glas und Mica Heizelementen ist, dass keine Stromschlaggefahr beim Berühren des Elementes beispielsweise mit einer Gabel vorliegt. Quarz Elemente werden aber langsamer rotglühend, was zu einer längeren Toastdauer führt. [50][51]

Newborough untersuchte 1987 die ersten 60 Jahre der Entwicklungen des elektrischen Toasters und beschreibt die üblichen Probleme, wie Regelungsmethoden und die thermische Reaktion in einem Toaster.[51] 1988 [52] näherte er den Toaster aus dem Aspekt der Effizienzverbesserung des Wärmetransfers an und testete geschlossene Toastergehäuse sowie unterschiedliche Abstände zwischen Brot und Heizelement (Separationsabstände).

Bis zu den 1940er Jahren waren Timer mit Uhrmechanik die üblichen Zeitregler, danach übernahmen Bimetallschalter diese Aufgabe. Mit Erscheinung des Transistors entwickelte sich die Abschaltung ebenfalls weiter und wurde durch RC Gliedern gelöst. Durch die Verbreitung von Mikrocontrollern und ASIC wurde der Toaster auch digitalisiert. Heute verwenden die meisten erhältlichen Toaster dedizierte ICs wie PT8A2514A, CMS12560, GA5210PL, PT8A2511, PT8A2512NE, BCT5511 etc.

Die meisten heute verkauften Toaster verwenden eine Toastausgabemechanik wie US3956978A - 1976. Durch Herunterpressen gegen die Federkraft des Hebels wird die Spule eines Elektromagneten unter Strom gesetzt und schließt den magnetischen Kreis mit ferromagnetischem Metall, welcher innen am heruntergedrückten Hebel befestigt ist. Ist der eingestellte Zeitpunkt erreicht, wird der Stromkreis des Elektromagneten unterbrochen und die Federkraft befördert den Toast nach oben.

Eine weitere Komponente des Toasters ist der Isolator, der die Aufwicklung des hitzebeständigen Drahtes ermöglicht. Hier hat sich die Verwendung von Mikanit – Pressglimmer durchgesetzt. Hierzu werden Abfälle des Blattglimmers mit einem Klebstoff verpresst. [47]- Dieses Material ist je nach Ausgangsgestein temperaturbeständig, Moscovita bis 600°C und bei Phlogopite bis 900°C. [49]

1.3 Bräunung

Toastbroat besteht aus Weizenmehl, Fett (Butter oder Margarine), Milch und Zucker. Aufgrund von lebensmittelrechtlichen Kriterien darf bei Butters toast nur Butter als Fett verwendet werden. Die Porenstruktur des Brötchens kann je nach Herstellungsprozess [5] Unterschiede aufweisen.

Verantwortlich für die Bräunung des Brotes, ist neben der Karamellisierung, die Maillard Reaktion. Die Maillard-Reaktion ist eine nicht-enzymatische Bräunungsreaktion. Hierbei werden Aminosäuren und reduzierte Zucker unter Hitzeeinwirkung zu neuen Verbindungen umgewandelt. [6]. Die Reaktion tritt ab ca. 140°C ein, bei weiterer Steigerung der Temperatur auf ca. 180°C beginnt die Pyrolyse, welche zum Verkohlen des Toastes führt. Dabei entstehen Acrylamide. Zur Risikoabschätzung bezüglich des Konsums von Acrylamiden existieren Forschungsergebnisse [55]

Die European Food Safety Authority (EFSA) warnt vor dem Toasten mit zu hoher Temperatur aufgrund gesundheitlicher Aspekte. Sie empfiehlt eine Farbe des fertigen Produkts von gold-gelb oder heller. [54]

Der perfekte Toast. „Perfekt“ ist eine individuelle Präferenz von Konsumenten und wird von vielen subjektiven Qualitätsmerkmalen bestimmt, wie beispielsweise das Verhältnis der inneren und äußeren Elastizität eines fertigen Toastbrotes.

Für diese Arbeit wird eine Definition für den perfekten Toast benötigt, damit die Bewertung der gerösteten Toasts möglich wird. Es wird basierend auf den vorgestellten gesundheitlichen Empfehlungen von EFSA so bestimmt, dass das fertige Brot keine dunkel-braunen oder schwarzen Stellen aufweisen darf und eine gold-gelbe oder hellere Farbe haben muss.

1.4 Neurale Netzwerke

Die Überlegung eine statistische Methode zu entwickeln, welche dem Funktionsprinzip eines menschlichen Neurons ähnelt, wurde schon 1943 von McCulloch und Pitts publiziert [18]. Schmidhuber beschreibt wie die Methoden Back Propagation [37], Gradient Descent und Deep Learning [30] historisch entstanden sind. LeCun trainierte 1989 schon einen Vorgänger der heutigen Convolutional Neural Netzwerke, für Schrifterkennung mit Backpropagation, welches später als LeNet [9] bekannt wurde.

Zur Anerkennung und Verbreitung des Verfahrens, führten Ereignisse wie die Leistung des AlexNet [10] Convolutional Neural Networks bei dem ImageNet Wettbewerb in 2012 [2]. Das Netz erreichte eine mehr als 10% geringere Fehlerrate gegenüber Mitbewerbern mit anderen Machine Learning Verfahren (Bspw. Oxford mit SVM Modelle). Dieses Netz verfügte über 62.3 Millionen Parameter, fünf Convolutional Layer und benötigte für jeden Feed Forward mehrere Millionen Rechenschritte. Das Trainieren eines solchen Netzes war nur durch GPU Utilisierung möglich. Weitere bekannte Convolutional Neurale Netzwerke der letzten Jahre waren: ZFnet - 2014[12], GoogLeNet - 2015[11], VGGNet – 2015 [14], RESnets - 2016 [13], DenseNet - 2016[31], Lenet-5 – 2019 [17]

In 2017 erreichte der Preis von Grafikkarten mit einer Billion Floating Point Operationen pro Sekunde (GFLOPS) 0.03\$ [3]. Heute existieren Machine Learning Bibliotheken wie Tensorflow, Pytorch, OpenNN, Caffe2 etc. welche alle GPU Unterstützung enthalten. Hersteller der Hardware Industrie, wie beispielsweise Nvidia, stellen explizite Software (CUDA) [4] und Hardware (Tesla, Jetson, Titan) zur Verfügung, welche auf den Einsatz von Neuralen Netzwerken abzielen.

Zum Zeitpunkt der Recherche für diese Arbeit, wurden keine Veröffentlichungen über den Einsatz von Machine Learning Algorithmen, wie Deep Learning für Toaster oder Brotbackgeräte, zur Bestimmung von Backqualität und -dauer, gefunden.

Ein vergleichbares Experiment wurde 2013 für Waffeln und Popcorn durchgeführt. Die Sensorik bestand aus einer elektrischen Nase (Multi Cell Gas Sensor) und dabei wurden Algorithmen wie Random Forests und Logistic Regression eingesetzt. Das Ziel war ebenfalls den Backzustand zu bestimmen [39].

Ein aktuell veröffentlichtes Experiment [72] zur Bestimmung von Bräunungsgrad von Kuchen in konvektiven Öfen, zeigte bessere Erfolge durch den Einsatz von NN als nonlinear polynomial Modellen.

2 Theoretische Grundlagen

Dieses Kapitel soll die Grundlagen für die Anwendung einer Klassifikation von Sensordaten mit Hilfe von Multi-Layer Neuronale Netzwerke beschreiben. Abbildung 2-1 stellt dar, dass sich diese Arbeit nur mit einem sehr engen Umfeld des Machine Learnings befasst.

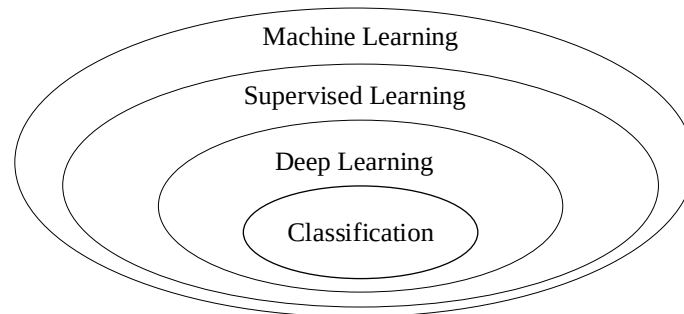


Abbildung 2-1: Umfeld dieser Arbeit, angelehnt an [20]

2.1 Was ist Machine Learning?

Beim maschinellen Lernen handelt es sich um eine Kombination von Algorithmen, die mit einem Input und mit einer Erwartung des menschlichen Anwenders arbeiten. Sie sollen den Zusammenhang zwischen der Eingabe und dem gewünschten Ergebnis selbstständig ermitteln und damit einen Vorgang des Lernens entwickeln. Dieses Lernen wird oft auch Trainieren genannt, welches mathematisch als ein Optimierungsproblem dargestellt wird. Die Aufgabe des Anwenders ist nicht mehr, ein Programm mit logischen Funktionen zu schreiben, die zum gewünschten Ergebnis führen. Vielmehr ist es die Aufgabe, den Input – beispielsweise ein Datensatz bestehend aus Bildern oder Sensordaten – und das angestrebte Ergebnis (Ausgang) für eine Maschine verarbeitbar vorzulegen.

Je nach verwendetem Verfahren, kann der konkret gewünschte Ausgangszustand auch verschwiegen werden, wie es bei einigen Arten vom Reinforcement Learning oder beim Unsupervised Learning der Fall ist.

Beim Reinforcement Learning wird kein richtiges Ergebnis vorgegeben, es wird aber das ausgegebene Ergebnis bewertet, wodurch die Maschine lernen soll. Beim Unsupervised Learning erhält die Maschine nur Inputs und soll während dem Lernprozess die Zusammenhänge und Strukturen zwischen den gezeigten Inputs erlernen.

Das gewünschte Ergebnis dieser Arbeit, ist als „Der perfekte Toast“ definiert. Dies soll durch Supervised Learning bewertet und an die Maschine zurückgeführt werden.

Mitchell [21] beschreibt den Begriff Lernen wie folgt: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Die auszuführenden Tasks können unterschiedlich sein, zum Beispiel: Regression, Übersetzung, Anomalie Detektion oder wie bei dem hier vorliegenden Anwendungsfall, Klassifikation. Das Lernergebnis muss geprüft werden, hierzu muss die Qualität / Güte / Leistung / Erfolg gemessen werden können. Diese Leistung kann an Faktoren festgehalten werden wie der Genauigkeit der Klassifizierung.

Die Bestimmung der Performance erfolgt idealerweise mit Daten, welche während des Trainingsprozesses nicht zum Lernen verwendet wurden. Diese haben die Erfahrung der Maschine noch nicht bereichern können. Daraus folgend ist die Erfahrung der Teil eines Datensatzes, die der Maschine während dem Lernen gezeigt wird.

Klassifizierung. In diesem Fall wird das Computerprogramm gefragt, einen Eingang zu einer von k Kategorien zu zuordnen. Es wird eine Funktion approximiert, welche die Eingangsvariablen x zu den diskreten Ausgangsvariablen y verbindet. Klassifikation wird dem Supervised Learning zugewiesen, da hier neben den Trainingsdaten auch die Zielwerte für die Modellbildung einbezogen werden.

2.2 Perceptron

Abbildung 2-2 zeigt ein Perceptron. Die Eingangsvariablen $x_0 \dots x_n$, der Bias b und der Ausgang y bilden mit der Aktivierungsfunktion f den Zusammenhang (2).

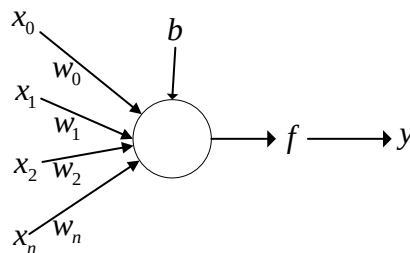


Abbildung 2-2: Perceptron dessen Ausgang in eine Aktivierungsfunktion geführt wird

Jeder Eingang ist mit dem Perceptron verbunden. Eine gewichtete Summe wird den Eingängen zugeführt, die durch den Bias b verschoben wird. Es entsteht eine lineare Funktion:

$$y = \sum_{i=0}^{n-1} x_i w_i + b \quad (2)$$

Der Perceptron selbst kann nur linear separierbare Probleme lösen. Soll beispielsweise ein einzelner Perceptron die Funktionalität eines NAND Gatters aufnehmen, werden die Parameter bestehend aus den Gewichten w_i und Bias b wie folgt bestimmt:

$$w_1 = w_2 = -2 \quad (3)$$

$$b = 3 \quad (4)$$

$$y = x_1 w_1 + x_2 w_2 + b \quad (5)$$

Aus der Gleichung (5) folgend die entsprechenden Ergebnisse:

Tabelle 2-1: NAND Wahrheitstabelle

x_1	x_2	y
1	1	0
1	0	1
0	1	1
0	0	1

Wie in der Abbildung 2-3 zu sehen ist dieses Problem linear separierbar.

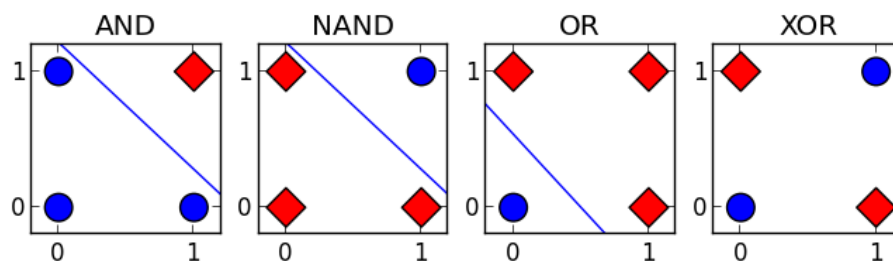


Abbildung 2-3: Separierbarkeit [34]

Mit der Einführung einer Nichtlinearität in Form einer Aktivierungsfunktion f werden auch nichtlinear separierbare Probleme (wie XOR in der Abbildung 2-3) annäherbar. Die gewichtete Summe (2) wird in die Aktivierungsfunktion geführt und ergibt den Ausgang.

$$y = f \left(\sum_{i=0}^{n-1} x_i w_i + b \right) \quad (6)$$

Die Gewichte und der Bias zusammen ergeben die Parameter, die durch das Training optimiert werden sollen.

2.3 Multi-Layer Perceptron

Durch Verwendung von multiplen Perceptronen, in mehreren Schichten angeordnet, kann jede stetige Funktion abgebildet werden [26]. Abbildung 2-4 zeigt ein Multi Layer Perceptron Netzwerk bestehend aus mehreren Dense / Fully Connected Layers.

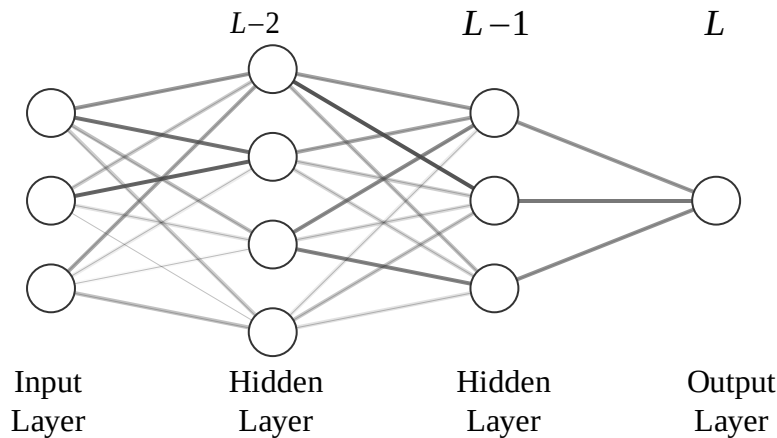


Abbildung 2-4: Multi Layer Perceptron

Die Durchsichtigkeit der Verbindungslinien, stellt symbolisch das Verhältnis der Gewichte zueinander dar. Die Aktivierung des Hidden Layer L wird von der Aktivierung der $L - 1$ Hidden Layer bestimmt. Ein Element in einem Hidden Layer hängt also nicht mehr wie in (2) gezeigt von den Eingangsvariablen ab, sondern nur von der Aktivierung des vorhergehenden Layers.

2.4 Fehler (Loss) Funktion

Unter Training von neuronalen Netzen versteht man, dass die Parameter, wie im NAND Beispiel vorgestellt, durch die Minimierung der Fehlerfunktion so kalibriert werden, dass sie das gewünschte Ergebnis am Ausgang annähern. Die Fehlerfunktion C kann an verschiedenen Wegen bestimmt werden. Beispielsweise kann die mittlere quadratische Abweichung (MSE) verwendet werden, welche aus der quadratischen Differenz zwischen dem wahren Wert $\hat{y}$ und der Aktivierung des letzten Layers besteht. Formel (7) zeigt den MSE für den letzten Hidden Layer eines Netzes mit N Neuronen in diesem Layer.

$$C(w, b) = \frac{1}{N} \sum_{i=0}^{n-1} (\hat{y}_i - (x_i w_i + b))^2 \quad (7)$$

Die Cross Entropy Fehlerfunktion wird oft bei Klassifikationsproblemen verwendet. Gleichung (8) zeigt für K Klassen im Output Layer des Netzes.

$$C(w, b) = - \sum_{i=0}^{K-1} (\hat{y}_i \log(x_i w_i + b)) \quad (8)$$

Die Minimierung der Funktion C erfolgt mit der Methode des stochastischen Gradientenabstiegsverfahren und die Gradienten werden durch den Backpropagation Algorithmus errechnet.

2.5 Gradient Descent

Dieser Algorithmus kann mit einem blinden Bergsteiger verglichen werden, der den Weg ins tiefste Tal der Umgebung, durch die Erkennung der Steigung des Berges finden möchte [27].

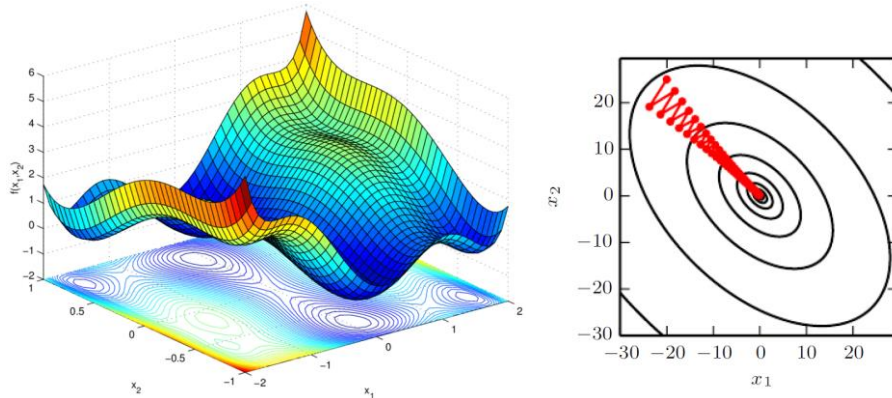


Abbildung 2-5: Six Hump Camel Back Function [35], SGD Iterative Darstellung [20]

In der Abbildung 2-5 wird eine dreidimensionale Fläche auf 2 Dimensionen projiziert. Die geschlossenen Flächen stellen die globalen und lokalen Minima dar. Das Ziel ist es, durch schrittweise Verstellung der Parameter x_1 und x_2 eines von diesen Minima zu erreichen. Die Verfolgung der Gradienten (Abbildung 2-6) garantiert nicht, dass die globalen Minima gefunden werden.

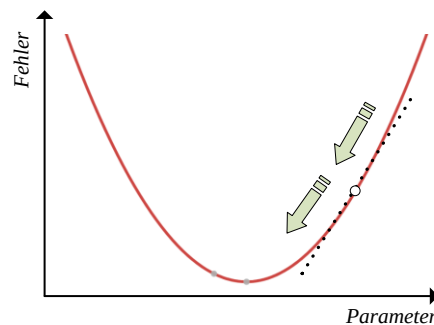


Abbildung 2-6: Verfolgung der Gradienten

Um den Gradienten von (7) zu erhalten, wird die partielle Ableitung nach w und b berechnet. Bei der Suche nach den Minima kann nun eine weitere stückweise Annäherung erfolgen, indem in die entgegengesetzte Richtung des steilsten Anstieges die Gewichte und Bias mit einer definierten Lernrate weiterbewegt werden. Die wahren Ausgaben werden mit $\hat{y}$ gekennzeichnet.

$$-\nabla C = -\dot{C}(w, b) = - \begin{bmatrix} \frac{dC}{dw} \\ \frac{dC}{db} \end{bmatrix} = - \begin{bmatrix} \frac{1}{N} \sum -2x_i(\hat{y}_i - (x_i w_i + b)) \\ \frac{1}{N} \sum -2(\hat{y}_i - (x_i w_i + b)) \end{bmatrix} \quad (9)$$

Der Gradient ergibt sich aus den partiellen Ableitungen aller Parameter (Gewicht und Bias) (9) der Kostenfunktion. Das entgegengesetzte Vorzeichen stammt von der Überlegung, dass die Schritte eine Annäherung der Minima anstreben, wie Abbildung 2-5 für zwei Parameter zeigt. Diese partielle Ableitung für den Parameter bestimmt somit nicht nur die Richtung der Verstellung, sondern gleichzeitig auch wie relevant eine Änderung des Gewichtes für das Gesamtergebnis ist. Um die Parameter verändern zu können, muss noch eine Konstante eingeführt werden. Die Learning Rate bestimmt, wie groß die übernommene Änderung in dem jeweiligen Optimierungsschritt sein soll. In dem Code Snippet (Anhang A7) wird diese Konstante als lr an die Funktion `updateWeights(...)` übergeben.

Um dieses Verfahren auch bei Verwendung von Aktivierungsfunktionen anwenden zu können, müssen diese ebenfalls differenzierbar sein. Bei realen Anwendungen verfügen Netze über wesentlich mehr Schichten und Parameter als in Abbildung 2-5 oder Abbildung 2-6 dargestellt. Hier ist die visuelle Vorstellung nicht mehr möglich. Bei dem Beispiel konnte der Fehler direkt für den jeweiligen Parameter ermittelt werden, für die zwischenliegenden Schichten eines Multi Layer Netzwerkes sind hierzu aber zusätzliche Operationen notwendig. Um die Berechnung der Fehler und die Optimierung auch für diese Parameter zu ermöglichen, wird zunächst der Back Propagation Algorithmus vorgestellt, welcher genau dieses Problem löst. Die Idee liegt darin, den Fehler, welcher am Ausgang des Systems vorliegt, auf dem gesamten Modell zu verteilen, beziehungsweise auf den vorherigen Layer des Netzes zurückzuführen.

2.6 Back Propagation

Die Back Propagation wird im Folgenden an einem Netz mit einzelnen Neuronen in den Layern vorgeführt. Anschließend wird das Beispiel auf die generalisierte Form für mehrere Neuronen in dem Layer erweitert. Die intuitive Herleitung mittels der Kettenregel wurde von Dreyfus 1963 [37] gezeigt und von Werbos für NN angewendet [1].

Für die beim Perceptron vorgestellte gewichtete Summe wird die Variable z eingeführt (10). Die Aktivierung a mit dem Super Skript $a^{(L)}$ (11) stellt die Aktivierung des L -ten Layers dar. In diesem Beispiel besteht die Kostenfunktion (12) aus der quadratischen Differenz zwischen der letzten Aktivierung und dem wahren Ausgang $\hat{y}$.

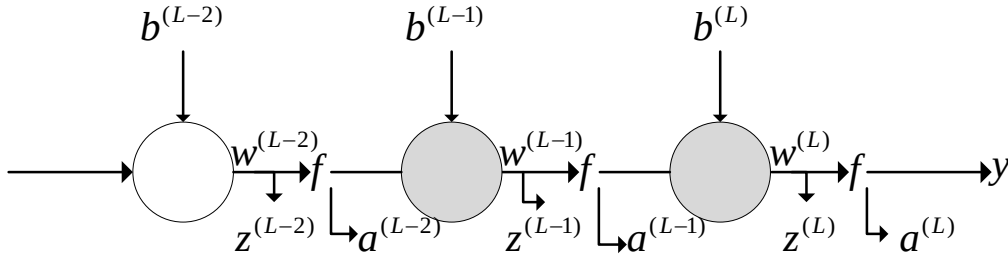


Abbildung 2-7: Single Perceptron Netzwerk – Backpropagation

$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)} \quad (10)$$

$$a^{(L)} = f(z^{(L)}) \quad (11)$$

$$C_0 = (a^{(L)} - \hat{y})^2 \quad (12)$$

Wie Abbildung 2-7 zeigt, hängt die Aktivierung des letzten Layers $a^{(L)}$ von der Aktivierung des vorherigen Layers $a^{(L-1)}$ ab. Der Algorithmus bestimmt die Sensitivität der Fehlerfunktion auf die Änderungen der einzelnen Parameter des Netzes.

Wie (13) zeigt, soll die Änderung der Fehlerfunktion durch Veränderung des Gewichtes $w^{(L)}$ bestimmt werden. Wenn man dem Netz entlang geht ist es ersichtlich, dass eine Änderung von dem Gewicht $w^{(L)}$ zu einer Änderung in $z^{(L)}$ führt. Dies führt wiederum zur Änderung am $a^{(L)}$, welches sich direkt auf die Kostenfunktion C_0 auswirkt. Formuliert man dies in einer Gleichung ergibt sich:

$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C_0}{\partial a^{(L)}} \quad (13)$$

Die Berechnung der partiellen Ableitungen für die Terme an der rechten Seite wird mit den Zusammenhängen (14), (15) und (16) dargestellt.

$$\frac{\partial C_0}{\partial a^{(L)}} = 2(a^{(L)} - \hat{y}) \quad (14)$$

$$\frac{\partial a^{(L)}}{\partial z^{(L)}} = \dot{f}(z^{(L)}) \quad (15)$$

$$\frac{\partial z^{(L)}}{\partial w^{(L)}} = a^{(L-1)} \quad (16)$$

Somit wurde eine Komponente des Gradientenvektors bereits berechnet. Diese Berechnung erfolgt für jeden weiteren w sowie für die b Parameter des Netzes, hierzu wird (13) beispielsweise so modifiziert, dass die Sensitivität der Kostenfunktion auf die Änderung des Bias untersucht wird:

$$\frac{\partial C_0}{\partial b^{(L)}} = \frac{\partial z^{(L)}}{\partial b^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C_0}{\partial a^{(L)}} \quad (17)$$

Soll nicht nur eine Aufnahme aus dem Trainingsdatensatz in Betracht gezogen werden, wird der Durchschnitt für alle n Messungen im Datensatz berechnet:

$$\frac{\partial C}{\partial w^{(L)}} = \frac{1}{n} \sum_{k=0}^{n-1} \frac{\partial C_k}{\partial w^{(L)}} \quad (18)$$

Es scheint eine deutliche Vereinfachung zu sein, dass das Beispiel nur ein einzelnes Neuron in jedem Layer beschreibt. Für mehrere Neuronen in einem Layer erhöht sich die Komplexität mit der Einführung von j und k Indizes. Dadurch, dass alle Neuronen des $L - 1$ Layer mit allen Neuronen von dem L -ten Layer verbunden sind, hängt das einzelne Neuron nun von der Summe aller Aktivierungen der Neuronen des vorherigen Layers ab. Die Neuronen im $L - 1$ Layer werden mit k indexiert und die Neuronen im L Layer mit j :

$$\frac{\partial C_0}{\partial w_{jk}^{(L)}} = \frac{\partial z_j^{(L)}}{\partial w_{jk}^{(L)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}} \quad (19)$$

Die Berechnung für die Änderung der Kostenfunktion durch Änderung der Aktivierung aus dem $L - 1$ Layer, wird zu einer Summe aller $n_L - 1$ Aktivierungen.

$$\frac{\partial C_0}{\partial a^{(L-1)}} = \sum_{j=0}^{n_L-1} \frac{\partial z_j^{(L)}}{\partial a_k^{(L-1)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}} \quad (20)$$

2.7 Stochastic Gradient Descent

Die Motivation hinter der stochastischen Version des Gradient Descent Verfahrens (SGD) ist es, die Anzahl an Rechenschritten zu minimieren. Anstatt aus dem Datensatz für jedes Element (für den ganzen Trainingssatz) den Gradient der Kostenfunktion zu berechnen und anhand dessen Durchschnitts den nächsten Satz an Parametern zu ermitteln, wird stochastisch ein Element oder ein Teil der Elemente im Trainingsset bestimmt, welches für den Optimierungsschritt relevant wird. Durch die Einführung von weiteren Parametern und durch Beobachtung der Tendenz, wie

sich die Änderungsrate der Gewichte zwischen den Annäherungsschritten entwickelt, kann das Verfahren weiter optimiert werden. Durch Einführung vom Momentum kann die Learning Rate dynamisch für jeden Schritt bestimmt werden. Diese weiterführenden Techniken bauen auf das SGD Verfahren als Grundlage auf, wie das adam [19], adadelata [22], adagrad [24], nadam [23] etc.

2.8 Overfitting und Cross Validation

Das Ziel ist ein generalisiertes Modell zu entwickeln, welches die Trainingsdaten nicht auswendig lernt. Es soll den Zusammenhang zwischen Eingang und Ausgang entwickeln damit neue Daten, die den gleichen Prozess wie die Trainingsdaten darstellen, erfolgreich erkannt und vorhergesagt werden können. Dieses Auswendiglernen wird in der Literatur Overfitting genannt. Um diesem entgegenzuwirken, kann eine Methode wie die K Fold Cross Validation verwendet werden.

Es sei beispielsweise ein binäres Klassifikationsproblem gegeben, wo Spinatknödel von Speckknödeln unterschieden werden müssen. Der Datensatz besteht aus n Knödel Bilder in welchen immer ein Knödel ersichtlich ist. Zusätzlich liegt die Information vor, was für ein Knödel es ist. Das Modell wird k mal (Abbildung 2-8 zeigt ein Beispiel mit $k = 3$) bei erneuter Selektion der Trainings- und Testdaten gefittet, anschließend wird das Modell an den Testdaten, die während dem Training nicht verwendet wurden, getestet.

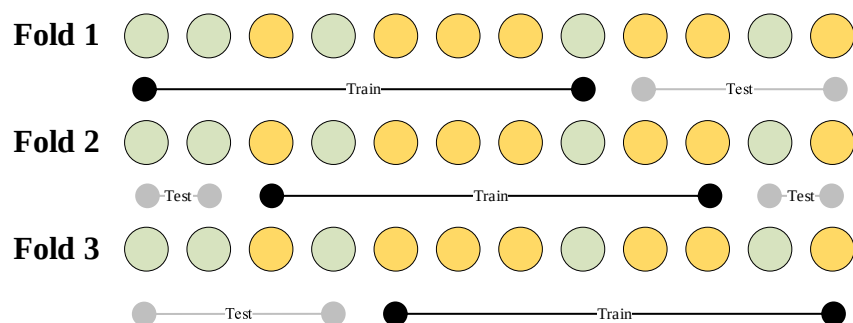


Abbildung 2-8: Cross Validation mit K-Fold

Durch die mehrmalige zufällige Partitionierung der Daten (K Fold Cross Validation) und durch die mehrmalige Wiederholung des Vorgangs kann ein Durchschnitt der Testgenauigkeiten bestimmt werden. Dies stellt die tatsächliche Performance des Netzes besser dar. Dieses Verfahren eignet sich ideal für die Modellselektion und Performance-Estimation bei kleineren Datensätzen [32]. Erweiterungen dieses Verfahrens führen auch eine randomisierte Mischung der Daten zwischen den Folds durch [25]. Weitere Versionen, wie der Stratified K Fold, sorgen dafür, dass in jeder Partitionierung die Anzahl der unterschiedlichen Klassen gleich ist. Diese Eigenschaft ist nützlich bei unbalancierten Datensätzen, wenn eine oder mehrere Klassen über wesentlich weniger Beispieldaten verfügen wie die anderen Klassen des Datensatzes.

2.9 Konvolution

Konvolution ist ein assoziatives Verfahren. Durch die Operation werden die räumlichen und zeitlichen Abhängigkeiten in einem Datensatz beibehalten. Dadurch eignet sich dies ideal für die Bearbeitung von Bild [31] und Time Series. Man kann Sensordaten bestehend aus mehreren Kanälen, wie ein zweidimensionales Grayscale Bild wahrnehmen. Die Überlegung diese Daten so zu behandeln, als wären sie in der Image Domain ist in [29] gezeigt. Durch Anwendung von Konvolution ist es möglich, die Anzahl der zu optimierenden Parameter so zu vermindern, dass keine signifikanten qualitativen Verluste im Ergebnis erzielt werden. Das Beispiel in Abbildung 2-9 bezieht sich in diesem Fall nicht auf ein Bild sondern auf Sensordaten, die mit der Breite b jeweils die Daten von einem anderen Sensor beinhalten, die mit der gleichen Sampling Rate in der Länge t vorliegen.

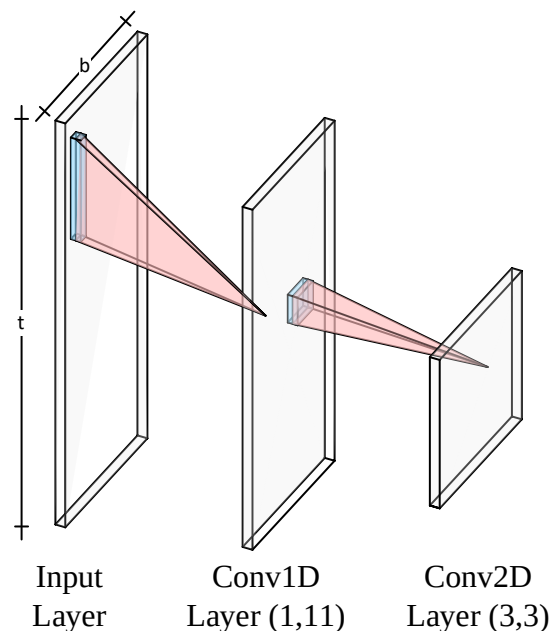


Abbildung 2-9: Convolution Visualisierung

Abbildung 2-9 zeigt zwei Convolutional Layer. In der ersten Schicht wird mit einem länglichen eindimensionalen Kernel (1x11) der Datensatz entlang der Zeitachse (t) abgetastet. Für diesen numerischen Fall kann die Faltung wie (21) es zeigt durchgeführt werden [40].

$$(f * k)(i) = \sum_{j=0}^{m-1} k(j) \cdot f(i - j) \quad (21)$$

Die Anwendung auf ein konkretes Beispiel ergibt sich aus Abbildung 2-10. Die Funktion f liegt mit den numerischen Werten in einem Array vor. Der Kernel k mit den Filterwerten entspricht ebenfalls einem Array. Dieses wird auf dem Element mit Index $i=2$ positioniert.

$$f = \begin{bmatrix} 2 & 5 & 4 & 4 & 3 & 5 \end{bmatrix}$$

$$k = \begin{bmatrix} 0.4 & 0.5 & 0.5 \end{bmatrix}$$

Abbildung 2-10: Convolution 1D Array

Aus dem Berechnungsschritt dieses Beispiels folgt das Ergebnis:

$$4 \cdot 0.4 + 4 \cdot 0.5 + 5 \cdot 0.5 = 8.15 \quad (22)$$

Am Rand (des Arrays) kann wahlweise Padding oder die Reduktion der länglichen Dimension als Lösung gewählt werden. Es ist möglich Kernel einzusetzen welche nicht die benachbarten Elemente, sondern in m und n Richtung, auch versetzte Indizes beinhalten. Diese Art an Convolutional Kernel verwenden die Dilation, wie es in der Abbildung 2-11 III. ersichtlich ist. Die Rate wie viele Einheiten des Kernels zwischen zwei Schritten weitergeschoben werden, kann ebenfalls bestimmt werden [27]. Die Strided Convolution mit verschiedenen Verschiebungen ist in Abbildung 2-11 I. und II. ersichtlich [28].

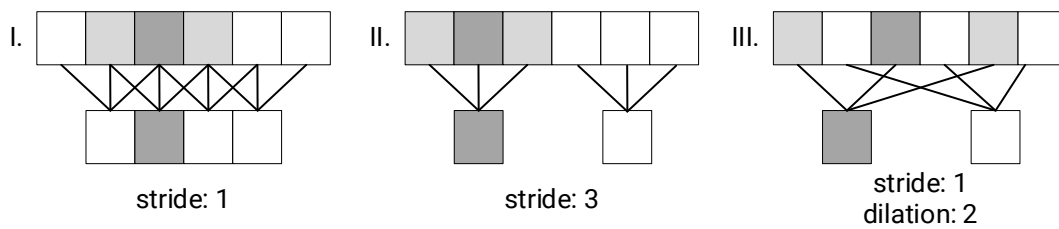


Abbildung 2-11: Convolution – Stride und Dilation

Wie bei dem Fully Connected Layer besitzen die Verbindungen des Convolutional Layers - die durch das Kernel bestimmt werden - ebenfalls Gewichte und Bias. Die mathematische Beschreibung, den Darstellungen (NN Abbildung 2-4 , CNN Abbildung 2-11) entsprechend, ändert sich dadurch, dass nur mehr die vorherigen Aktivierungen beachtet werden müssen, die von den Kernel Iteratoren m und n erfasst werden wie es in (23) gezeigt wird. An dieser Stelle ist auch die Reduktion an notwendigen Optimierungsparametern im Vergleich zum Fully Connected Layer ersichtlich. Bei der Abbildung 2-9 in der zweiten Schicht wird ein symmetrischer 3x3 2D Kernel eingesetzt. Die Iteratoren für m, und n sollen in diesem Fall entsprechend zwischen 0 und 2 erfolgen. Die generalisierte Form gilt für den Fall, in welchem der Datensatz einen Kanal aufweist (ähnlich wie bei der Anwendung für Grayscale Bild und Sensordaten):

$$x_{(i,j)}^{(L)} = f \left(\sum_m \sum_n w_{(m,n)}^{(L)} a_{(i+m,j+n)}^{(L-1)} + b^L \right) \quad (23)$$

2.10 Aktivierungsfunktionen

Diese Funktionen sind dafür verantwortlich, dass die notwendigen Nicht-Linearitäten wie in 2.2 beschrieben, in das Netz gebracht werden. Die vorgestellten Verfahren SGD und Backpropagation stellen die Anforderung an diese Funktion, dass sie differenzierbar sein muss. Eine weitere relevante Eigenschaft von diesen Funktionen ist der Wertebereich, den sie am Output erzeugen.

Wie in [41] beschrieben, eignen sich sigmoide Aktivierungsfunktionen als letzter Layer bei binären Klassifikationsproblemen, folgend aus dem Wertebereich 0 bis 1.

$$\phi(z) = \frac{1}{1 + e^{-z}} \quad (24)$$

Abbildung 2-12 zeigt einen ReLU (Rectified Linear Unit). Diese Funktion erlaubt einen Ausgang in den positiven unendlichen Wertebereich wachsen zu lassen und rektifiziert alle Werte kleiner null (25). Vorgestellt wurde die Funktion durch Nair und Hinton 2010 und die Anwendung zwischen den Hidden Layer eines Netzes wird von Glorot [16] und Zhang [38] empfohlen. Dadurch, dass die Berechnung im Vergleich zu tanh oder sigmoid Funktionen mit geringem Rechenaufwand erfolgen kann, wird der Lernprozess beschleunigt.

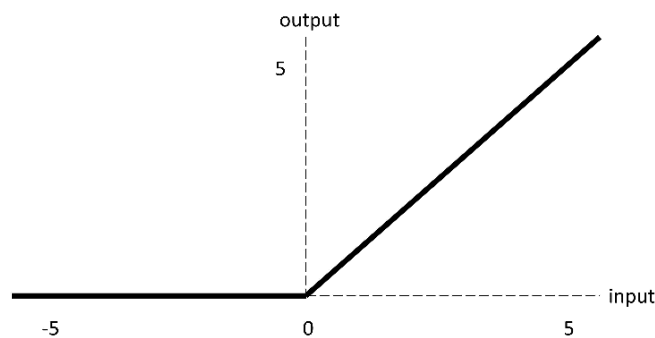


Abbildung 2-12: Rectified Linear Unit – ReLU [42]

$$f_{relu}(z) = \max(0, z)$$

$$f_{relu}(z) = \begin{cases} 0, & \text{if } z < 0 \\ x, & \text{if } x \geq 0 \end{cases} \quad (25)$$

Soll ein nicht normalisierter Ausgang eines neuronalen Netzwerkes zu einer Wahrscheinlichkeitsverteilung über den vorhergesagten Klassen gemappt werden, kann die Softmax (26) Funktion verwendet werden.

$$\phi_{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=0}^{k-1} e^{z_j}} \quad (26)$$

2.11 Max Pooling

Max Pooling ist ein Downsampling Layer [36]. Dieser kann gleichzeitig auf mehrere Dimensionen angewandt werden. Abbildung 2-13 zeigt den Effekt an einem numerischen Beispiel. Bei I. wird ein Kernel mit der Größe 2x2 angewendet. Von den im Kernel erfassten Elementen wird jeweils der größte selektiert. Die Anwendung auf eine Dimension mit einem 1x2 Filter ist in II. ersichtlich.

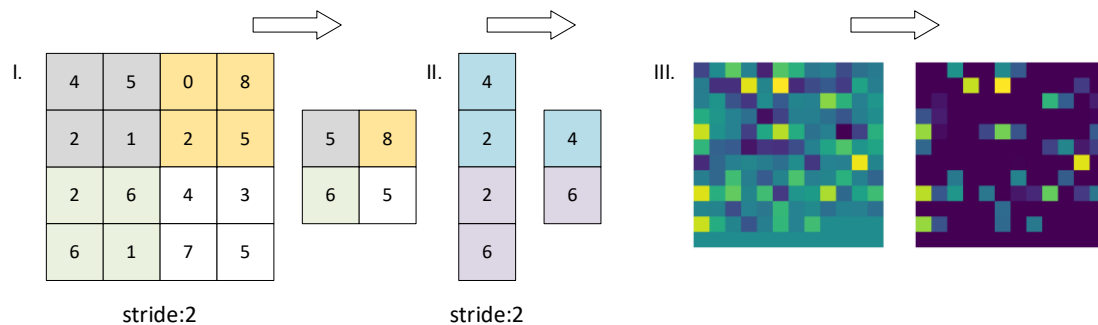


Abbildung 2-13: Max Pooling

Max Pooling wird häufig nach dem Convolutional Layer und nach der Aktivierung angewandt. Dieser Effekt zeigt an einem Beispiel eines Netzes dieser Arbeit, ist bei der Abbildung 2-13 als III. inkludiert.

2.12 Dropout

Ein Dropout Layer ist eine weitere Maßnahme gegen Over Fitting [33]. Die Idee ist, stochastisch Neuronen aus dem System zu eliminieren wie in der Abbildung 2-14 ersichtlich und damit zu hohe Co-Adaptation zu vermeiden. Die durchgestrichenen Einheiten sind eliminiert worden und bauen keine Verbindungen auf.

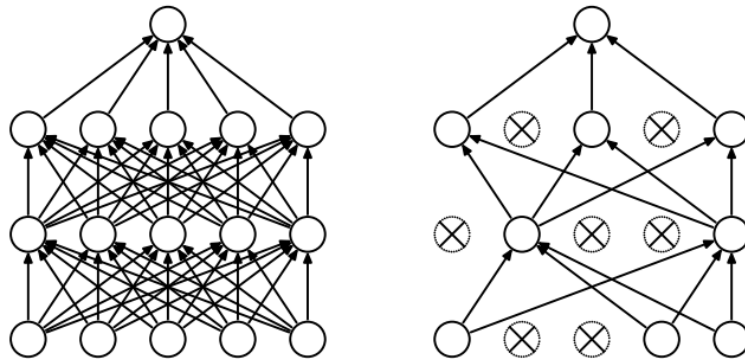


Abbildung 2-14: Links – Standard Neural Netzwerk, Rechts – Dropout Neural Netzwerk [33]

Dies erhöht den Grad der erfolgreichen Generalisierung und führt zu verbesserten Ergebnissen, auch in Kombination mit Convolutional und Max Pooling Layer. [33]

2.13 Fitness

Die Optimierung erfolgt zwar durch die Minimierung der Fehlerfunktion, das eigentliche Ziel ist aber meistens ein Modell für eine andere Metrik optimal auszulegen. Abbildung 2-15 zeigt den Trainingsverlauf über die Epochs mit sinkender Fehlerrate und steigender Genauigkeit.

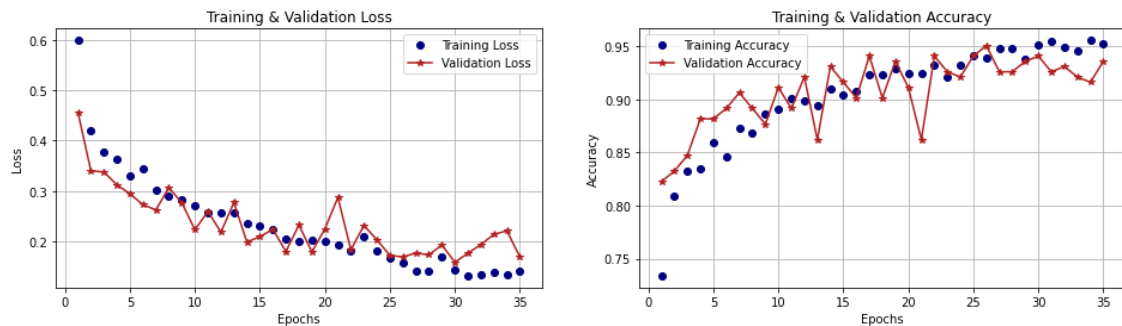


Abbildung 2-15: Iterationen - Epochs

Die Prädiktionen bei einem Klassifikationsproblem mit beispielsweise drei Klassen, A, B und C können in einer Confusion Matrix dargestellt werden, wie Abbildung 2-16 zeigt. Für die Klasse A erfolgten in diesem Beispiel insgesamt 17 Hervorsagen, daraus waren 12 richtig (TP) und 5 sind fälschlicherweise der Klasse C zugeordnet worden (FN). Es gab auch drei Prädiktionen, die zu der Klasse B gehört hätten, aber A zugeordnet worden sind (FP). Die weiteren 68 Testfälle, welche für die restlichen zwei Klassen richtig erraten worden sind, werden mit (TN) gekennzeichnet.

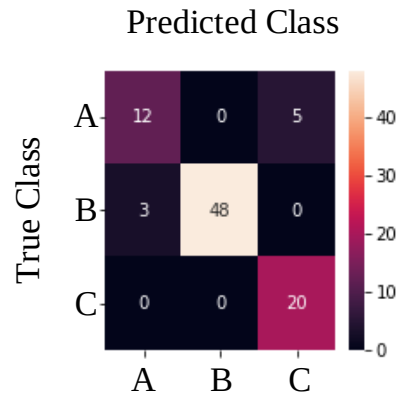


Abbildung 2-16: Confusion Matrix

Aus diesen Teilmengen (TP – True Positive, TN – True Negative, FP – False Positive, FN – False Negative) können unter anderen die folgenden Metriken erstellt werden:

$$Precision = \frac{TP}{TP + FP} \quad (27)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (28)$$

$$Recall = \frac{TP}{TP + FN} \quad (29)$$

$$F1\ Score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \quad (30)$$

Falls für die Anwendung nur wichtig ist, dass möglichst viele Ergebnisse richtig erraten werden, ist die Genauigkeit (28) die optimale Metrik, denn ein geringer Anteil an falschen Vorhersagen beeinflusst das Ergebnis nicht wesentlich.

Sind hingegen bei der Anwendung falsche Vorhersagen kritisch, dient der F1-Score als optimale Metrik, denn dieser bestraft falsche Vorhersagen mehr.

Der F1-Score ist das harmonische Mittel zwischen Recall und Präzision. Bei unbalancierten Klassen kann die Genauigkeit ein sehr gutes Ergebnis zeigen, auch wenn eine der unterrepräsentierten Klassen völlig falsch identifiziert wurde. Um einen Überblick über die tatsächliche Leistung des Netzes auch bei solchen Klassen zu erhalten, werden die Metriken Recall (29) (Sensitivität), Precision (27) und F1-Score (30) je Klasse ausgewertet.

3 Entwicklung eines modularen Messsystems

Das Ziel war es, ein Messsystem zu bauen, welches in der Lage ist, digitale und analoge Sensoren abzufragen und dessen Daten aufzuzeichnen. Mit der Intention die Daten für ein Machine Learning Problem zu sammeln, muss dieses System in der Lage sein, eine Klassifizierung ebenfalls von einem menschlichen Anwender, nach Abschluss der Messung, abzufragen. Hierzu wurde am Anfang des Projektes festgelegt, dass keine externe Anwendung (beispielsweise Smartphone-App) für diese Aufgabe zuständig sein soll, sondern die Funktionalität direkt am Gerät vorhanden sein muss.

Dadurch, dass der Einsatz von digitalen Sensoren geplant war, welche örtliche Messungen durchführen, war es nicht möglich, alles auf einer Leiterplatte zu integrieren. Anstatt dieser, wurde ein modulares System geplant, welches eine flexible Platzierung der Sensormodule erlaubt. Hierzu war eine notwendige Anforderung, die Verkabelung günstig und mit einfachen Mitteln realisieren zu können, ohne externe Arbeitsleistungen und speziellen Kabelkonfektionierungen.

Es wurden nur JST-, RJ45- und Lötanschlüsse verwendet. Ein weiteres Ziel war, das System unter erhöhten Temperaturbedingungen betreiben zu können. Auch dies stellt besondere Anforderungen an die Verkabelung. Herkömmliche im Netzbereich verwendete UTP Kabel besitzen meist eine maximale erlaubte Betriebstemperatur von 60°C. Die ausgewählten hochflexiblen silikonisolierten 8 x AWG 28 Steuerleitungen (Si-SL-O) erlauben eine maximale Betriebstemperatur von 150°C. Der Hersteller Metrofunk bietet auch hitzebeständige, mit Teflon isolierte Leitungen an, welche für die weitere mehradrigen Verkabelungen verwendet worden sind.

Digitale Sensoren verfügen meistens über Schnittstellen, welche für die Kommunikation innerhalb einer Leiterplatte, Backbone vorgesehen sind. Es wurde eine Lösung gesucht, um die störungssichere Funktionalität für mehrere flexibel platzierbare digitale Sensoren zu ermöglichen.

Eine weitere Voraussetzung war, alle PCBs des Messsystems händisch bestücken zu können, dies führte zur Verwendung von 0805 SMD Bauteilen und die Vermeidung von BGAs.

Am Beginn des Projektes war noch nicht klar, in welcher Form die Realisierung des Neuralen Netzes durchgeführt wird. Es wurde angestrebt mehrere Schnittstellen zur Außenwelt zu ermöglichen.

Für das Messsystem sind insgesamt 19 unterschiedliche PCBs entwickelt worden.

- **Mainboard**
- **Spannungsversorgung und Leistungselektronik Modul**
- **Rohr Sensor Modul**
- **Analog Modul**
- **IR Temperatur Sensor Modul**
- **Farbsensor Modul**
- **LED Beleuchtungsmodul für den Farbsensor**
- **Wägemesszelle 24bit ADC Modul**
- **Encoder Modul**
- **Display Modul**
- **Kapazität zu Frequenz Konverter Modul**
- **Licht zu Frequenz Konverter Modul**
- Licht zu Analoge Spannung Konverter Modul
- **ESP32 USB Programmer**
- UART zu RS485 Aufsteckmodul
- RS485 zu USB-Konverter Modul
- I2CD Bus T Splitter Modul
- **Aufsteckadapter Mainboard** für Frequenz Eingänge für den UART und **IO Port**

Die fett markierten Komponenten werden im finalen Toaster Messsystem verwendet und in dieser Arbeit näher erklärt. Die Vorevaluierung und Auswahl der Sensoren und Komponenten, sowie die Schritte der Prototypisierungsphase werden nur stellenweise erwähnt, hauptsächlich wird nur der Endzustand des Systems vorgestellt.

Die Module des Systems verwenden die folgenden Größen:

- $32mm \times 32mm$ – Alle Module mit RJ45 Anschluss
- $16mm \times 16mm$ – Konverter Module
- $96mm \times 24mm$ – LED-Beleuchtung
- $96mm \times 96mm$ - Mainboard
- $64mm \times 64mm$ – Versorgungsmodul

3.1 Mainboard

Als Herz des Systems wurde der ESP32-WROOM-32U SOC Module von Espressif gewählt. Dieses SOC bietet einen Dual Core CPU, 520 KiB SRAM und 128Mbit Flash Speicher. Neben klassischen Peripherien wie I2C, SPI Master, PWM Motor Controller, UART etc. bietet der Controller eine integrierte Wi-Fi und Bluetooth Peripherie. Eine externe ISM Band Antenne kann über den U.FL Konnektor des Moduls angebunden werden. Abbildung 3-1 zeigt beide Seiten des Mainboards mit allen Anschlüssen und Komponenten, außer den Pin-Headern. Diese Darstellung wird bei jedem näher erklärten Modul verwendet und dient zu besseren visuellen Vorstellung des Beschriebenen.

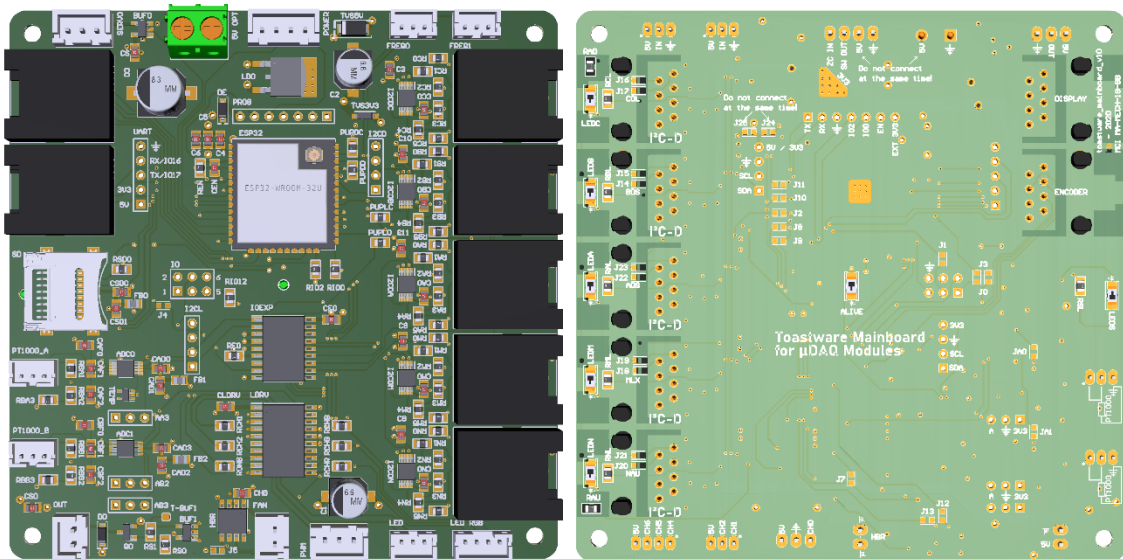


Abbildung 3-1: Mainboard

Abbildung 3-2 dient als Systemübersicht und zeigt das Messsystem aus dem Blickwinkel des Mikrocontrollers. Es sind die Besetzung der Peripherien und die Anbindung der Komponenten und Module an dem Mainboard zu sehen.

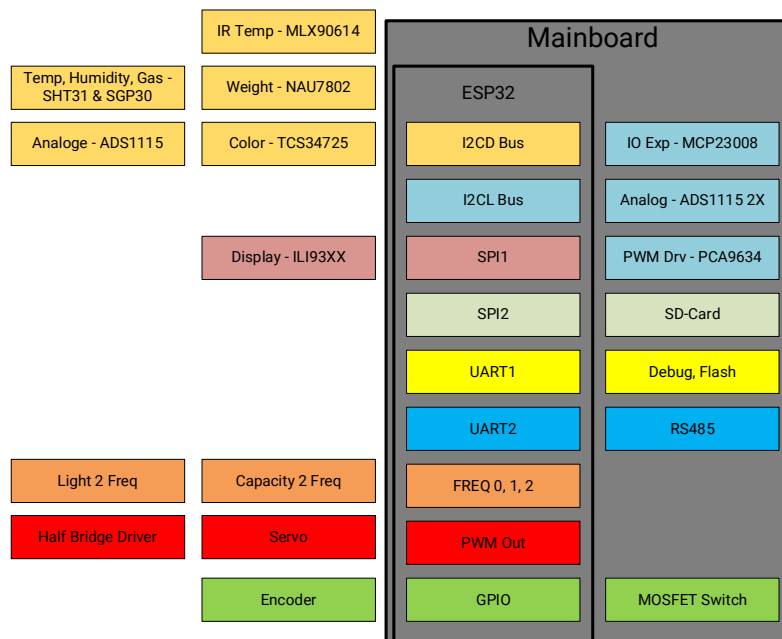


Abbildung 3-2: Übersicht aus dem Blickwinkel des Mikrocontrollers

Die Aufzeichnung der Daten wurde durch eine SD-Karte realisiert. Festgelegt war eine Sampling Rate von 2Hz gewährleisten zu können, angestrebt wurde das Messsystem so auszulegen, dass es mit 10Hz sampeln kann. Dies führt bei 25 Kanälen bei 4 Byte Variablen zu einer Datenmenge von 1kB/s. Da keine besonders großen Datenraten benötigt werden und die SDIO Schnittstelle auch mehr Pins am Controller benötigt, wurde die SPI kompatible Implementierung gewählt.

3.1.1 Spannungsversorgung

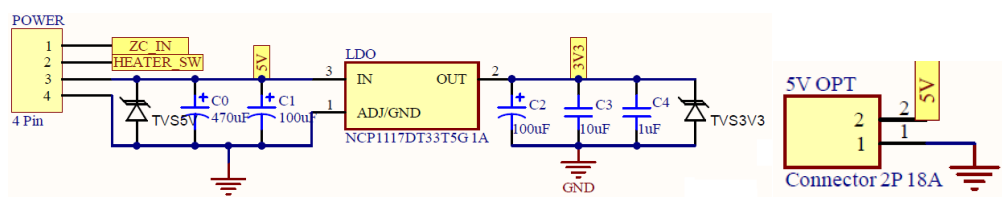


Abbildung 3-3: Spannungsversorgung Mainboard

Das Mainboard wird über einen JST-XH Stecker mit 5V versorgt und stellt mit Hilfe eines LDOs (NCP1117), 3.3V zur Verfügung. Über den gleichen Port (POWER) wird das Zero Crossing Detektor Signal zum Controller geführt, sowie ein geschalteter Ausgang des Controllers zurück zum Versorgungsmodul (und Leistungselektronik). Alle externen Module werden mit der 5V Versorgungsspannung gespeist und jedes Modul soll sich die notwendigen Spannungen lokal zur Verfügung stellen. Dies geschieht meistens mit einem weiteren LDO (AP2112, AP7315) oder bei den angeschlossenen Konverter Modulen mittels einer Spannungsreferenz (NCP51460).

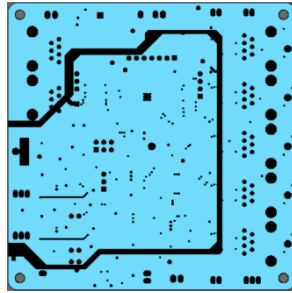


Abbildung 3-4: Spannungsversorgung - Planes

Wie Abbildung 3-4 zeigt, wird einer der innenliegenden Layer in zwei Spannung Planes geteilt. Außen liegt 5V und in der Mitte des Boards 3.3V Spannung an. Ein weiterer Layer wird als Ground Plane verwendet. Die zwei verbleibenden Layer des 4 Layer PCBs werden als Signal Layer genutzt. Der Controller kann über den Programmier-Stecker Abbildung 3-5 auch direkt mit 3.3V versorgt werden, die Betriebsspannung des ESP32 Moduls von 2.3 V bis 3.6 V [8] wird trotz Spannungsabfall auf der Schottky Diode DE erreicht und eine Rückspeisung wird verhindert.

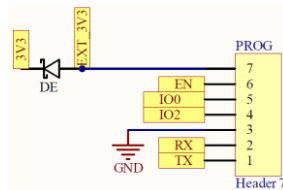


Abbildung 3-5: Programmierschnittstelle

Das Board kann direkt mit 5V über den OPT Anschluss versorgt werden, an welchem auch externe 5V Verbraucher angeschlossen werden können. Um das System vor Überspannung und Spannungsspitzen zu schützen, werden für beide Boardspannungen TVS Dioden eingesetzt.

3.1.2 I2C

Das Mainboard verfügt über zwei I2C Busse. Eines verbindet nur Komponenten auf dem Mainboard, dieser Bus wird folgend I2CL (Local) genannt. Auf dem I2CL Bus werden die Komponenten nicht voneinander isoliert und jeder Teilnehmer ist immer verbunden. Von allen Teilnehmern wird die maximale zulässige Busgeschwindigkeit genutzt (1MHz).

Der zweite I2C Bus verbindet externe digitale Sensoren über Kabel zum Mainboard, dieser Bus wird folgend I2CD (Distributed oder Differential) genannt. Das Mainboard verfügt über 5 I2CD Ports die einzeln aktivierbar sind. Jeder Port endet in einer RJ45 Buchse. Bevor die SCL und SDA Signale das Board verlassen, werden diese mithilfe von PCA9615 Bausteinen in differentielle Form transformiert. Diese sind „2-channel multipoint Fast-mode Plus differential I2C-bus buffer with hot-swap logic“[68]. Abbildung 3-6 zeigt den Schaltplan von einem dieser Ports. Der differentielle Buffer ist transparent für die Teilnehmer der Kommunikation und erkennt automatisch die

Richtung. Die I2CD Signale werden zwischen dem Port und dem Baustein differentiell geroutet und zwischen zwei RJ45 Ports werden sie über Twisted Pairs transferiert. Durch den EN Pin des ICs werden die Ports aktiviert, zusätzlich wird die Bus Aktivität mit einer Status LED veranschaulicht, welche beim Ein- und Ausschalten der Ports mitgeschaltet wird.

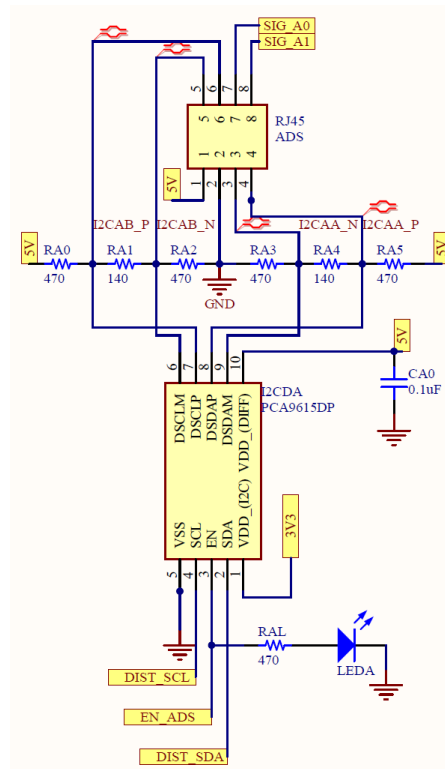


Abbildung 3-6: I2CD Port Schematik

Je Port werden zwei weitere Signalleitungen definiert, welche für die Interrupt / Alarm Signale der verbundenen I2CD Sensormodule zur Verfügung stehen. Jedes Signal ist einzeln mit den Jumpern J14-J23 verbindbar. Für die Signalleitungen sind je nach Anwendungsfall die Pull Up and Pull Down Widerstände bestückbar.

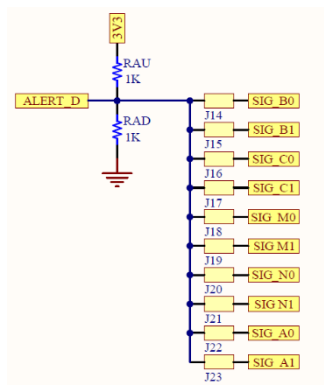


Abbildung 3-7: I2CD Signal Selektion

Die Alarm / Interrupt Signale der I2CL Teilnehmer verfügen über die gleichen Optionen wie die von I2CD. Beide I2C Busse haben auf dem Board noch einen zusätzlichen Pin-Header, um Aufsteckmodule direkt am Board als zusätzliche Erweiterung verwenden zu können.

3.1.3 GPIO Expansion

Der ESP32 verfügt über 39 GPIO Pins, wovon 5 verwendet werden, um den QSPI Flash im SOC Module anzubinden. Es verbleiben daher nur mehr 33 für die Anwendung. Dies wird am Board mithilfe des MCP23008 IO Expander gelöst, welcher hauptsächlich dafür verantwortlich ist, dass die einzelnen I2CD Ports aktiviert und deaktiviert werden können bzw. schaltet diese IC zusätzlich die Display Hintergrundbeleuchtung und das Display Reset. Weitere Pin-Erweiterungen werden durch den PCA9634D (8-bit Fm+ I2C-bus LED driver) ermöglicht, welcher für die Ansteuerung von Status Leds und PWM - steuerbarer Einheiten verwendet werden kann. Dieses IC bietet auch programmierbare Intervall Timer. Dadurch lässt sich eine Pulsfolge, beispielsweise ein Blinkzyklus mit einmaliger Konfiguration, ohne weitere Interaktion mit dem IC realisieren. Beide Expansion ICs sind an dem I2CL Bus gebunden.

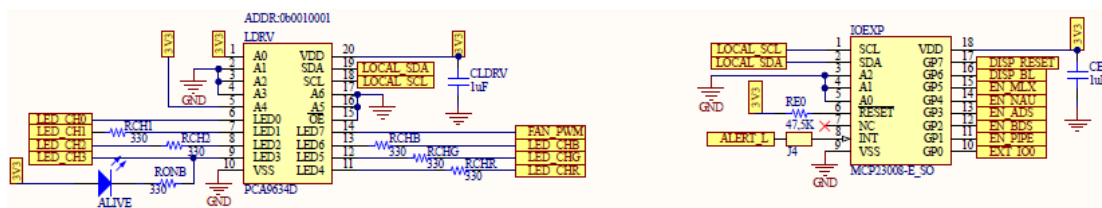


Abbildung 3-8: GPIO und PWM Expansion

3.1.4 Ansteuerung von Aktoren

Das Mainboard verfügt über eine Halbbrücken IC (BD6211F). Dadurch kann beispielsweise ein Lüfter oder ein DC Motor betrieben werden.

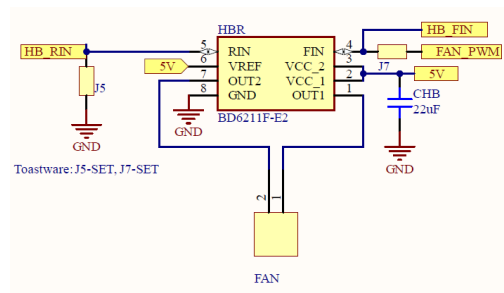


Abbildung 3-9: Halbbrücke Schematik

Die verwendeten Pins um den IC anzusteuern, können über Jumper (J5, J7 in der Abbildung) umgerootet werden.

Abbildung 3-10 zeigt einen weiteren Ausgang am Mainboard. Es wird ein nicht invertierender Buffer verwendet (NL17SZ125) um den 3.3V Ausgang des ESP32 zu einem 5V Logik Ausgang zu konvertieren. Der Ausgang des Buffers wird auf ein Pad geführt, hier kann eine Lötverbindung erfolgen oder das Signal des Buffers kann dazu verwendet werden um den N-Kanal-Leistungsmosfet Q0 von Infineon anzusteuern.

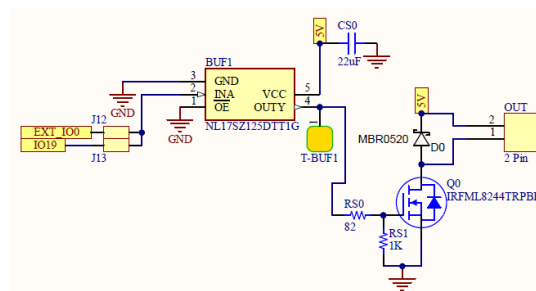


Abbildung 3-10: Ausgang mit N Kanal MOSFET

Für die Ansteuerung und Versorgung von Servos wurde ebenfalls ein Ausgang vorgesehen. Hier wird ebenfalls ein nicht invertierender Buffer als Voltage Level Translator verwendet, da ältere Servo Modelle oft 5V Logik Spannung benötigen.

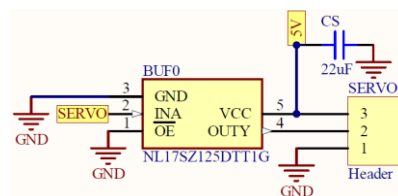


Abbildung 3-11: Halbbrücke Schematik

3.1.5 ADCs

Auf dem I2CL sind zwei ADC ICs (ADS1115) angebunden. Neben der Einsparung an Pins am Controller bieten sie eine bessere Genauigkeit (16bit vs. 12bit) als die im SOC integrierte Peripherie.

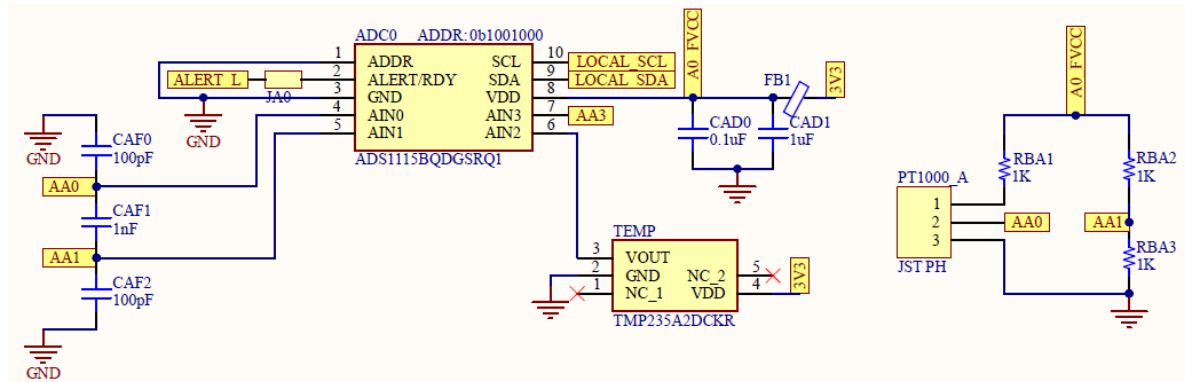


Abbildung 3-12: Analog Schnittstelle für drei Leiter PT1000 Messung

Zwei Anschlüsse beider ADCs, wie bei dem ersten IC AIN0 und AIN1, sind für die Messung in Wheatstone'sche Messbrücken Konfiguration mit 3 Leiter PT1000 Elementen ausgelegt. Am ersten ADC ist zusätzlich ein analoger Temperatursensor (TMP235) angebunden, um eine Temperaturmessung des Boards zu ermöglichen. Die restlichen Anschlüsse der beiden ADCs sind zum Pin-Header geführt.

3.1.6 Schnittstellen

Neben dem Programmierstecker und den I2CD und I2CL Aufsteckplätzen, kann das Board über den UART Pin-Header mit einem RS232 oder RS485 Aufsteckmodul erweitert werden. Weitere Anschlüsse sind in RJ45 Ausführung für einen Drehencoder und ein Display mit unidirektionaler SPI Kommunikation vorhanden. Auf dem Board sind noch zwei JST-PH Ports vorgesehen, die jeweils eine Spannungsversorgung und einen Frequenz Eingang Pin beinhalten. Durch den Pin-Header IO kann die Anzahl von solchen Eingängen auf 4 erweitert werden.

Dieses Board hat, neben der Verwendung als Messsystem in einem Toaster und HMI für Klassifikationsinformation, zahlreiche andere Verwendungszwecke. Mit der Nutzung des Halbbrückentreibers und des Servo Ausgangs kann das Board zum Beispiel als Kern einer fahrenden ferngesteuerten Plattform dienen.

3.2 Versorgungsmodul

Diese Platine beinhaltet die folgenden Funktionen:

- Spannungsversorgung - AC/DC Konverter 5V Ausgang
- Zero Crossing Detektor mit galvanischer Trennung
- Schalter für 230V AC Lasten mit galvanischer Trennung

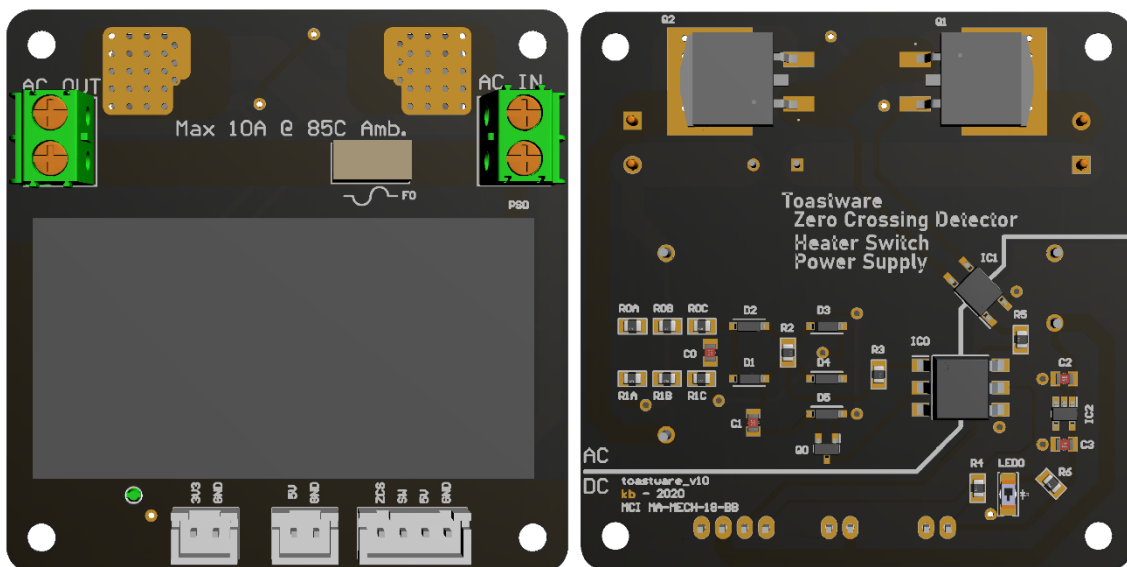
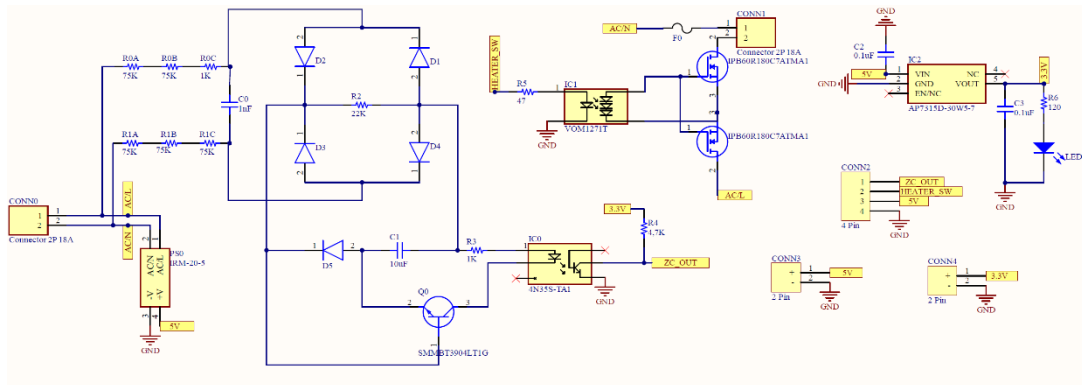


Abbildung 3-13: Versorgungsplatine Platzierung der Bauteile, Anschlüsse

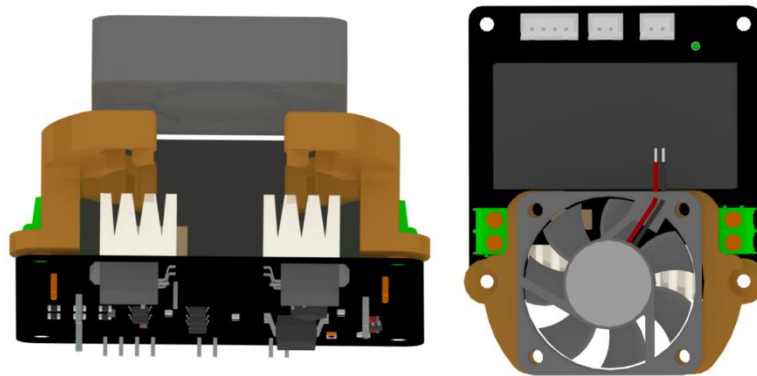
Für die AC/DC Wandlung wurde ein fertiges Modul (IRM-20-5) von Meanwell verwendet, denn dieses kann 20W Leistung bei 5V auszugeben. Dieses Modul wird Through Hole auf der PCB montiert. Die galvanisch getrennte Zero Crossing Detektion (ZCD) wird für die Bestimmung des idealen Ein- und Ausschaltzeitpunktes benötigt. Hier wird eine bestehende Schaltung verwendet. Die detaillierte Beschreibung der Schaltung ist unter [46] zu finden.

Um AC Lasten schalten zu können wurde ein SSR mit zwei gespiegelt zusammengeschalteten IPA60R060C7 von Infineon realisiert, Abbildung 3-14. Als Gate Treiber wurde der Photovoltaik Gate Treiber der VOM1271T verwendet, dieser integriert auch die fast turn off Elektronik im IC und ist in der Lage mit $t_{on} = 53 \mu s$ und $t_{off} = 24 \mu s$ zu schalten.

Die Leiterbahnbreite mit 200mil für den Hauptstrompfad wurde nach IPC-2221 ausgelegt. Bei einer Kupferdicke von 2 oz/ft² erwärmt sich die Leitung bei 10 A um maximal 25 °C. Für die Absicherung des Gerätes wird eine 10 A Sicherung auf dem Modul verlötet.



Durch den RDS_{on} von 0.06Ω bei einer Last von 5A beträgt der gesamte Leistungsverlust an beiden MOSFETs insgesamt 3W. Bei den maximal spezifizierten 10A sind dies insgesamt 12W. Um die Kühlung der MOSFETs zu gewährleisten, wurde an beiden Layern eine Kupferfläche vorgesehen die durch VIAs verbunden sind. An der Oberseite des Boards werden an diesen, unter Hochspannung stehenden Kupferflächen, Keramik-Kühlkörper montiert, um die Gefahr eines Stromschlages zu vermindern. Über den Kühlkörpern wird ein 5V Lüfter platziert, welcher an einer HAT-PLA Kunststoff Konstruktion mit einer M3 Mutter verschraubt wird.



Die PCB wird an einer Aluminiumplatte mit Abstandsbolzen verschraubt. Der Spalt zwischen den MOSFETs und der Aluminiumplatte wird mit Wärmeleitpads gefüllt, damit Wärme auch in die Aluminiumplatte geleitet werden kann.

3.3 Analog Digital Converter Modul

Das Analog Modul verwendet den gleichen IC – ADS1115 wie das Mainboard, aber in diesem Fall werden die Eingänge nicht in Brückenkonfiguration genutzt, sondern Single Ended. An diesem Modul können vier analoge Sensoren wie der LMT86LP Temperatursensor angeschlossen werden.

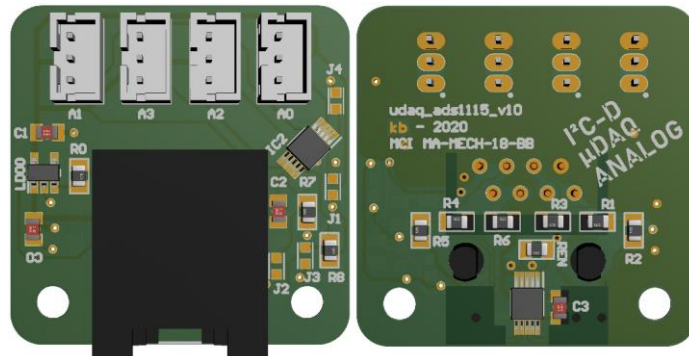


Abbildung 3-16: ADC Modul

Dieser IC kann mit 4 verschiedenen Adressen konfiguriert werden. Die Adressselektion wird bei diesem Modul ebenfalls durch Jumper ermöglicht.

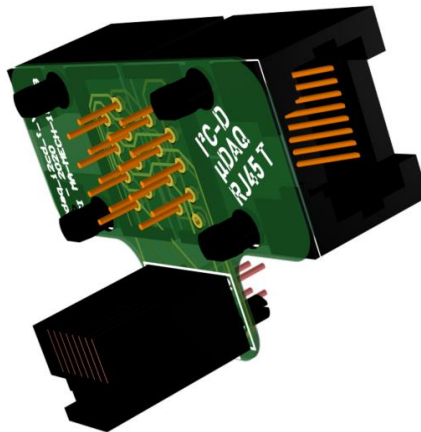


Abbildung 3-17: RJ45 T Splitter

Durch Verwendung des RJ45 T-Stückes Abbildung 3-17, welches ebenfalls für dieses Messsystem entwickelt wurde, können an einem I2CD Port maximal 4 solche Module angeschlossen werden. In diesem Fall sollen die Terminierungswiderstände nur beim letzten Modul gesetzt werden. Würde man an jeden I2CD Port des Mainboards vier verkettete analoge Module anschließen, könnten über 80 analoge Kanäle erzeugt werden. Die programmierbare Datenrate bei diesen Modulen liegt zwischen 8 und 860 Samples pro Sekunde, das integrierte PGA erlaubt einen Messbereich von ± 256 mV bis ± 6.144 V.

3.4 Wägemesszellen Modul

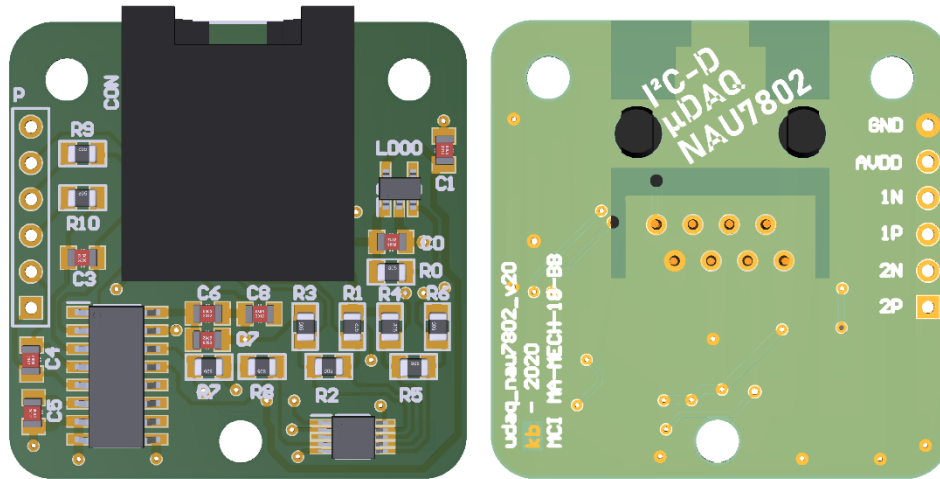


Abbildung 3-18: Wägemesszellen Modul

Der 24-Bit Dual-Channel ADC für Brückensensoren von Nuvoton wird ebenfalls als I2CD Modul gebaut. Diese PCB bietet die Möglichkeit, zwei Brückensensoren anzuschließen. Die Plattformwägezelle wurde aus einer Taschenwaage entnommen, welche einen Messbereich von 0 – 500 gr hat. An der Druck- und Zugseite der Plattform, sind jeweils ein bis zwei DMS aufgebracht, welche als Wheatstone'sche Vollbrücke verschaltet werden. Durch eine Messung konnte festgestellt werden, dass die Brücke aus 1k Ω Elementen besteht. Der NAU7802 verfügt ebenfalls über eine 400Khz I2C Schnittstelle und kann somit in das I2CD Bus Konzept integriert werden.

3.5 Rohr Sensor Modul

Hier kommen zwei Sensoren des Herstellers Sensirion zum Einsatz:

- SGP30 - Multi-Pixel Gas Sensor
- SHT31 – Temperatur und Luftfeuchtigkeitssensor

Der SGP30 verwendet die MOx Technologie und integriert 4 unabhängige Zellen in dem DFN Gehäuse. Die folgende Abbildung zeigt das Blockschaltbild von den Signalen und dem Aufbau des Sensors:

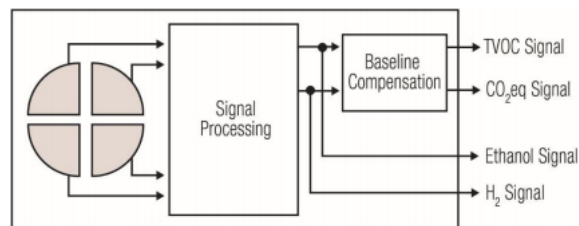


Abbildung 3-19: Gas Sensor Blockschaltbild

Der Sensor misst die Ethanol- und Wasserstoffkonzentration und kalkuliert aus den Werten eCO₂ und VOC Messungen. Für das interne Signal Processing des Gassensors kann die absolute Luftfeuchtigkeit zyklisch im Speicher des Gassensors abgelegt werden. Dies wird im Sensor intern zur Kompensation verwendet. Die absolute Luftfeuchtigkeit wird anhand der Werte des SHT31 Sensors bestimmt und im SGP30 Sensor bei jedem Messzyklus upgedatet. Der SGP30 muss mit 1.62 bis 1.98V versorgt werden und die Logik Spannung ist ebenfalls mit 1.8V angegeben.

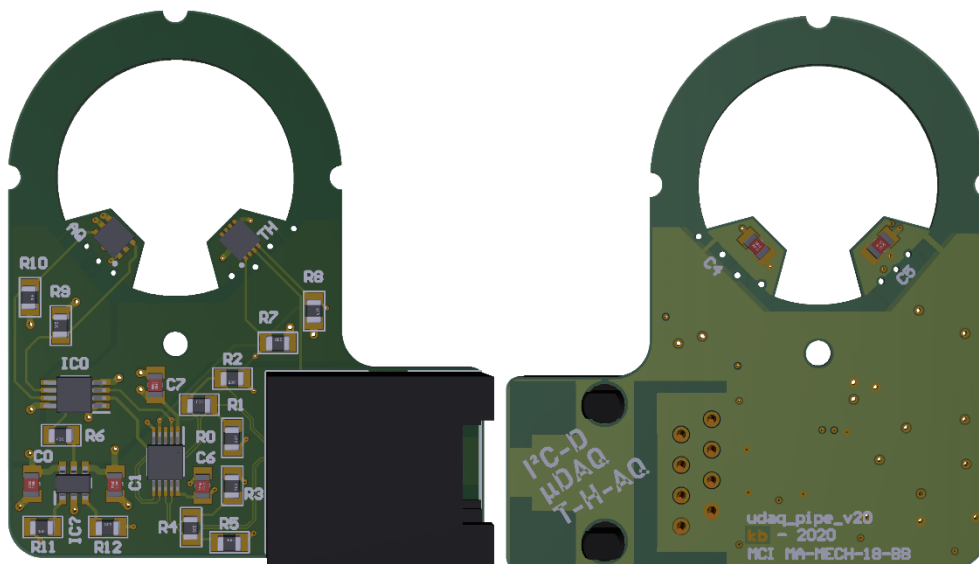


Abbildung 3-20: Rohrsensor Modul

3.6 Frequenz Konverter

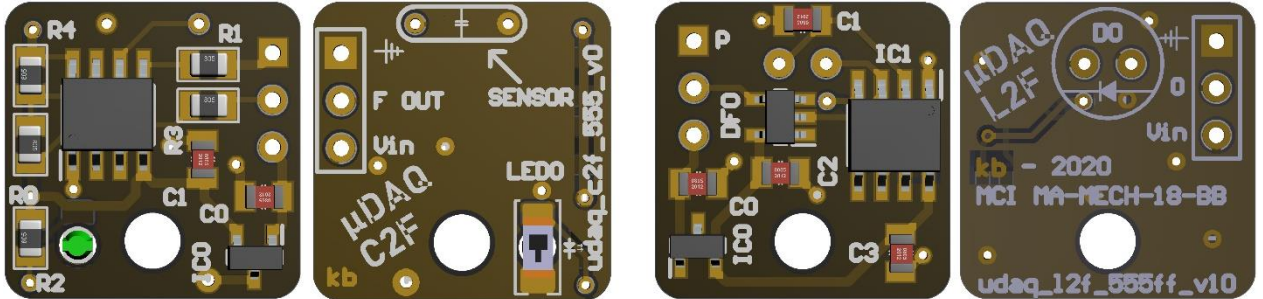


Abbildung 3-22: ADC Modul

Als Basis beider Schaltungen dient der TLC555 CMOS Timer IC welcher auch mit Frequenzen von bis zu 2Mhz betrieben werden kann.

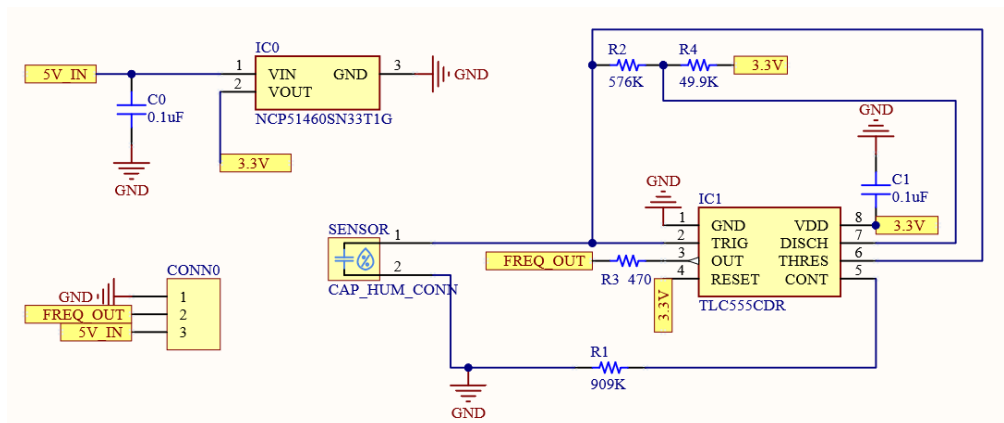
3.6.1 Kapazität zu Frequenz Konverter

Diese Schaltung Abbildung 3-23 verfolgt das typische astabile Design für den 555 Timer [64]. In diesem Fall wird die variable Timing Kapazität der Feuchtigkeitssensor sein, welcher die Ausgangsfrequenz bestimmt. Die Schaltung ist äquivalent zu [45] welche die Erweiterung der [65] Schaltung ist. Der Kondensator (in dem Fall der Sensor) wird zwischen 1/3 und 2/3 der VDD Spannung durch R2 und R4 aufgeladen und durch den DISCH Pin über R2 entladen. Der daraus erzeugte Duty Cycle hängt mit dem Verhältnis der beiden Widerstände zusammen. Das Control Signal über den R1, bewirkt die Verschiebung der durch den internen Spannungsteiler eingestellten Schwellwertspannungen und dient als Temperaturkompensation wie in [45] vorgeführt.

Mit ansteigender relativer Luftfeuchtigkeit, steigt die Kapazität des Sensors und die die erzeugte Frequenz sinkt. Der Konverter wird entfernt von direkten Hitzequellen im Gehäuse platziert und die kapazitiven Sensoren werden über geschirmte Leitungen angeschlossen. Der Schaltplan Abbildung 3-23 zeigt die Schaltung für den Sensor HS1101, welcher im finalen Aufbau nicht verwendet wurde. Jedoch wird die Schaltung mit zwei temperaturfesten Polymer Feuchtigkeitssensoren von BB-sensor bestückt und in unterschiedlichen Frequenzbereichen betrieben, die aus den verschiedenen nominalen Kapazitäten folgen. Die Auslegung und Prototypisierung werden nicht weiter angeführt, den Typ und die Parameter der Sensoren zeigt die Tabelle:

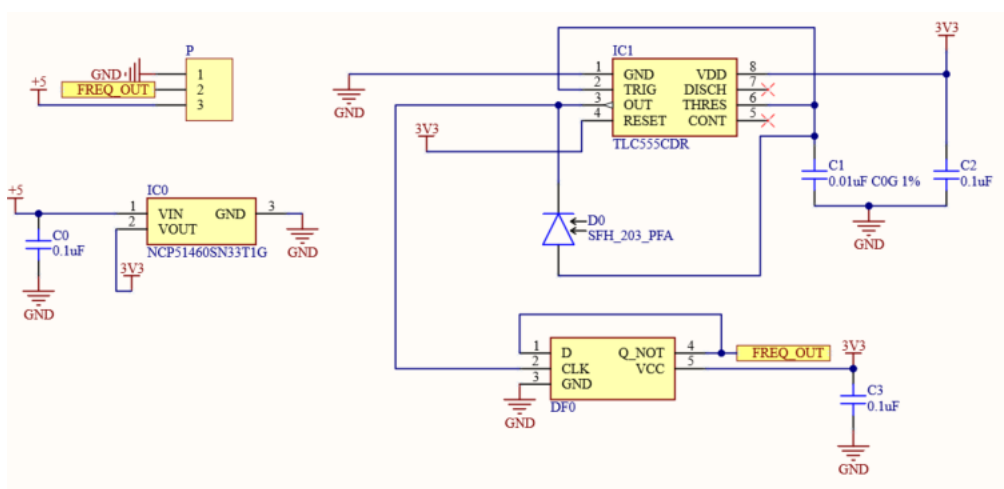
Tabelle 2: HT Feuchtigkeitssensoren des Messsystems [66][67]

Typ	Bezeichnung Sensorposition	Kapazität bei 30 % RH / pF	Maximale Einsatz Temperatur / °C
KFS330-MIN	HTA	200 ± 40	190
KFS140-D	HTB	150 ± 50	150



3.6.2 Licht zu Frequenz Konverter

Im Vergleich zu der vorherigen Schaltung wird hier der Wert des Kondensators fixiert (Abbildung 3-24) und die Entladung erfolgt nicht über den 555 Timer, sondern über eine Photodiode. Als Kondensator wird ein temperaturstabiles C0G Bauteil gewählt. Dieser Kondensator wird durch die Reverse Biased Photodiode geladen. Die Zeit zum Laden ist proportional zu der auf die Diode einfallenden Lichtintensität. Wird der $2/3 V_{CC}$ Schwellwert erreicht, kippt der interne Komparator in dem 555 Timer um und schaltet den Ausgang auf LOW. Der Kondensator wird über die Diode in Forward Richtung entladen. Die Dauer der Entladung, ist nur ein Bruchteil der Periode, die zum Laden benötigt wird. Dadurch erzeugt der Ausgang kurze Flanken. Damit ein Rechtecksignal entsteht, wird ein D Flip-Flop zu einem T Flip-Flop, durch Rückführung des Q Not Ausgangs auf dem D Eingang umfunktioniert. Der entstehende einflankengesteuerte Flip-Flop kippt den Ausgang bei jedem Vorkommen des Steuersignals (steigende Flanke) um.



3.7 IR Temperatur Sensor Modul

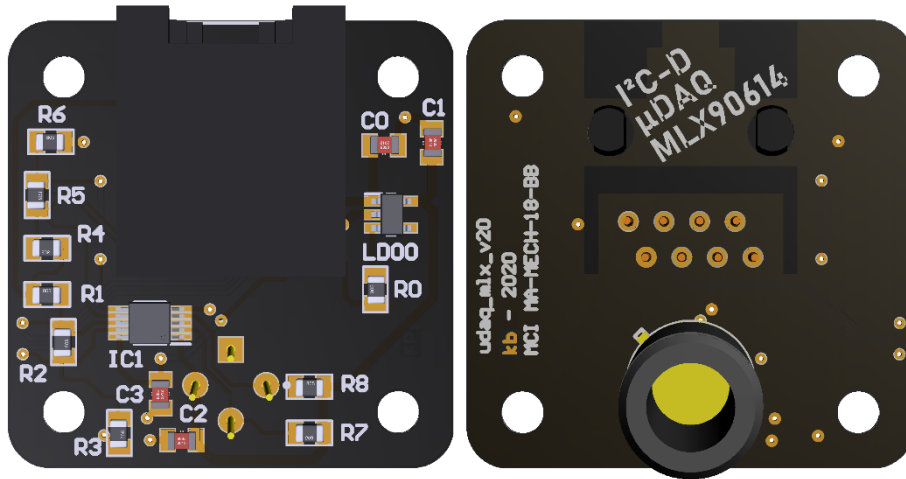


Abbildung 3-25: IR Temperatur Sensor

Der MLX90614 ist ein Automotive Grade Infrarotthermometer für die kontaktlose Temperaturmessung. Der IR Sensitive Thermopile Detektor Chip und das Signal Processing wird in einem TO-39 Gehäuse untergebracht. Die Kommunikation erfolgt über eine SMBus Schnittstelle, somit ist die Integration als I2CD Modul möglich. Der Sensor liefert zwei Temperatursignale -40 - 125 °C Umgebungstemperatur und -70 - 380°C Objekttemperatur. Es existieren auch Versionen des Sensors für medizinische Zwecke mit entsprechender Kalibrierung des Temperaturbereiches.

Zum Zeitpunkt der Entwicklung dieses Systems, ist die weltweite Corona-Pandemie ausgebrochen und dieser Sensor war, durch den Einsatz in kontaktlosen Fiebermesssystemen, weltweit ausverkauft (Information des Herstellers).

Es existiert zwar eine Ausführung des Sensors mit 5Grad FOV, welche sich für die Anwendung noch besser eignen würde, es konnte aber nur die BCF Version mit 5 Grad FOV angeschafft werden. Bei der Platzierung in einem Abstand von 30mm zu der bemessenen Oberfläche durch den FOV beträgt die Breite des Sichtfeldes:

$$(\tan(5^\circ) * 30mm) * 2 \approx 5.25mm \quad (31)$$

3.8 Farbsensor

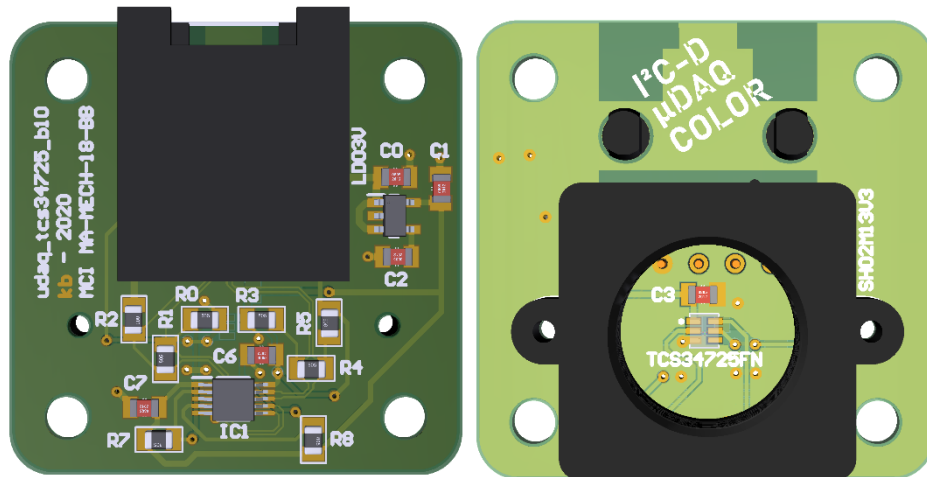


Abbildung 3-26: Farbsensor Modul

Es wäre möglich gewesen als Subsystem einen zusätzlichen Mikrocontroller nur für die Aufnahme einer Kamera zu implementieren oder zusätzlich noch einen Single-Board Computer einzusetzen um eine Kamera abzufragen, jedoch hätte es die Komplexität des Systems wesentlich erhöht und das entstandene Bild, hätte nur wenige Pixel notwendige Informationen beinhaltet. Der TCS34725 Farbsensor verfügt über ein 3x4 Photodioden Array mit IR Filter, welcher sich somit für den Einsatz neben Infrarotstrahlen eignet.

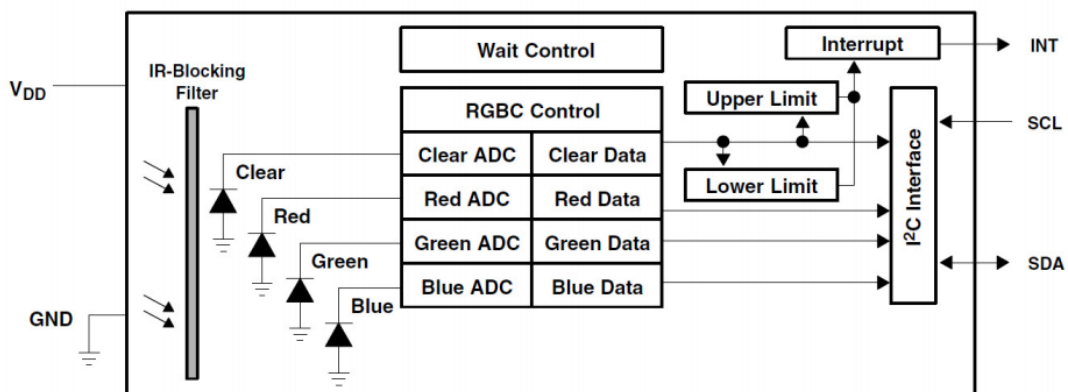


Abbildung 3-27: TCS34725 Blockschaltbild [69]

Der Sensor Liefert RGB Farbwerte und C als Gesamtintensität an einem getrennten Kanal. So wird erreicht, dass in dem Sensor 3 Photodioden mit R, G und B Filter versehen werden und die vierte Diode wird dem Licht ohne Filter ausgesetzt. Der IC integriert 16bit ADCs, welche die Photodioden

parallel sampeln und verfügt über eine I2C Schnittstelle mit 400kHz Datenrate. Die Schaltung ist identisch mit dem I2CD Port, welches beim Mainboard gezeigt wurde.

Die Brennweite f' wird gemäß der Abbildungsgleichung näherungsweise bestimmt. Mit a wird der abgezielte Arbeitsabstand, mit y' die Sensorgröße, und mit y die Gegenstandsgröße eingesetzt.

$$f' = \frac{a}{\frac{y}{y'} + 1} = \frac{27mm}{\frac{4mm}{0.4mm} + 1} \approx 2.45mm \quad (32)$$

Folgendes S-Mount Objektiv wird verwendet: Lensation BT2520, Focal Length: 2.5mm. Um das Objektiv über dem Sensor positionieren zu können, wird eine auf Leiterplatten montierbare Linshalterung verwendet.

Das Farbsensormodul selbst, verfügt über keine Lichtquellen. Da im Toasterinneren nur eine geringe Menge an Umgebungslicht herrscht, wird der Toast mit Auflicht-Beleuchtung von unten beleuchtet.

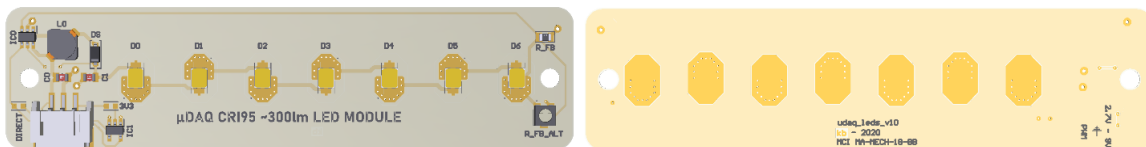


Abbildung 3-28: LED Modul

Für diese Aufgabe wird ein LED Modul entwickelt welches mit sieben J2835 CRI95 LEDs ausgestattet wird. Um von der Boardspannung auf die Summe der Forward Spannungen zu kommen, die durch die Serienschaltung der LEDs entsteht und um für eine gleichmäßige Helligkeit zu sorgen, wird ein Step Up Regler (AP3032) verwendet.

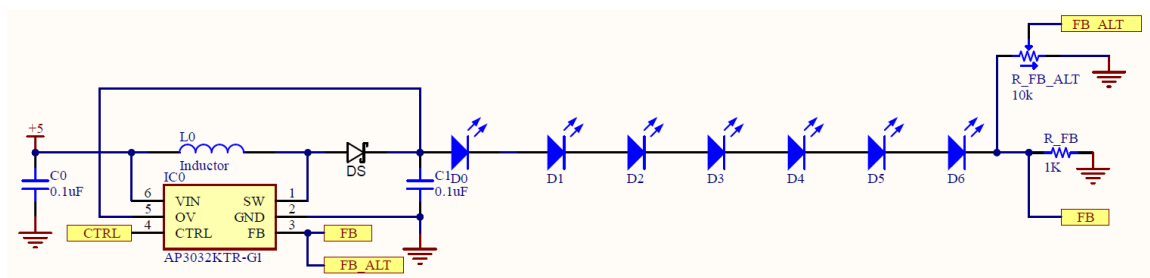


Abbildung 3-29: Schematik LED Modul

Für die Bestimmung des Feedback Stromes kann entweder ein SMT Potentiometer (R_FB_ALT) oder ein 0805 Widerstand (R_FB) bestückt werden.

3.9 Encoder

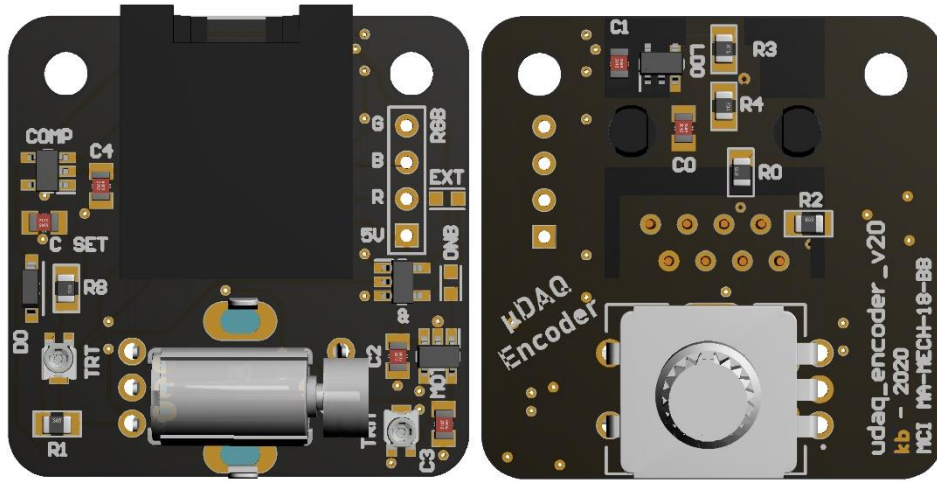


Abbildung 3-30: Encoder Modul

Die Eingaben können mit einem Drehencoder mit Taster erfolgen. Ein verzögerungsfreies haptisches und akustisches Feedback erfolgt durch die Logik, die mit einem Komparator und einem AND Gatter aufgebaut wird Abbildung 3-31. Durch einen Ausgang des Encoders (ENCA) wird während dem positiven Puls über den Pullup Widerstand R1 und den Potentiometer TRT der Kondensator C_SET aufgeladen. Beim negativen Puls des ENCA Ausgang wird der Kondensator über der Diode D0 und über den Widerstand R8 entladen.

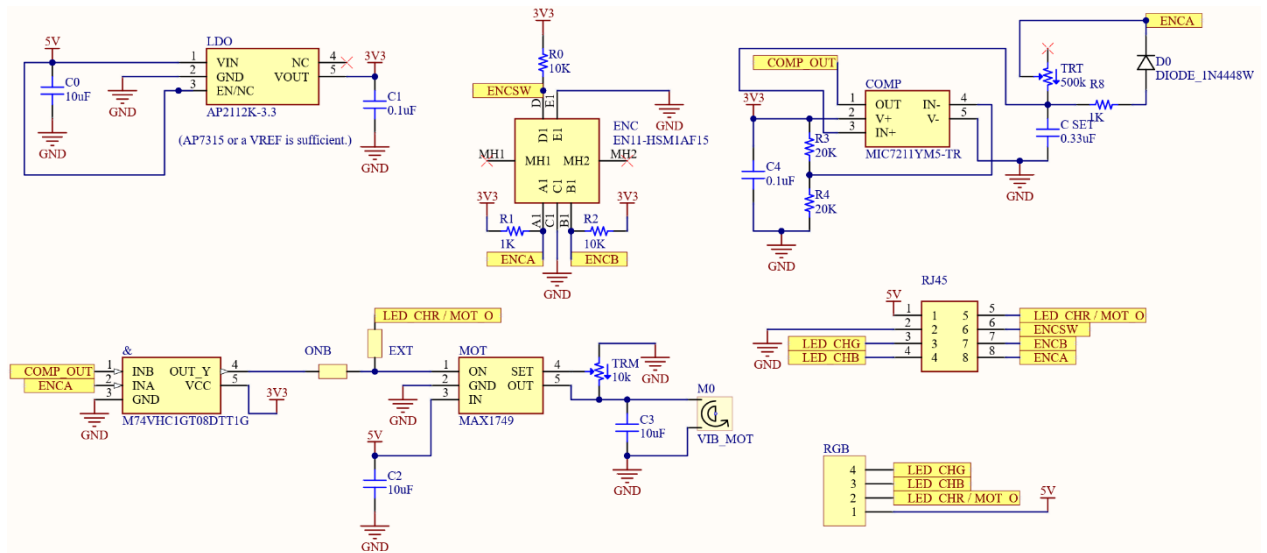


Abbildung 3-31: Encoder Schematik

Der Kondensator ist mit einem Komparator verbunden, welcher die Spannung zu der Spannung des Spannungsteilers bestehend aus R3 und R4 vergleicht. So lange die Spannung des Kondensators niedriger als die Referenzspannung ist, ist der Ausgang HIGH. Der Ausgang des Komparators ist zu dem AND Gatter verbunden, welcher als zweiten Eingang direkt das Signal ENCA sieht. Der Ausgang des And Gatters erzeugt nun Pulse, deren Dauer mit dem Potentiometer TRT gesteuert werden können. Rastet der Encoder während eines positiven Pulses am ENCA ein, wird trotzdem nur ein kurzer Puls erzeugt. Der Ausgang des AND Gatters steuert einen Vibrator Motor Driver an (MAX1749). Die Vibrationen werden mit einem zylindrischen durch Exzenterprinzip funktionierenden Vibrationsmotor (3160400051), welcher an der Rückseite direkt unter dem Encoder angebracht wird, erzeugt (Abbildung 3-30).

3.10 Diverse

3.10.1 ESP32 Programmer

Der ESP32 Programmer ist eine USB UART Schnittstelle, basierend auf dem IC CP2104 von Silicon Labs. Die Firmware des ICs wird von Silicon Labs zur Verfügung gestellt über Simplicity Studio angepasst und mit dem Namen des Gerätes sowie mit der Konfiguration der RX und TX Aktivität LEDs programmiert. Weitere ESP32 spezifische Bootstrap Konfigurationen werden durch Transistoren Logik realisiert. Die Steuersignale nDTR und nRTS steuern die EN, IO0 und IO2 Pins beim ESP32 an und somit wird das Gerät in Download Mode gesetzt. Der USB-Anschluss ist ESD geschützt durch Einsatz eines TVS Dioden Arrays.

3.10.2 Display

Das Display Modul verwendet ebenfalls die 32mmx32mm Baugröße, ein günstiges IPS LCD-Display mit 1.3" Diagonale, 240x240 Pixel Auflösung und 16bit Farbtiefe. Das Display nutzt den ST7789V Controller mit SPI Schnittstelle. Das Display Modul verfügt über den einheitlichen RJ45 Stecker und einen Transistor um die Hintergrundbeleuchtung zu schalten, sowie einen eigenen LDO für die Spannungsversorgung.

4 Aufbau des Protoyps

Um die Entwicklungszeit zu verkürzen, wurde ein bestehender Toaster mit der Sensorik und notwendigen Elektronik retrofitted, anstatt einen neuen Typ zu konstruieren. Der ausgesuchte Toaster von Dualit (Abbildung 4-1) aus dem Jahr 1994 gehört zu der Familie der Turn Up Toaster und die Brotausgabe und das Absenken erfolgen per Hand, über Betätigen eines Hebels. Von den drei Toasterschlitzten, wird nur der rechte Schlitz mit Heizelementen und Sensorik ausgestattet.



Abbildung 4-1: Dualit 2+1 Slice Toaster

Abbildung 4-2 zeigt die Explosionszeichnung von verbauten elektronischen und mechanischen Komponenten.

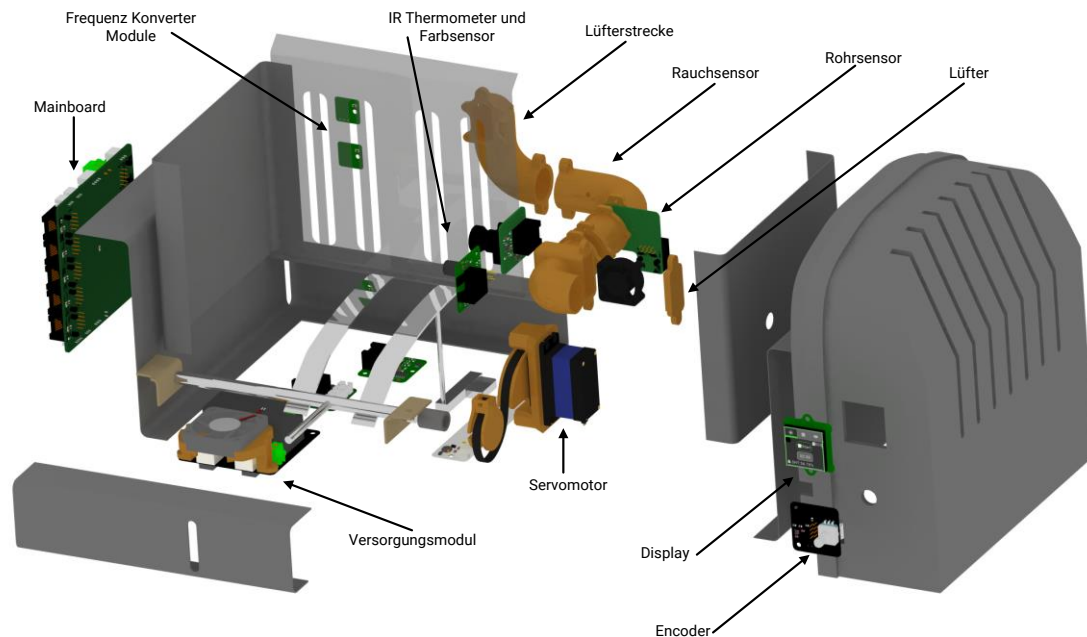


Abbildung 4-2: Explosionszeichnung des Prototyps

An der Seitenwand werden die zwei Konverterboards an das kalte Ende des Toasters angebracht. Das Mainboard ist auf der linken Seite des Toasters befestigt. Auf der Grundplatte befinden sich folgende Einheiten:

- Versorgungsmodul
- Wägemesszelle und Auslesemodul
- Analogmodul
- Auflicht Beleuchtung

An dem rechten Gussteil wird das Display aufgeklebt und der Encoder verschraubt, Anhang A1. Ein zugeschnittenes Plexiglas vor dem Display füllt die Aussparung für den originalen Schalter aus. Der Farbsensor sowie das IR Thermometer werden an der rechten Seitenwand des Toasters montiert.



Abbildung 4-3: Pressglimmerplatte mit Löchern für optischen Sensoren

Als Heizelemente wurden neue Dualit-kompatible Heizelemente eines Drittherstellers verwendet, die an der Montageposition der optischen Sensoren durchgebohrt wurden. Diese Heizelemente verfügen über einen Spalt von 9mm zwischen zwei Nichrome Windungen. Die Elemente werden parallelgeschaltet. Der Widerstand eines einzelnen Elementes beträgt 140Ω und die gesamte Heizleistung somit ca. 750W. In der Lüfterstrecke wird der Rohrsensor mit O-Ring Dichtungen verschraubt, sowie der Lüfter seitlich in eine Tasche eingeschoben. Die Kunststoffteile des Toasters werden aus Carbon Fiber Composite HTPLA (Protopasta) mit FDM Technologie und teilweise aus hitzebeständigem Photopolymer Harz (Druckwege Type D) mit UV DLP Technologie gedruckt. HTPLA (Protopasta) ist für eine Umgebungstemperatur von 155°C zugelassen, Photopolymer Harz für 260°C . An der Rückseite des Toasters verläuft die Lüfterstrecke, dessen Segmente mit O-Ringen abgedichtet sind. An der Außenseite beider Heizelemente wurden Keramikfasermatten mit Alubeschichtung aufgebracht. An der linken Seite liegt eine zweite Isolationswand.

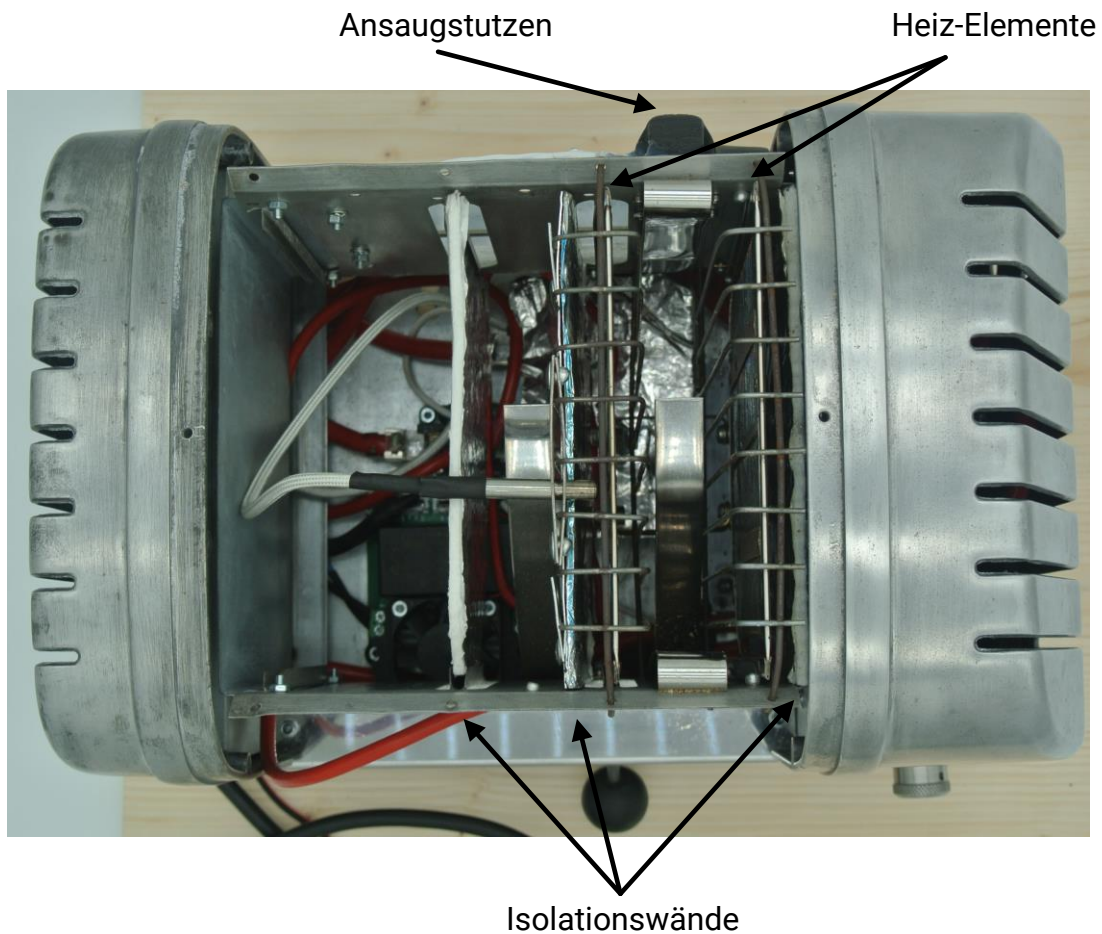


Abbildung 4-4: Obere Ansicht – fertiger Prototyp

4.1 Waage in einem Toaster

Durchgeführte Versuche zeigen, dass ein durchschnittliches Buttertoastbrot 14.4% Prozent seines Gewichtes in Form von Flüssigkeit verliert. Um diese Gewichtsreduktion während des Toastvorgangs feststellen zu können, wurde eine Lösung gesucht. Nach verschiedenen Versuchen mit Schrittmotoren und Hebelkonstruktionen, wurde eine Lösung unter Verwendung eines Servomotors und eines Riemens gefunden. Der Servomotor aus dem Modellbaubereich wird in einem Hilfsrahmen auf dem Seitenpanel verschraubt. Abbildung 4-5 zeigt den Toaster in der Seitenansicht. Die eingelegten Toasts werden durch eine Blechzunge von unten gehalten. Diese Zunge endet in einer Welle, welche normal auf die gezeigte Fläche ist. Durch drehen dieser Welle kann ein Toast angehoben oder abgesenkt werden.

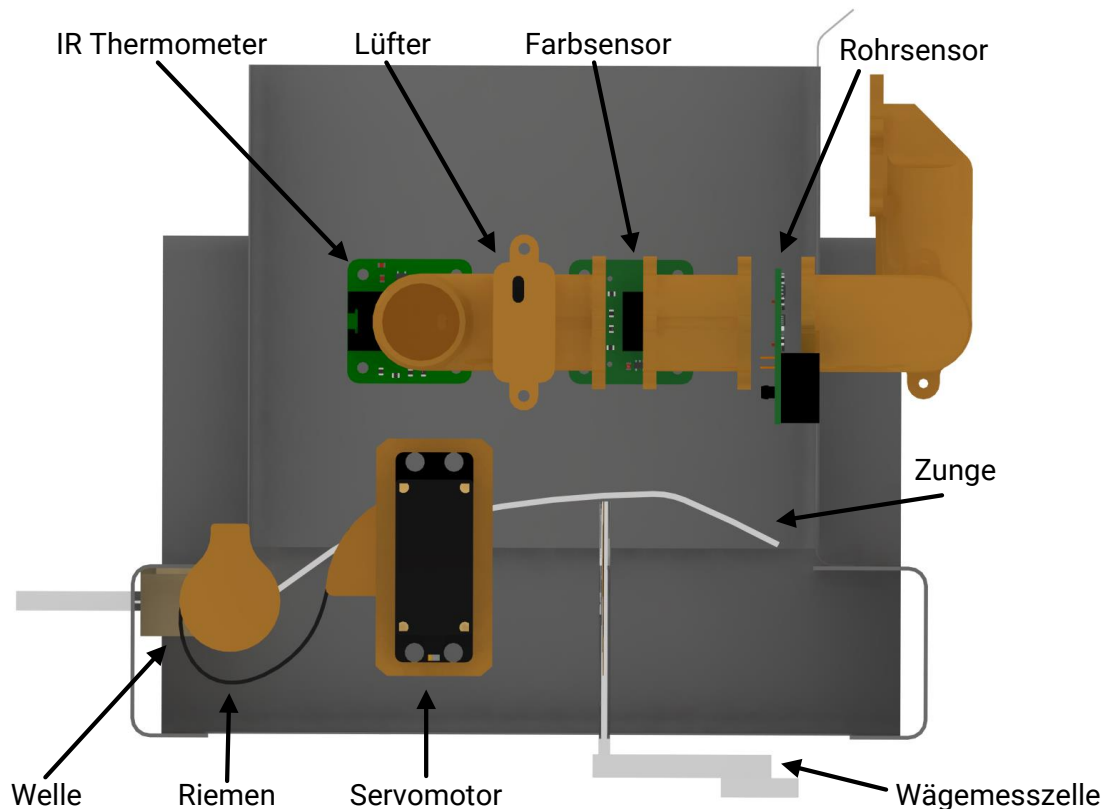


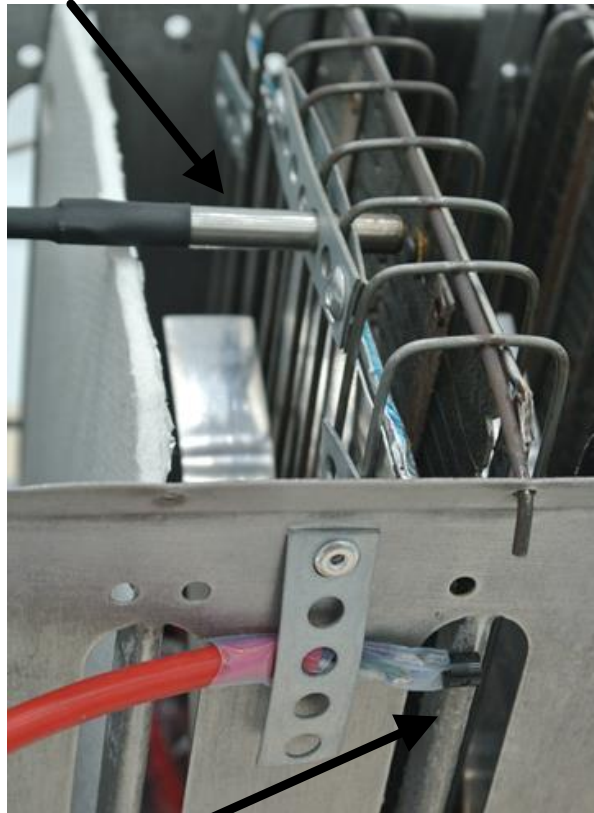
Abbildung 4-5: Seitenansicht - Waage

Am Ende dieser Welle wird eine Scheibe befestigt, damit diese zum Arretieren eines Riemens verwendet werden kann. Der Servomotor wird mit einer Aufnahme in der Form eines Kreissegments für den Riemen versehen. Auf diesem wird der Riemen mithilfe einer M3 Mutter befestigt. In vollständig abgesenkter Position liegt die Zunge an einer Stange an, welche die Verlängerung der Wägemesszelle ist. Wird der Toast abgesenkt, kann der Servo noch weiterdrehen nachdem die Zunge bereits aufliegt, wodurch der Riemen frei hängt und die Welle mechanisch entkoppelt wird.

4.2 Positionierung der Sensoren

Ein PT1000 Sensor im Metall Gehäuse wird mit Lochblechband und Vernietung direkt über dem inneren Heizelement angebracht. Ein analoger LMT86 Temperatursensor wird direkt an der Seite des Toasters montiert.

PT1000



LMT86

Abbildung 4-6: Platzierung der Temperatur Sensoren

Die Lüfterstrecke wird an der Seitenwand des Toasters verschraubt. Die Stelle, wo der Flansch anliegt, wird mit zusätzlichen Löchern versehen.

Direkt im Flansch, am Anfang der Lüfterstrecke, ist ebenfalls ein PT1000 Sensor montiert, in diesem Fall ohne Metall Gehäuse. Über dem PT1000 Sensor im Flansch wird der beim Frequenzkonverter 3.6.1 beschriebene HTA kapazitive Polymerfeuchtesensor montiert. Die Sensoren werden mit Silikon in dafür vorgesehene Öffnungen des Flansches eingeklebt. An der rechten Seite des Bildes sind zwei Kapazität zu Frequenz Konverter Boards ersichtlich. An dem Unterem Board ist ein weiterer Polymerfeuchtesensor (HTB) direkt montiert.

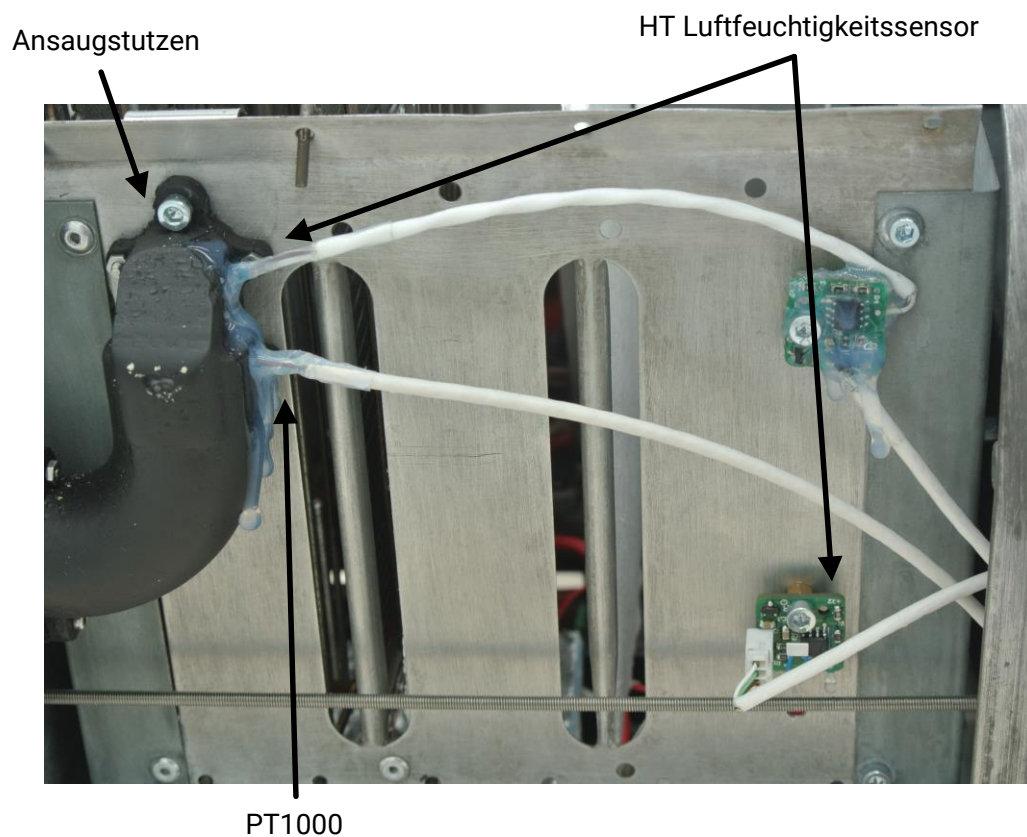


Abbildung 4-7: Platzierung der Sensoren in der Lüfterstrecke

Im zweiten Abschnitt befindet sich der Rauchsensor. An der Unterseite des Rohres wird die Photodiode montiert und die zwei Emitter in den in 4.4 vorgestellten Winkeln. Die Lüfterstrecke setzt sich nach einem Knick an der Seite des Toasters fort, dort ist der Rohrsensor ersichtlich sowie hinter diesem die beiden optische Sensoren. Die Strecke selber ist durch zweckentfremdete O-Ringe elastisch aufgehängt. Die Toastausgabemechanik ist in dem linken unteren Bereich ersichtlich.

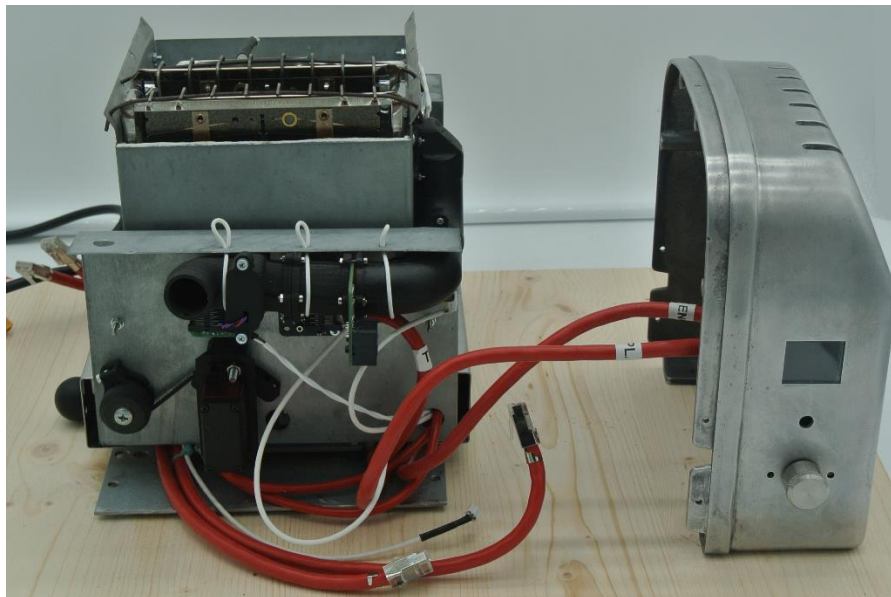


Abbildung 4-8: Seitenansicht des fertigen Prototyps

Auf der Grundplatte wird die Wägemesszelle auf ein Aluminiumpad geschraubt. Von oben wird diese ebenfalls vor Hitze isoliert. Die Beleuchtung wird für den Farbsensor in einem geneigten, auf die Toasteroberfläche gerichteten Winkel montiert. Auf der Grundplatte wird die Temperatur direkt am MOSFET Treiber sowie bei der Wägemesszelle gemessen, ein dritter Temperatursensor ist direkt nach außen geführt.

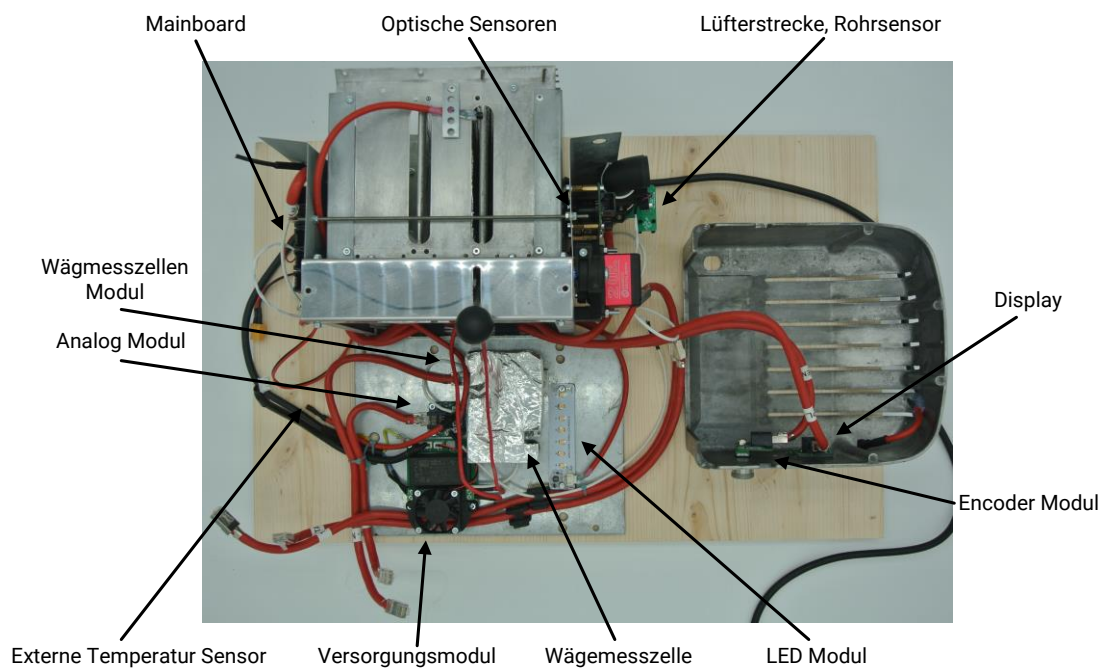


Abbildung 4-9: Grundplatte des fertigen Prototyps

Im Anhang A1 sind noch weitere Aufnahmen vom Prototyp zu sehen, im Anhang A2 wird ein Teil der weiteren mechanischen Modifikationen am Toaster aufgelistet.

4.3 Lüfterstrecke

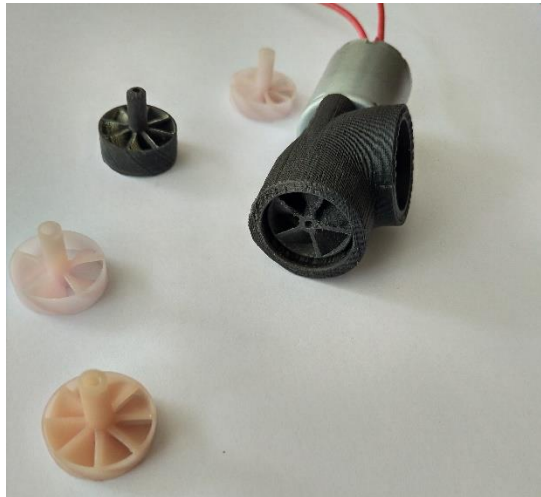


Abbildung 4-10: Erster Prototyp des Kurvenlüftergehäuses und mehrere Versuche von Lüfterrädern

Damit ein Gassensor, Luftfeuchtigkeitssensor oder Rauchsensor, welche nicht für sehr hohe Temperaturen geeignet sind, direkt in dem Luftstrom platziert werden können, musste eine Lösung gefunden werden. Es wurde eine Lüfterstrecke verwendet, welche die Luft des Toasterinnenraumes teilweise ansaugt. Dieser Ansaugstutzen ist von der direkten Hitzequelle weit genug entfernt, um eine Vermischung mit genug Umgebungsluft zu ermöglichen. Damit bekommt man eine für die Bauteile erträgliche Temperatur, aber trotzdem noch die charakteristischen Informationen aus dem Röstraum.

Die erste Überlegung war es, einen Venturi Injektor für die Erzeugung des Fluidstromes zu verwenden. Jedoch wurde nach wenigen vereinfachten Simulationen klar, dass für geringe Luftströme über 10 Watt an Lüfterleistung nötig wären.

Die Suche nach einem Lüfter, welcher die erlaubte Betriebstemperatur von mindestens 100°C zulässt und eine Baugröße von 40mm × 40mm × 40mm nicht überschreitet, endete erfolglos. Es wurde ein Kurvenlüfter samt Lüfterrad entworfen. Bei diesem Konzept ist der Motor nicht direkt dem heißen Luftstrom ausgesetzt. Nach vielen Variationen wurde diese Lösung ebenfalls verworfen. Die Kombination aus bürstenbehaftetem DC Motor und Lüfterrad, bei 5V Bordspannung und die daraus folgende Drehzahl hat sich als nicht ausreichend erwiesen und lieferte bei dieser Baugröße zu wenig Luftstrom. Im Anhang A1 sind noch weitere Bilder zu dieser Lösung ersichtlich. Mit der Verwendung von verkleinerten BLDC Motoren aus dem Quadrocopter Bereich

würde sich diese Art von Lüfter realisieren lassen. Jedoch zeigten die ersten realen Messungen, dass die Lüfterstrecke maximal 85°C erreicht.

Daher eignet sich für den Prototyp als temporäre Lösung auch ein kommerzieller Lüfter, mit einem Temperaturbereich bis 70 Grad, anstatt ein eigens Lüfterkonzept und die Ansteuerungselektronik zu konstruieren. Die Halterung für den Lüfter wurde austauschbar konstruiert, sollte es durch erhöhte Beanspruchung während der Messung zu Defekten kommen. (Delta ASB02505SHA-AY6B Lüfter 25 × 25 × 10mm mit 5V und 4.3 CFM)

4.4 Rauchsensor in einem Toaster

Geplant ist, einen kostengünstigen Rauchsensor in der Lüfterstrecke zu integrieren. Dazu wird ein optisches Messverfahren gewählt, welches auf die Streuung von Licht durch Rauchpartikel beruht. Deng in [44] und [43] untersuchte die Einsatzmöglichkeiten eines Sensors mit zwei Emittoren (400nm und 800nm) und einer Photodiode. Durch diese Arbeit wurden die optimalen Winkel bestimmt die für die Detektion von schwarzen sowie grauen Rauch geeignet sind [43]. Durch den Einsatz kann zusätzliche Robustheit erreicht werden, indem das Fluid mit unterschiedlichen Wellenlängen des Lichtes geprobt wird. Es wurde eine UV LED (405nm VAOL-3GUV0Y4), eine IR LED (850nm OP265FAB) und eine Photodiode (SFH203P) mit einem sensiblen Bereich für 400nm-1000nm ausgewählt. Der UV-Emitter wurde wie [44] entsprechend in den Winkeln 112° und in 45° angeordnet (Abbildung 4-12).

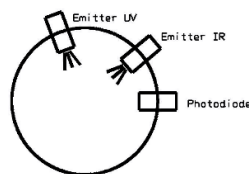


Abbildung 4-11: Anordnung der Komponenten des Rauchsensors

Der Zustand des Sensors wird in einem 10ms Zeitfenster aufgeteilt, wie es in der Abbildung 4-12 ersichtlich ist. Für drei Perioden wird jeweils mit der UV-Diode (blau) oder dem IR (rot) beleuchtet, dazwischen werden beide LEDs für eine Periode ausgeschaltet (schwarz), um die Lichtmenge die durch Umgebungslicht vorhanden ist, zu messen. Die grauen Perioden in Abbildung 4-12 werden nicht berücksichtigt, da hier zwischen den Emittoren umgeschaltet wird.

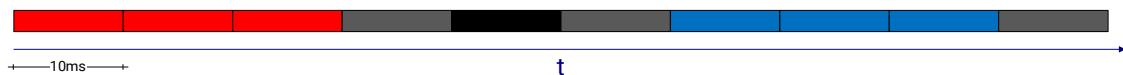


Abbildung 4-12: Messablauf Rauchsensor

Der Rauchsensor wurde bei den finalen Messungen nicht verwendet, obwohl er während der Prototypisierung ein gutes Ansprechverhalten mit Räucherstäbchen gezeigt hat. Während den Messungen wurde ein Toast nie so stark verbrannt, dass er Rauch produziert hätte. Die Methode

basierend auf Lichtstreuung, hat sich für diese Art der Messungen in einer Lüfterstrecke nicht bewährt. Untersuchungen mit einem Verfahren, das auf Absorption von Licht basiert, könnte noch weitere Ergebnisse liefern.

4.5 Fertiger Prototyp, Übersicht der Sensorik

Das Gehäuse konnte samt Elektronik erneut verschlossen und in Betrieb genommen werden, die Abbildung zeigt den fertigen Prototyp.



Abbildung 4-13: Fertiger Prototyp

Die Tabelle 3 zeigt in der Übersicht die montierten Sensoren und deren Position, sowie den Signalnamen, mit dem die Zuordnung im Source-Code sowie in den Abbildungen möglich ist.

Tabelle 3: Übersicht Sensor Kanäle des Messsystems

Gemessene Eigenschaft	Signal Name	Sensor Position
Temperatur	PT1000A	Im Ansaugflansch
Temperatur	PT1000B	Über dem Heizelement
Temperatur	A0	An der Leistungselektronik
Temperatur	A1	Bei der Wägemesszelle
Temperatur	A2	An dem Seiten Panel im Toastraum
Temperatur	A3	Außerhalb vom Gehäuse
Luftfeuchtigkeit (HTA)	F1	Im Ansaugflansch
Luftfeuchtigkeit (HTB)	F2	An dem Seitenpanel im Toaster, nicht im Toastraum
Farbe	R	Farbsensor
Farbe	G	Farbsensor

Farbe	B	Farbsensor
Intensität	C	Farbsensor
Temperatur	MLX_TA	IR Thermometer Ambient Temperatur
Temperatur (Kontaktlos)	MX_TO	IR Thermometer Objekt Temperatur
Luftfeuchtigkeit	SHT_H	Rohrsensor - SHT31
Temperatur	SHT_T	Rohrsensor - SHT31
Hydrogen	rawH2	Rohrsensor - SGP30
Ethanol	rawEth	Rohrsensor - SGP30
eCO2	eCO2	Rohrsensor - SGP30 (Im Sensor Kalkuliert)
VOC	VOC	Rohrsensor - SGP30 (Im Sensor Kalkuliert)
Absolute Luftfeuchtigkeit	Abs. Humi.	Rohrsensor - SHT31 (Im MCU kalkuliert)
Gewicht	Waage	Wägezelle mit NAU7802

4.6 Firmware

Der ESP32 Toaster ist ein CMake Projekt und baut auf das ESP-IDF auf. Dies beinhaltet eine komplette Toolchain für die Programmierung, Debugging des ESP32 sowie Low Level Driver für die Peripherien. Das Projekt nutzt dessen SPI Master, Wifi Driver etc. Die Details der Entwicklung werden nur teilweise erläutert, in der digitalen Anlage ist die Firmware für den Toaster vollständig enthalten. Um die zwei Kerne des Controllers optimal ausnutzen zu können und um die Entwicklung zu erleichtern, setzt das ESP-IDF auf FreeRTOS. Die wichtigsten Tasks der Anwendung und deren Prioritäten werden übersichtsmäßig dargestellt (Abbildung 4-14) und in den folgenden Absätzen erklärt.

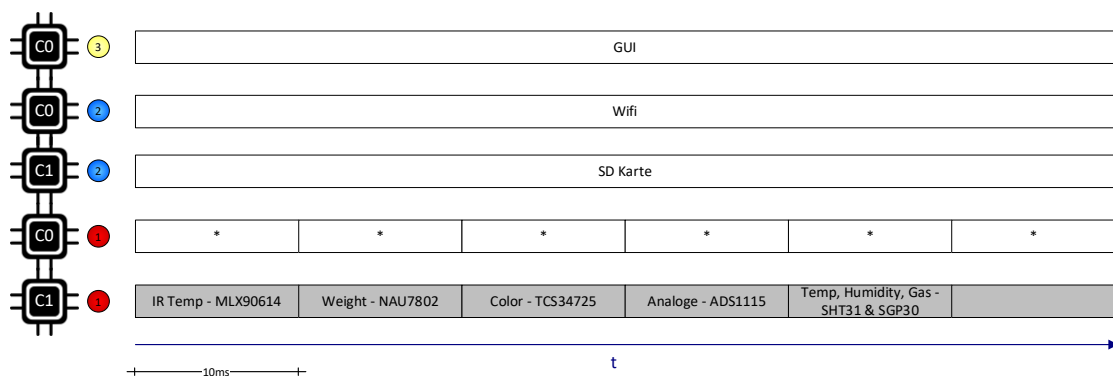


Abbildung 4-14: Aufteilung der Aufgaben zwischen den Kernen

4.6.1 DAQ

Um 8 verschiedene I2C Geräte zeiteffizient in Betrieb zu nehmen werden grundsätzliche Lese- und Schreibfunktionen, die auf den ESP-IDF Interrupt basierten Low Level Driver aufbauend erstellt. Doch zeigen I2C Geräte in der Realität viele Eigenheiten die angepassten Routinen benötigen. Wo existierende Bibliotheken in C oder C++ für anderen Plattformen (wie STM32, Arduino, MSP430, Mbed etc.) gefunden werden, werden diese portiert. Zeitlich kritische Programmteile die zyklisch abgearbeitet werden müssen werden gestoppt und relevante Bereiche werden neu geschrieben.

Ein Timer mit 10ms Interrupts wird konfiguriert. In diesem ISR erfolgt das Auslesen von allen Pulse Counter Modulen, welche die 4 Frequenzeingänge sampeln. Dieses Interrupt aktiviert den Main Control Task (C0_I2CL) direkt durch einen Semaphor. Das Context Switching des FreeRTOS Schedulers dauert in diesem Fall nur wenige μSec , da ein Task mit hoher Priorität aufgeweckt wird. Die Abbildung 4-15 zeigt das vereinfachte Ablaufdiagramm, beginnend mit dem Auslösen dieses Timers.

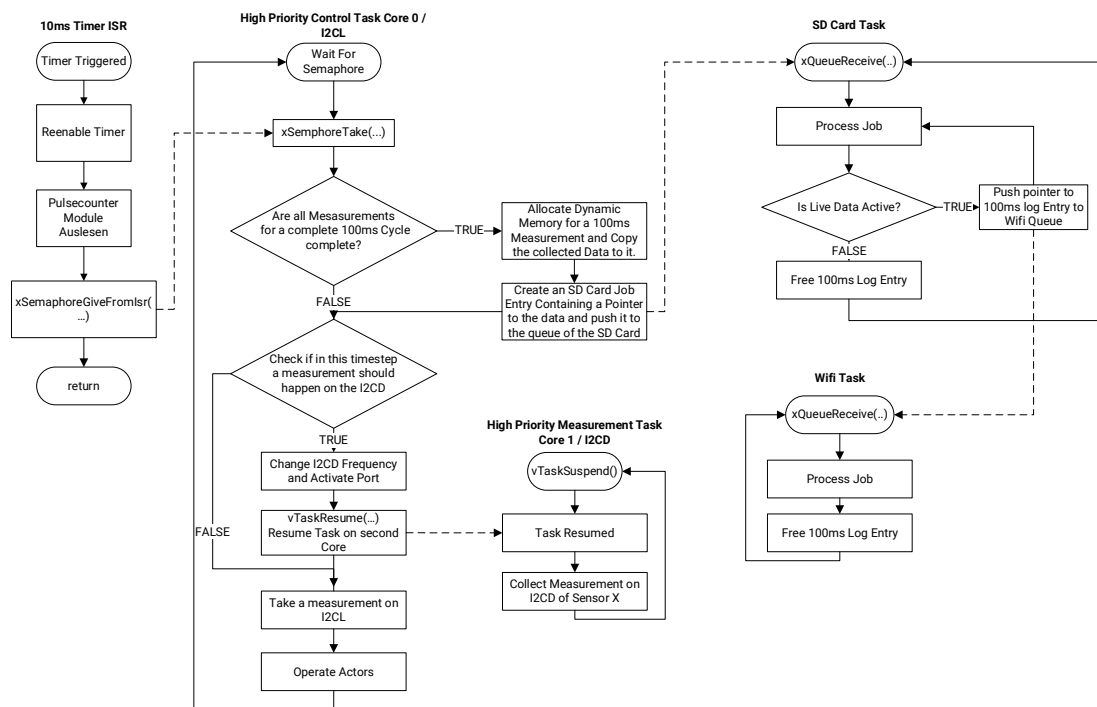


Abbildung 4-15: Vereinfachter Ablauf DAQ

Es werden zehn 10ms Zeitfenster definiert, die während eines 100ms Messzyklus erfolgen, in der Abbildung 4-14 sind diese mit einem Stern markiert. Die I2CD Module werden in der Reihe nach, in

diesen 10ms Fenster abgefragt. Diese Abarbeitung erfolgt an dem zweiten Core in dem C1_I2CD Task, welcher in der Abbildung 4-14 grau markiert ist. Da die I2CD Geräte, welche eine getrennte I2C Peripherie verwenden, in einem getrennten Kern abgearbeitet werden, kann parallel zu diesen Schritten der Core 1 die I2CL Module abfragen und weitere Aufgaben erledigen. Dadurch ergibt sich, dass jedes I2CD Modul alle 100ms abgefragt wird und die I2CL ADCs alle 10ms.

Der C0_I2CL Task prüft zu Beginn jedes 10ms Zeitfensters, ob ein I2CD Port in dieser Periode abgefragt werden soll. Dementsprechend kann der I2CD Port über den am I2CL verbundenen GPIO Expander aktiviert werden und den Task starten (vTaskResume() im Ablaufdiagramm). Der Task C1_I2CD ist ebenfalls ein High Priority Task und wird sofort gescheduled und fortgesetzt. Beide Tasks schreiben die Messdaten in einen globalen Struct des Typs log_entry_100ms_t.

```
typedef struct __attribute__((packed))
{
    int16_t pt1000a[10];
    int16_t pt1000b[10];
    int16_t board[10];
    int16_t freq0[10];
    int16_t freq1[10];
    int16_t freq2[10];
    tcs3472x_color_t col;
    int16_t a[4];
    int32_t weight;
    uint16_t mlx_object;
    uint16_t mlx_ambient;
    uint16_t aq_TVOC;
    uint16_t aq_eCO2;
    uint16_t aq_rawH2;
    uint16_t aq_rawEthanol;
    int32_t sht_t;
    int32_t sht_h;
    float smoke_ir;
    float smoke_uv;
} log_entry_100ms_t;
```

Dieses Struct definiert auch wie die Daten in dem Logfile geschrieben werden und spielt noch eine zentrale Rolle. `__attribute__((packed))` bewirkt, dass der Speicher lückenlos gefüllt wird und kein Padding verwendet wird. Die Definitionen die auch von der PC-Anwendung verwendet werden sollen, wie dieser Struct werden in dem x_platform.h und x_platform.c festgehalten.

In beiden Tasks wird ein FSM abgearbeitet, wobei der Haupt Control Task (C0_I2CL) neben sampeln der I2CL ADCs noch weitere Aufgaben ausführt:

- Steuerung des Startens und Stoppens einer Messung
- Steuerung des Servos
- Steuern des LED Moduls (für den Farbsensor)
- Steuerung des Rauchsensors (IR und UV Leds), sowie Berechnung des prozentualen Rauchwertes

Sind die Daten vollständig vorhanden, alloziert der C0_I2CL Task ein Stück dynamischen Speicher in der Größe eines log_entry_100ms_t und kopiert die Daten aus dem globalen Struct in den neu allozierten Speicher. Es wird ein Job für die SD-Karte hinzugefügt mit dem Pointer zu dem

Speicher und mit dem Kommando, dass es in einem aktuell offenen Logfile geschrieben werden soll. Beim Starten und Stoppen einer Messung signalisiert der Task an die SD-Karte durch die entsprechenden Jobs, dass ein neues Logfile beginnt oder abgeschlossen werden soll.

4.6.2 SD-Karte

Es wird die Chen FATFS Library verwendet um das Filesystem zu handhaben. Die SD-Karte wird in dem Sinne abstrahiert, dass sie zu einem Gerät wird, welches Jobs annimmt, die in einem Queue gereiht werden. Die Jobs dieses Queues werden in einem unterpriorisierten separaten Task abgearbeitet. Nach dem Boot wird die kleinste unbesetzte Log File Nummer bestimmt und der Task setzt sich in Warteposition. Die folgenden Job Typen werden definiert:

- `SD_WRITE`
- `SD_CREATE_NEW_FILE`
- `SD_FINISH_FILE`
- `SD_PROCESS_NFO_LOG`

Beim `SD_CREATE_NEW_FILE` und `SD_FINISH_FILE` wird das Logfile mit der aktuellen Nummer angelegt oder abgeschlossen. Beim `SD_WRITE` schreibt der Task den erhaltenen `log_entry_100ms_t` in dem aktuell offenen Logfile. Ist die Option Live Data in dem GUI des Toasters selektiert, werden die dynamischen Speicherstücke nach dem Schreiben nicht freigegeben, sondern an den Wifi-Queue zugestellt. Ist diese Option nicht aktiv, wird der Speicher sofort freigegeben. Der Vorteil von dieser Hintereinanderreihung von Konsumenten des gleichen Datenstückes ist, dass die gleichen Daten nur einmal am Anfang der Kette kopiert werden müssen. Die Queues stellen sicher, dass die Reihenfolge der Datenstücke sich nicht ändert.

Bei dem `SD_PROCESS_NFO_LOG` Job, wird ein File mit der Nummer und Metadaten (wie Klassifikationsinformationen) der aktuellen Messung gespeichert. Diese Art an asynchroner Verarbeitung funktioniert stabil und unauffällig.

Es werden also insgesamt zwei Files je Messung gespeichert, die `RLD####.bin` welche in 100ms Stücke gruppierte Messdaten beinhaltet und die `CLS####.bin`, welche zusätzliche Klassifikationsinformationen beinhaltet und später vorgestellt werden.

4.6.3 Wi-Fi

Es werden mehrere Netzwerkkonfigurationen hardcodiert hinterlegt (Heimnetzwerk, Mobiler AP). Nach dem Boot versucht der Toaster eines von diesen Netzwerken zu finden und sich zu verbinden. Der Zustand der Verbindung wird angezeigt:

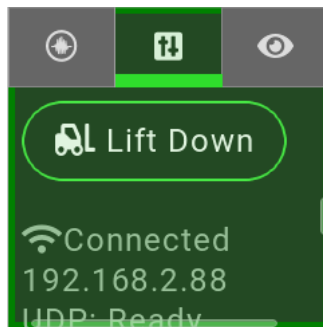


Abbildung 4-16: Toaster GUI zeigt den Verbindungsstatus und den Button um den Toast abzusenden

Nach der Verbindung wird eine neue IP-Multicast Gruppe erstellt. Vorteil von Multicast Kommunikation ist, dass die Teilnehmer die IP-Adresse voneinander nicht kennen müssen, nur die fixierte Multicast Adresse. Mehrere Teilnehmer können das gleiche Paket erhalten. Als Transport Layer Protokoll wird UDP verwendet. Ähnlich der SD-Karte verfügt Wi-Fi über einen eigenen Task und arbeitet einen Query ab.

4.6.3.1 Live Data

Aktiviert der User den Live Data Mode durch das GUI, wird jede in dem Wifi Queue landende `log_entry_100ms_t` sofort an die Multicast Adresse 224.224.224.224 verschickt. Nach dem Versenden wird das Speicherstück befreit.

4.6.3.2 Download Mode

Wird der Download Mode durch das GUI aktiviert, beginnt der Toaster zyklisch eine Nachricht auf die Multicast Adresse zu schicken.

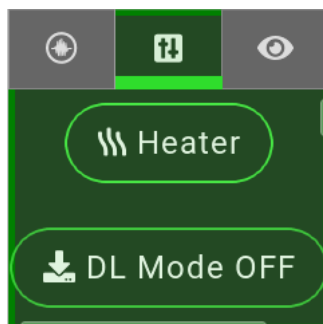


Abbildung 4-17: Ansteuerung Heizer und Einschalttaste für den Download Mode

Somit bewirbt er seinen Service - Download Mode - auf dem Netzwerk. Dadurch erhalten die Multicast Teilnehmer auch direkt die Adresse des Senders (Toaster). Der Download Mode aktiviert eine State Maschine, welche das einzelne Abfragen des Logfiles ermöglicht. Wird ein Logfile

angefragt, schickt der PC eine Nachricht direkt an die IP-Adresse des Toasters, somit wird dem Toaster die Adresse des Download Clients bekannt. Die weitere Kommunikation während des Downloadvorgangs erfolgt immer direkt (unicast) zwischen dem Server (Toaster) und dem Download Client (PC).

4.6.4 GUI

Das GUI läuft in einem eigenen unterpriorisierten Task, abhängig von der Systemlast, mit bis zu 33 FPS. Um die Display Buffer zu zeichnen wird LVGL verwendet, ein Open Source in C geschriebene leichtgewichtige Bibliothek. Das Updaten des Displays erfolgt über einen DMA gestützten SPI Driver, wodurch nicht das ganze Display, sondern nur Teile des gesamten Pixel-Arrays gebuffert werden. Die Aktualisierung des Displays erfolgt für Teilbereiche die Änderungen beinhalten, es wird nicht bei jeder Aktualisierung der gesamte Displayinhalt transferiert. Die verwendeten Icons werden durch Zusammenführung der Roboto Font und des Fontawesome UTF8 Font Types erzeugt. Teile des entstehenden Fonts werden in dem Projekt importiert. Die Ikonen als Glyphen benötigen dadurch nicht nur weniger Speicher, sondern können auch einfacher gehandhabt werden, das Setzen von Ikonen in Labels wird im Programmcode zu einer Aktion ähnlich zu einem `printf()`. Die erstellten Screenshots sind auf einem PC erstellt worden, mit LVGL ist es möglich den Bildschirminhalt mittels beispielsweise SDL zu rendern. Der Sourcecode ist identisch mit dem am Mikrocontroller, nur die Werte der Variablen die das reale System liefern würde, wurden für diesen Zweck künstlich hinterlegt.

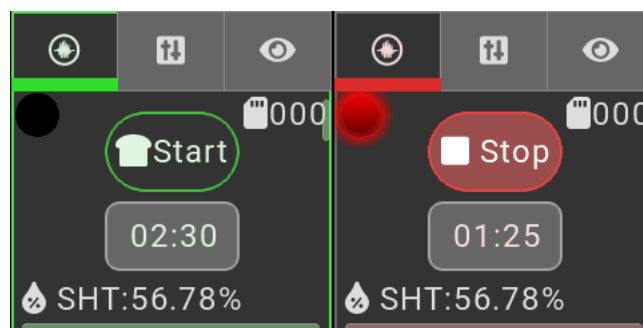


Abbildung 4-18: Hauptbildschirm des Toasters

Die Bedienelemente werden in Gruppen zusammengefasst, welche für die Navigation an der aktuellen Maske benötigt werden. Durch Drehen des Encoders, welcher in einem 1ms Tick Hook ausgelesen wird, wechselt man zwischen den Elementen und durch Drücken werden diese selektiert. Aktionen die im GUI erfolgen, führen zum Aufrufen von Callback-Funktionen, in denen die entsprechenden Flags gesetzt werden, die für die verschiedenen FSMs, welche in dem System laufen, relevant sind. Das GUI (Abbildung 4-19) zeigt den Wert von jedem Sensor an, dies erwies sich als hilfreich, während der Fehlersuche und der Konfiguration der Sensoren. Die Hauptfarben des GUIs ändern sich, wenn ein Toastvorgang aktiv ist, Abbildung 4-18 zeigt den Hauptbildschirm in beiden Modi.

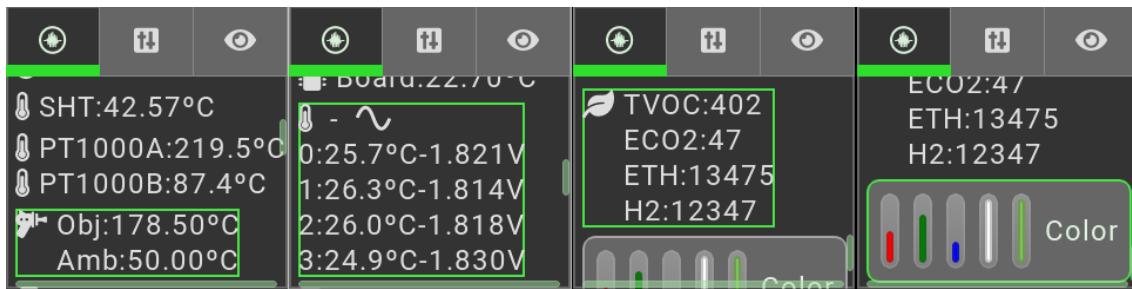


Abbildung 4-19: Anzeige der Sensordaten

Durch das GUI können auch alle Aktoren des Systems getestet werden. Durch Knopfdruck ist die Ansteuerung des Toastliftes und das Ein- und Ausschalten der LED-Beleuchtung und des Heizelementes möglich.

4.6.5 Heizer

Die Nulldurchgänge des Versorgungsnetzes, werden als Interrupts mit fallenden Flanken eines GPIO Pins wahrgenommen. Dies wird in dem vereinfachten Ablaufdiagramm dargestellt:

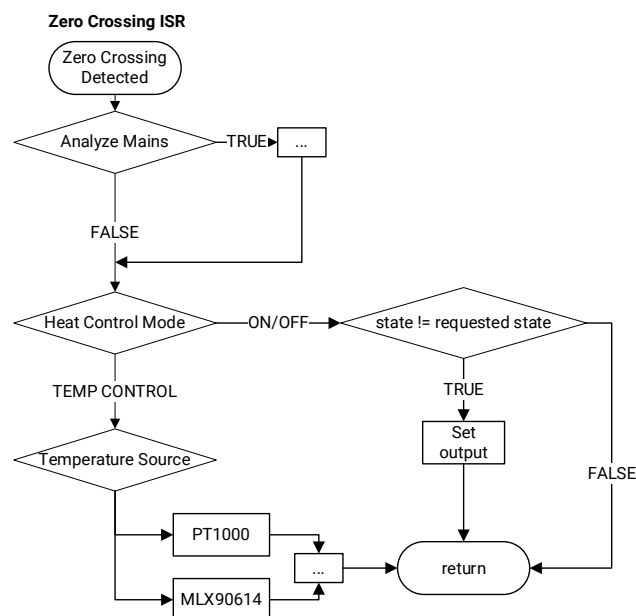


Abbildung 4-20: Ablauf Diagramm – Heater Control

Nur in diesem Interrupt kann der Heizer ein- und ausgeschaltet werden. Es wurden Regler Modi vorbereitet, welche aber während der finalen Messreihen nicht verwendet werden. Wahlweise kann der PT1000 Sensor über das Heizelement oder das IR Thermometer als Temperaturquelle gewählt werden. Die Temperatur kann auch in dem GUI selektiert werden für einen Ein Punkt Regler:



Abbildung 4-21: Temperatur Regler Auswahl Menü

Eine Pulspaketregelung wurde auch vorbereitet, jedoch nicht eingesetzt. Da weitere Überlegungen in dieser Arbeit nicht im Detail betrachtet werden (ideales Strahlungsspektrum, Ideale Eindringtiefe der IR Strahlung beim Toastbrot, Zusammenhang zwischen Glühtemperatur und Strahlungsspektrum, Auswirkung einer On Off Regelung auf diese Eigenschaften) stellt sich die Frage, ob und in welcher Art eine Regelung bei einem Toaster sinnvoll genutzt werden kann.

Eine weitere vorstellbare Funktion des ZCD wäre, mit dem Messsystem einen Stromausfall zu detektieren und das System für den Ausfall vorzubereiten (Daten speichern, Files Schließen).

4.7 PC-Software

Die PC-Software ist ein in C und C++ geschriebenes Command Line Tool. Abbildung 4-22 zeigt die möglichen Parameter mit denen das Programm gestartet werden kann, um verschiedene Aufgaben zu erledigen.

```
***** Toaster Tool *****
* arg 1: mode arg2: first idx, arg3: second idx
*
* modes:
* l - receive live data
* p - plot data
* x - plot and write raw data to csv
* y - plot and write converted data to csv
* z - save plots to png
* g - write raw data to csv
* d - download data
* a - live predict
* c - write classificatin information to file
*****
```

Abbildung 4-22: Toaster Tool Programm Funktionalität

In dem x_platform.h und x_platform.c werden die Funktionen, die für die Konvertierung der Sensordaten benötigt werden, neben den verwendeten Typdefinitionen, ebenfalls zwischen dem Mikrocontroller und dem Programm geteilt.

Zum Interpretieren der Logfiles kann ein Pointer des `100ms_log_entry_t` Types entlang des Files geschoben werden und damit Zugriff auf jeden Messwert ermöglichen. Die mit 10ms gesampelten Sensoren, werden down gesampelt auf 100ms und im gleichen Schritt erfolgt die Konvertierung für die graphische Darstellung.

Die Visualisierung der Daten erfolgt mit GNU Plot. Dieses Open Source Plot Programm wird über POSIX Pipe aus C instruiert. Der Vorteil dieses Interfaces ist es, dass die Daten im gleichen Format zum Plotten übergeben werden können, welches auch zum Schreiben ins CSV File benötigt wird. Da das Abspeichern als CSV für die Weiterverarbeitung in Python sowieso notwendig ist, muss nur mehr der File Descriptor des Files gegen den Pipe Descriptor von GNU Plot ausgetauscht werden.

Das Tool kann ebenfalls eine Verbindung zum Toaster aufbauen. Die Daten können während dem Toastvorgang alle x Sekunden aktualisiert dargestellt werden, die Messdaten werden akkumuliert und das Wx Widgets Terminal, welches GNU Plot verwendet, wird aktualisiert. Weiters können die Daten direkt vom Toaster heruntergeladen werden. Die Notwendigkeit ergibt sich dadurch, dass die SD-Karte im Toasterinneren unzugänglich platziert wurde. Die Klassifikationsinformationen sowie der Live Predictor werden nachfolgend erläutert.

5 Wie Fragt man die Maschine ob der Toast fertig ist?

5.1 Aufnehmen der Daten

Zum Aufnehmen der Daten musste entschieden werden wie die Bewertung so stattfindet, dass daraus später eine Maschine lernen kann. Es wurde ein System gewählt, in welchem nach dem Toastvorgang eine von drei möglichen Benotungen vom Benutzer vergeben werden kann.

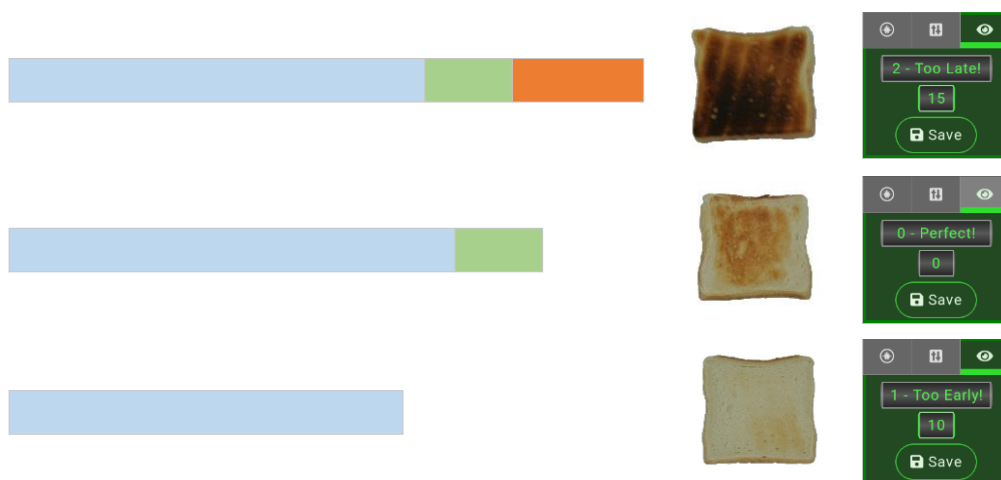


Abbildung 5-1: Bewertung

Die Einteilung erfolgt in drei Kategorien, für welche die folgenden Abkürzungen für die restliche Arbeit eingeführt werden:

- 1 – Too Early – Zu Früh - **TE**
- 2 – Too Late – Zu Spät - **TL**
- 0 – Perfect – Perfekt - **P**

Für **TE** und **TL** wird von dem Bediener eine zusätzliche Angabe benötigt. Es soll in 5 Sekunden Schritten ausgewählt werden, wie viele Sekunden später oder früher der Toast ideal (**P**) gewesen wäre. Das Messsystem hat für die nächste Messungsdauer die mitgegebene Sekundenzahl berücksichtigt und diese entsprechend kürzer oder länger gewählt. So konnte auch vermieden werden, dass Toasts während der Messaufnahmen unnötig lang getoastet wurden. Zusätzliche Metadaten sind auch mitgespeichert worden, wie beispielsweise die Anzahl an Toasts seit dem Start des Gerätes. Die Klassifikation dieser Messung beinhaltet die folgende Information:

Tabelle 4: Metadaten Einer Messung Teil 1

RATING	RATING_STR	RATING_SECS	DURATION	T_SINCE_START
2	TOO_LATE	15	1360	1

Die Bewertung des Anwenders war TL und es wurde geschätzt, dass 15 Sekunden früher der Toast Klasse P gewesen wäre. Insgesamt dauerte die Messung 136 Sekunden und es war die erste Messung seit Betriebsbeginn.

Tabelle 5: Metadaten Einer Messung Teil 2

NAU_UP	NAU_DOWN	U_TEMP_C	U_HEATER_MODE	U_TEMP_S	U_UCODE
-166782	-12988	190	0	1	16574

Die Initialwerte der Waage im angehobenen Zustand war `NAU_UP` und im heruntergefahrenen Zustand der Hebelmechanik und ohne Toast `NAU_DOWN`. Es wurde kein Temperaturcontroller verwendet.

Die Daten werden als Rohdaten ohne jegliche Konversion, mit der gleichen Sampling Rate von $100ms$ als CSV exportiert. Hierzu müssen die mit $10ms$ gesampelten Kanäle down gesampelt werden. Zusätzlich wird in dem exportierten CSV zu jeder Zeile die Information eingefügt, ob eine der drei Kategorien gerade an der Messung bis zu der jeweiligen Zeile zutrifft. Ein Beispiel eines CSVs zeigen die Tabelle 6 bis Tabelle 8. Die letzte Spalte von Tabelle 8 zeigt die eingefügte Klassifikationsinformation bei einem Übergang von TE zu P. Das Zeitfenster für die Klasse P wird mit 5 Sekunden definiert.

Tabelle 6: Rohdaten einer Messung Teil 1

Time/100ms	SHT_T	SHT_H	MLX_TA	MLX_TO	PT1000A	PT1000B	FREQ0	FREQ1	FREQ2
100500	54780	11915	24838	15865	3505	5673	299	6494	9043
100600	54810	11853	24844	15865	3507	5675	300	6495	9043
100700	54810	11833	24851	15865	3506	5674	300	6495	9043
100800	54810	11915	24851	15865	3502	5672	299	6496	9043
100900	54823	11875	24857	15865	3509	5685	300	6495	9043
101000	54823	11875	24863	15865	3508	5688	299	6495	9043
101100	54823	11833	24870	15865	3507	5689	302	6495	9043
101200	54836	11875	24870	15866	3501	5682	298	6495	9043
101300	54850	11834	24874	15866	3511	5698	299	6495	9042
101400	54879	11811	24881	15866	3515	5702	297	6495	9042
101500	54892	11811	24881	15866	3517	5705	298	6495	9041
101600	54866	11811	24889	15867	3511	5694	299	6495	9041
101700	54922	11837	24897	15867	3521	5711	299	6494	9041
101800	54892	11856	24904	15867	3524	5715	299	6494	9042

Tabelle 7: Rohdaten einer Messung Teil 2

Time/100ms	COL_R	COL_G	COL_B	COL_C	TVOC	eCO2	rawH2	rawEthanol	Weight
100500	1956	1040	855	3184	373	1212	13085	19143	212429
100600	1956	1040	855	3184	353	1283	13083	19145	212146
100700	1953	1037	853	3177	360	1258	13086	19137	211799
100800	1953	1037	853	3177	353	1249	13090	19144	211714
100900	1953	1037	853	3177	347	1250	13075	19155	211583
101000	1953	1037	853	3177	362	1249	13087	19145	211596
101100	1953	1037	853	3177	364	1244	13083	19138	211569
101200	1953	1037	853	3177	361	1249	13088	19132	211402
101300	1949	1033	850	3167	363	1243	13077	19141	211381
101400	1949	1033	850	3167	383	1242	13082	19147	211270
101500	1949	1033	850	3167	374	1251	13086	19129	211045
101600	1949	1033	850	3167	348	1259	13084	19139	211148
101700	1949	1033	850	3167	357	1242	13078	19166	210809
101800	1949	1033	850	3167	353	1257	13081	19156	211204

Tabelle 8: Rohdaten einer Messung Teil 3

Time/100ms	A0	A1	A2	A3	A4	ABS_HUMI	Class.
100500	13041	13888	10780	14617	13161	12555	1
100600	13040	13888	10779	14616	13158	12506	1
100700	13039	13887	10774	14617	13161	12485	1
100800	13039	13887	10774	14620	13163	12571	1
100900	13039	13888	10770	14617	13163	12537	1
101000	13038	13886	10767	14617	13164	12537	1
101100	13039	13888	10766	14615	13161	12492	0
101200	13038	13886	10767	14615	13161	12544	0
101300	13041	13887	10762	14617	13159	12509	0
101400	13039	13886	10762	14616	13160	12500	0
101500	13037	13886	10759	14619	13163	12508	0
101600	13034	13886	10756	14618	13162	12493	0
101700	13031	13888	10755	14617	13162	12551	0
101800	13034	13886	10750	14618	13159	12555	0

Während der Messungen wurde der Toaster nach allen 5 - 10 Toasts mit zwei Ventilatoren auf Anfangstemperatur abgekühlt (Abbildung 5-2). Einer wird durch einen USB-Anschluss auf einem optionalen Ausgang des Toasters betrieben. Der Zweite wird direkt vom Stromnetz versorgt.



Abbildung 5-2: Messsystem wird auf Anfangstemperatur gekühlt

Der Gedanke zum Kühlen des Systems auf Anfangstemperatur war, den realen Einsatz eines Toasters nachzuempfinden. Weiters konnte beobachtet werden, dass sich die Toastzeiten bei dem thermisch eingeschwungenen System nicht mehr deutlich ändern. Es sind über 300 Toastbrote des Typs „Butter Toast“ (Lieken Brot und Backwaren GmbH - Golden Toast) getoastet worden. Die Abbildung 5-3 zeigt die Messdaten aus einer Messung.

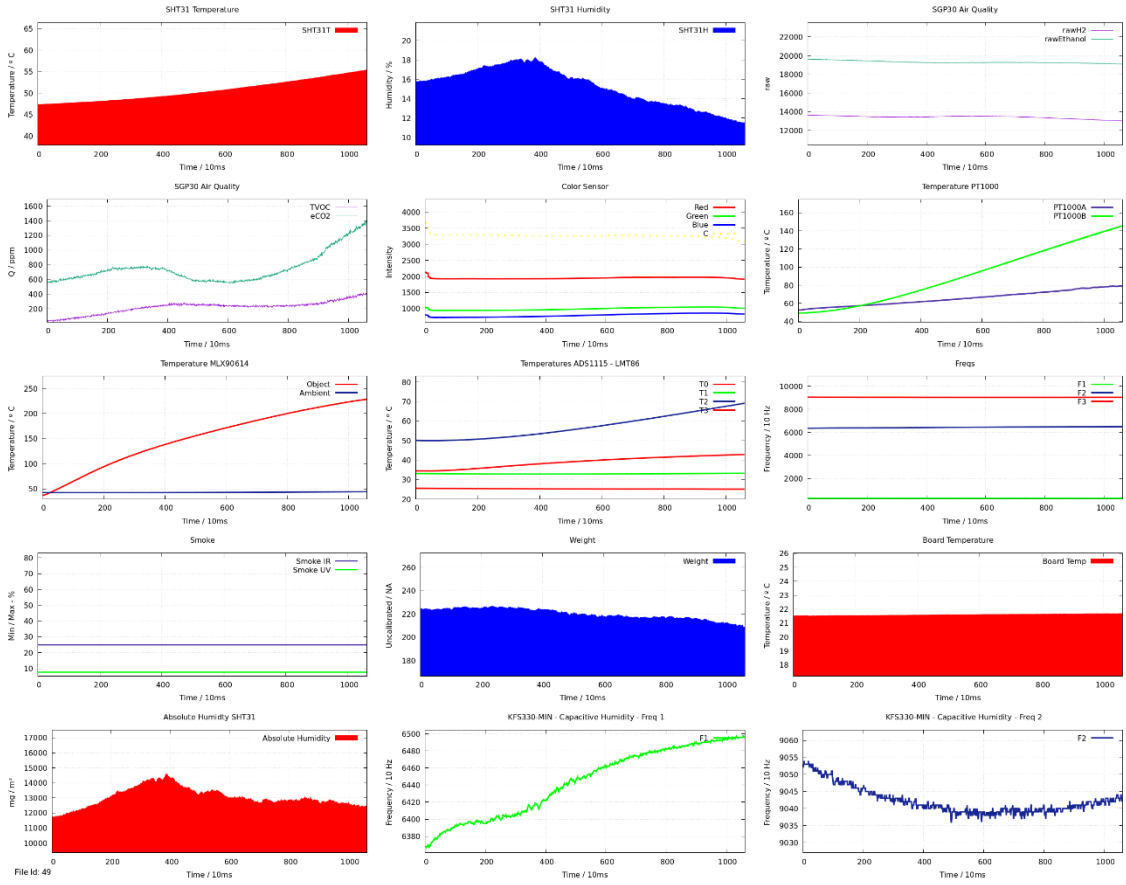


Abbildung 5-3: Messung Nr. 49

5.2 Preprocessing

Die Vorverarbeitung erfolgt mit einem Python Script. Es werden alle vom Toaster Tool exportierten CSV Files eingelesen und je Kanal wird der kleinste und größte Wert bestimmt. Die Konvertierung wird für einen Wert eines Kanals nach dem folgenden Schema [15] durchgeführt. In dem Vektor $\vec{x}$ liegen die Werte von einem Sensor vor, die normalisiert werden sollen. Die Berechnung für das Element x_i erfolgt nach Formel (33) und (34) wodurch der skalierte Wert x_{scaled} entsteht.

$$x_{std} = \frac{x_i - \min(\vec{x})}{\max(\vec{x}) - \min(\vec{x})} \quad (33)$$

$$x_{scaled} = x_{std} \cdot (\maxRange - \minRange) + \minRange \quad (34)$$

Wobei für *minRange* 0 und für *maxRange* 1 verwendet wird. Mit dieser Vorgehensweise werden die Messdaten konvertiert.

Eine Messung die mit TL bewertet wurde, beinhaltet auch Informationen über das Zeitfenster, in denen der Toast noch einem Zustand der Klasse P oder TE entsprach (Abbildung 5-1). Diese Aufsplittung der Messungen ist möglich, durch die eingegebene Informationen der menschlichen Bewertung. Setzt man diese Überlegung auf die Daten um, erhält man multiple generierte Messungen aus einer ursprünglichen Aufnahme. Abbildung 5-4 zeigt den Ablauf einer Messung mit dem Ergebnis TL

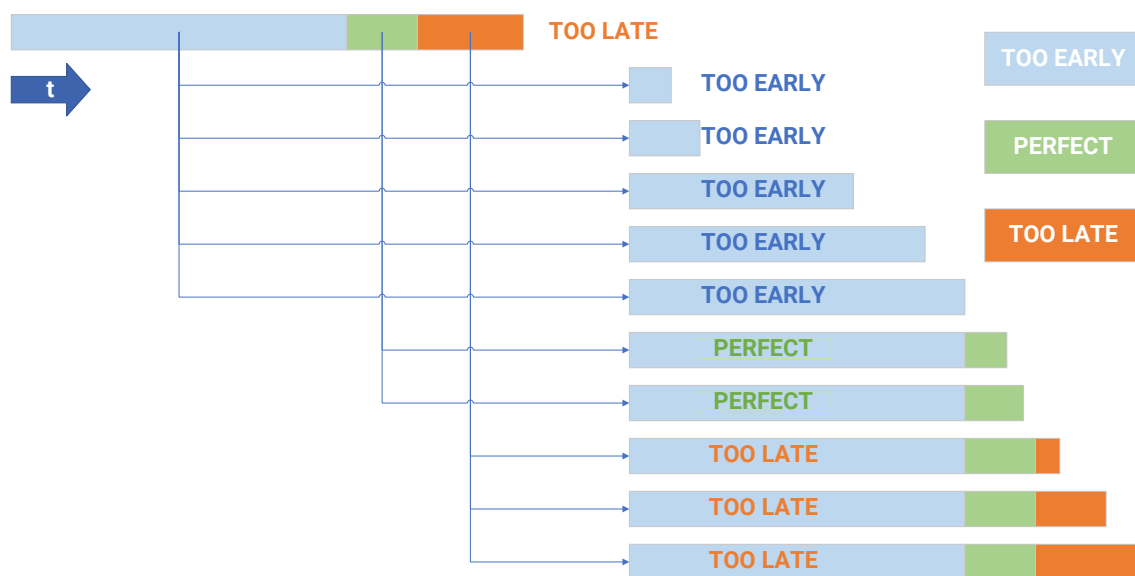


Abbildung 5-4: Verfahren zur Generierung von Messdaten.

Jede generierte Messung bekommt eine Klassifizierung nach TE / P / TL. Dieses Verfahren ähnelt dem, was bei der Objekterkennung bei Bildern in der Literatur, als Data Augmentation [63] genannt wird. Hier wird zwar nicht das Bild transformiert, gespiegelt oder verschoben, womit ein Objekt an verschiedenen Stellen des Bildes auch erkannt wird, jedoch erfüllt dies den gleichen Zweck und multipliziert die Anzahl an Trainingsdaten.

Die Länge der Datensätze ist abhängig von der Methode der Messungen und der Art der Generierung. Damit diese eine einheitliche Matrix- oder Arrayform erhalten, um mit diesen die gleichen Operationen durchführen zu können, werden sie mit dem Anfügen von Nullen (Zero Padding) auf die gleiche Länge gebracht. Es wird eine Länge gewählt die größer ist wie die längste Messung, womit alle Messungen einheitlich 155 Sekunden lang werden.

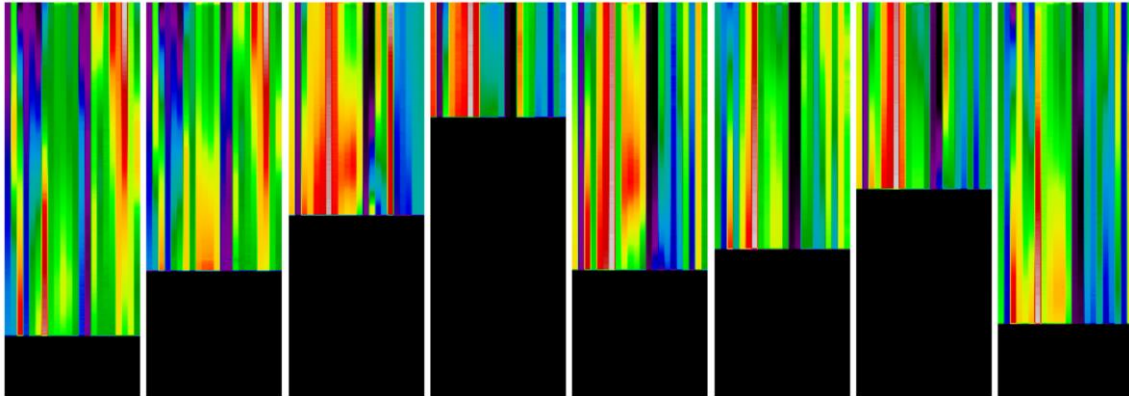


Abbildung 5-5: Sensordaten- Darstellung in der Bild Domain

Wenn man sich die 22 Spalten an Sensordaten nach dem Zeropadding als einzelne Bilder vorstellen möchte, kann die Abbildung 5-5 erstellt werden. Dieses Bild ist was die Maschine aus den Daten sieht.

5.2.1 Random Slicing

Die erste Implementierung der Methode generiert aus jeder TL Messung zusätzlich 3 TE, 3 TL und eine P Messung. Für die generierte Messung werden innerhalb der Zeitfenster prozentuale Grenzen bestimmt, beispielsweise zwischen 5% und 95% des Zeitfensters. Bei P Messungen wird innerhalb der Mitte des Zeitfensters (Verteilung um 25% der Gesamtlänge) gesucht. Innerhalb der so gelegten Grenzen werden die Längen für die generierte Messung zufällig gewählt. Die Anzahl der generierten Messungen ist nicht abhängig von der Länge des Zeitfensters des jeweiligen Zustandes (TL, TE).

Mit dieser Vorverarbeitungsmethode werden Klassifikationsgenauigkeiten von 91% bis 94% Genauigkeit erreicht, jedoch bleibt bei Mehrklassenklassifizierung die Genauigkeit für die Erkennung von P und TL Klassen unter 85% Genauigkeit.

Tabelle 9: Binäre Klassifikation Auswahl aus den getesteten Modellen

Klasse	Generierte Messungen
TE	1115
TL	620
P	416

5.2.2 Time Slicing

Bei dieser Methode werden die zusätzlichen Messungen in einheitlichen Zeitabständen (z.B. 100ms) aus einer bestehenden Messung generiert. Diese Umsetzung führt zu zahlreichen

Messdaten, jedoch wird durch den Ursprung dieser Daten (Toaster verbringt viel mehr Zeit im TE Zustand wie in P oder TL) eine starke Imbalance zwischen den Klassen eingeführt. Bei den Testdaten leisteten die Modelle eine hohe Gesamtgenauigkeit von über 97%, die Klassen P und TL werden nur mit weniger als 80% Genauigkeit erkannt.

Tabelle 10: Binäre Klassifikation Auswahl aus den getesteten Modellen

Klasse	Generierte Messungen
TE	25833
TL	1800
P	832

5.2.3 Random Slicing With Padding

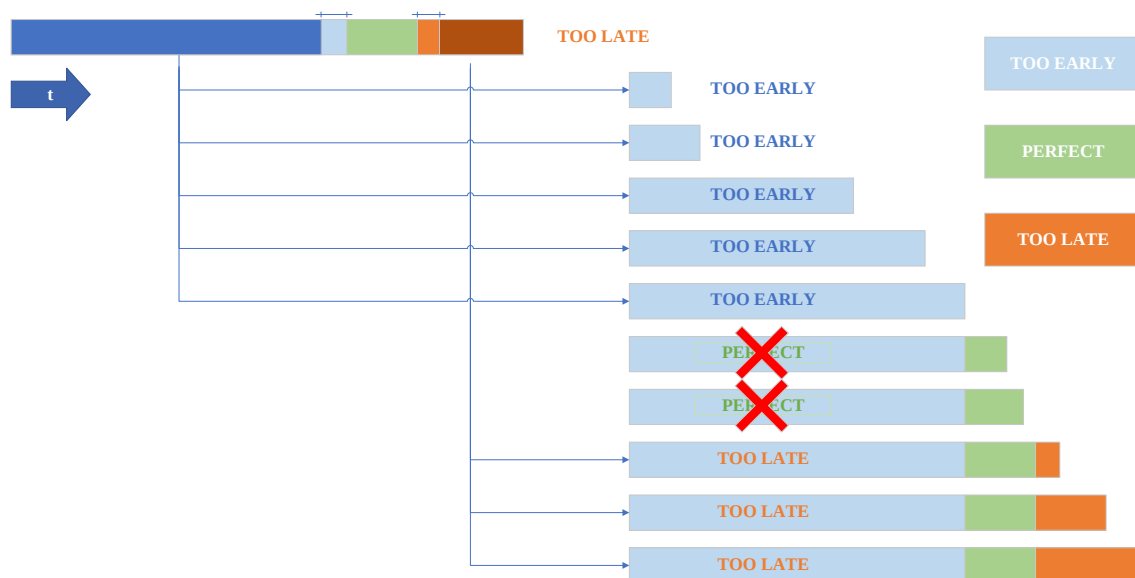


Abbildung 5-6: Optimiertes Verfahren zur Generierung von Messdaten.

Diese Methode Abbildung 5-6 versucht die Unsicherheit zu mindern, welche durch den menschlichen Anwender bei der Bewertung der Toasts entsteht. Die Idee dabei ist, bei den TL Messungen für die P Zeitfenster keine zusätzlichen Messungen zu generieren. Weiters wird ein pro Klassifikationsfall definierter Sicherheitsabstand zu den Fenstergrenzen eingeführt. Dies bedeutet, dass die generierten TE Messungen nicht bis zu der Grenze des P Fensters wahrgenommen werden, sondern 25 Zeitschritte davor aufhören. Bei den P Messungen werden nur die ersten 3 Sekunden des Perfekt Zeitfensters zur Generierung von weiteren Messungen verwendet. Die gleichen Maßnahmen werden auch für die Generierung von TL Messungen verwendet, hier wird auch ein Abstand zu dem Ende der Perfekt Messungen eingeführt. Die Anzahl der generierten Messungen wird dynamisch nach Länge des jeweiligen Fensters bestimmt. Lange TE und TL Phasen in einer Messung führen also zu mehr generierten Messungen.

Tabelle 11: Binäre Klassifikation Auswahl aus den getesteten Modellen

Klasse	Generierte Messungen
TE	1744
TL	728
P	728

Dieses Verfahren führt zu den besten Ergebnissen bei den aufgenommenen Daten, welche folgend präsentiert werden.

5.3 Modellauswahl

Es gibt viele Möglichkeiten, die Ergebnisse des Trainingsverfahrens durch Optimierung von Hyperparameter zu verbessern, wie

- Batch Größe (Wie viele Messdaten für einen Iterationsschritt – Epoch mitbeachtet werden sollen)
- Lernrate (Gesamt, sowie Regulatoren für einzelnen Layer)
- Die Gewichtung der Klassen durch die angepassten Fehlerfunktionen verändern, um beispielsweise Imbalancen zwischen den Klassen entgegenzuwirken.
- Anzahl der Trainingsepochen

Parameter der einzelne Layer sind neben den betreffenden Bedingungen für das gesamte Training mögliche Justierungsoptionen. Die verwendeten Kernel Größen und die Erzeugung von N dimensionaler räumlicher Zusammenhänge sind ebenfalls Mittel der Justierung. Es wurden zahlreiche Netzstrukturen und Hyperparameter Kombinationen ausprobiert, mit dem Einsatz von Preprocessing, beschrieben in 5.2.1 und 5.2.2. Es wurde auch versucht, angepasste Fehlerfunktionen zu verwenden, die eine Gewichtung der Klassen TL und P anheben.

Diese Versuche sind in der digitalen Anlage ersichtlich. Jedoch hat die Optimierung von diesem Parameter und der Fehlerfunktion nur sehr wenig Unterschied zu den Ergebnissen mit den vorliegenden Daten bewirkt. Die Veränderung der Vorverarbeitung wie in 5.2.3. beschrieben und auf die Daten des Toasters umgesetzt, hat zu einer Verbesserung der Ergebnisse um 4-10% an Genauigkeit gesorgt. Es werden keine weiteren Vergleiche zwischen den erarbeiteten Vorverarbeitungsmethoden angeführt und auch keine Optimierungsvergleiche der Hyperparameter.

Das Training der neuronalen Netze erfolgt mit dem Tensorflow Backend unter Keras. Die Netze verwenden Binary sowie Categorical Cross Entropy als Fehlerfunktion, sowie adam [19] als Optimierer. Eine weitere Methode, Early Stopping [71] wird verwendet um das Training abubrechen, wenn der Fehler sich zwischen einer gewählte Anzahl an Iterationen nicht mehr signifikant ändert. Dieses Delta an Änderung ist ein weiterer Parameter dieser Methode.

Die Parameter der Operationen / Layer die das Netz bestimmen, sowie die Veränderung der Dimensionalität zwischen den einzelnen Schritten wird in den Abbildung 5-8 und Abbildung 5-9 erfasst. Die Modelle behalten die Namen, die in der digitalen Anlage verwendet werden, damit eine Zuordnung zum Source-Code möglich ist. Alle verwendeten Netze bestehen aus den in Kapitel 2 vorgestellten Operationsblöcken, wie Konvolution, Aktivierung, Dropout und Fully Connected Layer. Der Layer Flatten bewirkt, dass N dimensionale Arrays zu 1D Arrays geformt werden. Es werden auch rechnerisch aufwändigere Netzarchitekturen getestet wie Squeeze and Excitation (Anhang A5 – Modell B) nach [62] und Inception [11] Blöcke ähnelnde Parallele Convolutional Netze (Anhang A5 – Modell C und Modell D). Die zeigen ebenfalls gute Ergebnisse, wie die folgend vorgestellte aus wenigen Layern bestehenden einfacheren Netzen.

5.3.1 Binäre Klassifikation

Aus dem Sichtpunkt des Toasters bestehen nach dem Start eines Toastvorgangs nur zwei mögliche Entscheidungen: weiter rösten oder aufhören. Setzt man diesen Gedanken bei den aufgenommenen Daten um, können sie in eine binäre Klassifikationsform gebracht werden. Jede generierte Messung, die mit P oder TL bezeichnet wurde, wird als Stop gekennzeichnet und die Messungen mit dem Label zu früh werden Continue genannt. Durch die Zusammenführung von beiden Klassen ist eine nahezu 50% - 50% Verteilung zwischen den beiden Klassen im Datensatz entstanden. Das Training erfolgte mit Stratified KFold Cross Validation mit vier Splits. Die letzte Aktivierung des Netzes bleibt bei allen Versuchen gleich. Hier wird die sigmoide Funktion verwendet, damit ein Wert zwischen null und eins ausgegeben wird. Die zwei Grenzwerte sind auch die zwei zu identifizierenden Klassen:

- 0 – STOP
- 1 – CONTINUE.

Es werden vier der versuchten Netzarchitekturen gezeigt Abbildung 5-7. Modell A verwendet nur Fully Connected und Dropout Layer. Um die Menge der Daten zu reduzieren, wird an dem ersten Layer ein Maxpooling Layer eingesetzt, mit einer Pool Size von 2. Dies reduziert die Eingangsdaten um die Hälfte. Modell AL ist ein Versuch, um zu sehen ob wesentlich weniger Parameter zu einer deutlichen Verschlechterung des Ergebnisses führen. Modell C sowie Modell TB verwenden Convolutional Layer vor Fully Connected Layer. Mit Operationen der Konvolution werden nicht nur Filter erzeugt, es wird auch die anfängliche Länge von 1550 Zellen und Breite von 22 Spalten Schicht für Schicht reduziert. TB weist einen Convolutional Layer auf, der auch die in Abbildung 2-9 gezeigte Dilation in dem Filter umsetzt. Die Ergebnisse sind an einem separierten Testdatensatz ermittelt worden, welcher aus 272 Beispielen besteht.

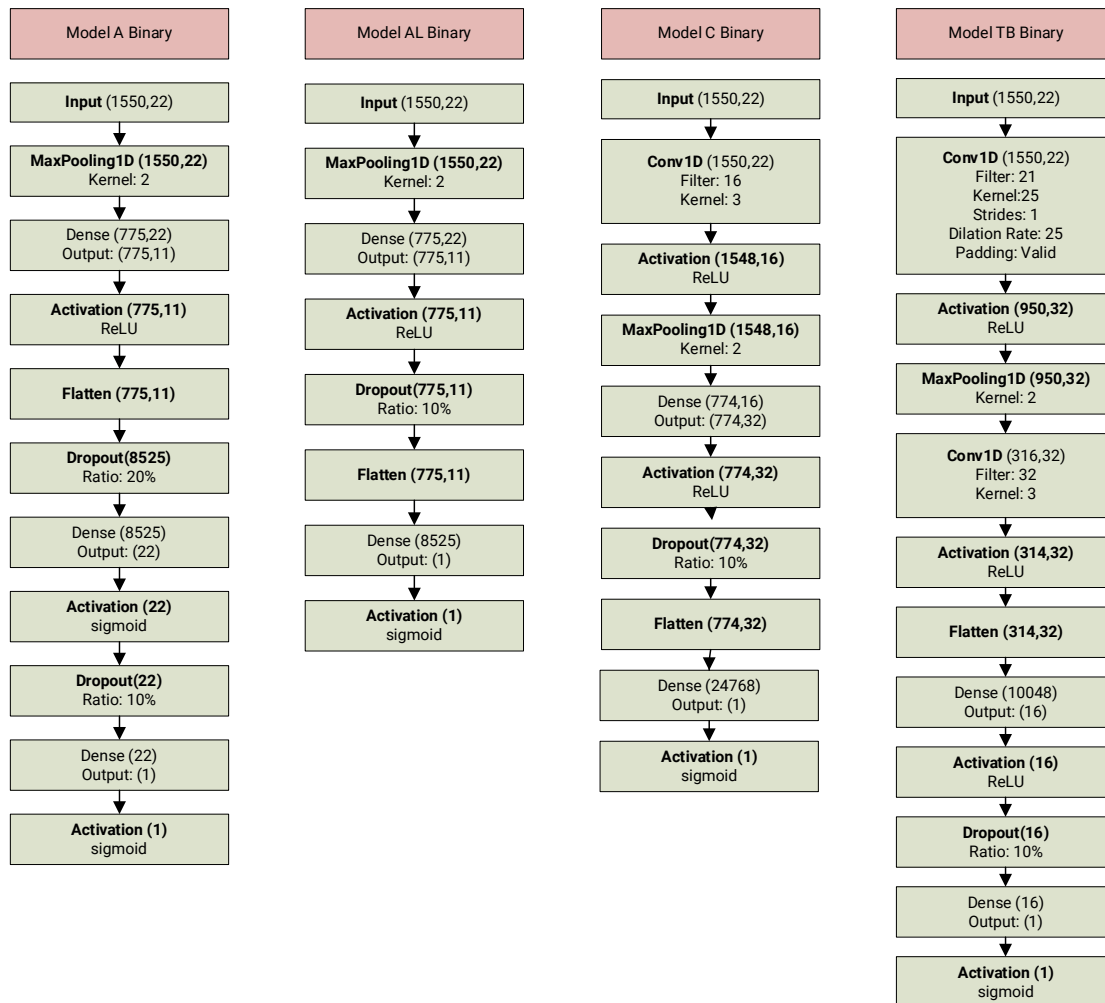


Abbildung 5-7: Modelle für binäre Klassifikation

Die Tabelle zeigt die Anzahl an Parametern, Netztypen und die ermittelten Genauigkeiten:

Tabelle 12: Binäre Klassifikation Auswahl aus den getesteten Modellen

Modell	Parameter	Typ	Accuracy - Test	Accuracy - Validation
A	187848	NN	0,97	0.99
AL	8779	NN	0.98	0.98
C	26385	CNN	0.97	0.99
TB	181537	CNN	0.97	0.98

Die Tabellen (Tabelle 13, Tabelle 14, Tabelle 15, Tabelle 16) beinhalten die Testergebnisse der gezeigten Modelle A, AL, C, und TB im Detail.

Tabelle 13: Ergebnisse Modell A

	Präzision	Recall	F1-Score	Support
STOP	0.93	1.00	0.96	121
CONTINUE	1.00	0.94	0.97	151
macro avg.	0.97	0.97	0.97	272
weighted avg.	0.97	0.97	0.97	272

Tabelle 14: Ergebnisse Modell AL

	Präzision	Recall	F1-Score	Support
STOP	0.96	0.99	0.98	121
CONTINUE	0.99	0.97	0.98	151
macro avg.	0.98	0.98	0.98	272
weighted avg.	0.98	0.98	0.98	272

Tabelle 15: Ergebnisse Modell C

	Präzision	Recall	F1-Score	Support
STOP	0.94	0.98	0.96	121
CONTINUE	0.99	0.95	0.97	151
macro avg.	0.97	0.97	0.97	272
weighted avg.	0.97	0.97	0.97	272

Tabelle 16: Ergebnisse Modell TB

	Präzision	Recall	F1-Score	Support
STOP	0.95	1.00	0.97	121
CONTINUE	1.00	0.95	0.98	151
macro avg	0.97	0.98	0.97	272
weighted avg	0.98	0.97	0.97	272

Von Modell C wird die Verteilung der Vorhersagen bei den unaugmented (keine generierte Messung beinhaltende) Datensätze getestet. Das Ergebnis zeigt die Abbildung 5-8:

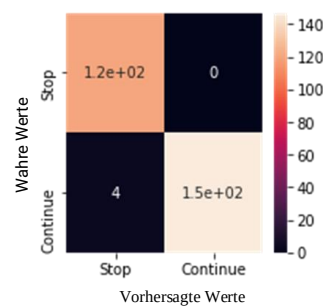


Abbildung 5-8: Confusion Matrix – Binäre Klassifikation

Alle Testfälle mit dem Label Stop werden richtig erkannt. Vier Testfälle die eigentlich als Continue erkannt werden hätten sollen, wurden falsch zugeordnet. Mehrklassen Klassifizierung

Die Abbildung 5-9 zeigt einen Teil der evaluierten Netzwerke für Klassifikation nach den drei Klassen. Als Gemeinsamkeit verwenden alle Netze als letzte Aktivierung die Softmax Funktion und verfügen über drei Ausgangsneuronen, damit die Ausgabe zu den drei erwarteten Klassen zugeordnet werden kann. Die Tabelle zeigt die Anzahl an Optimierungsparametern je Netz, die verwendete Arten an Operationen und den Typ des Netzes.

Tabelle 17: Multiclass - Auswahl aus den getesteten Modellen

Modell	Parameter	1D/2D	Typ
A	259003	2D	CNN
B	276499	2D	CNN
C	238627	2D	CNN
E	1689667	2D	CNN
F	724355	2D	CNN
G	2837891	2D	CNN
H	174627	1D	CNN
I	28041	1D	NN
J	122513	1D	CNN
K	231363	1D	CNN
T	95683	1D	CNN

Im Vergleich zu den Modellen der binären Klassifikation werden hier auch Operationen in zwei Dimensionen durchgeführt.

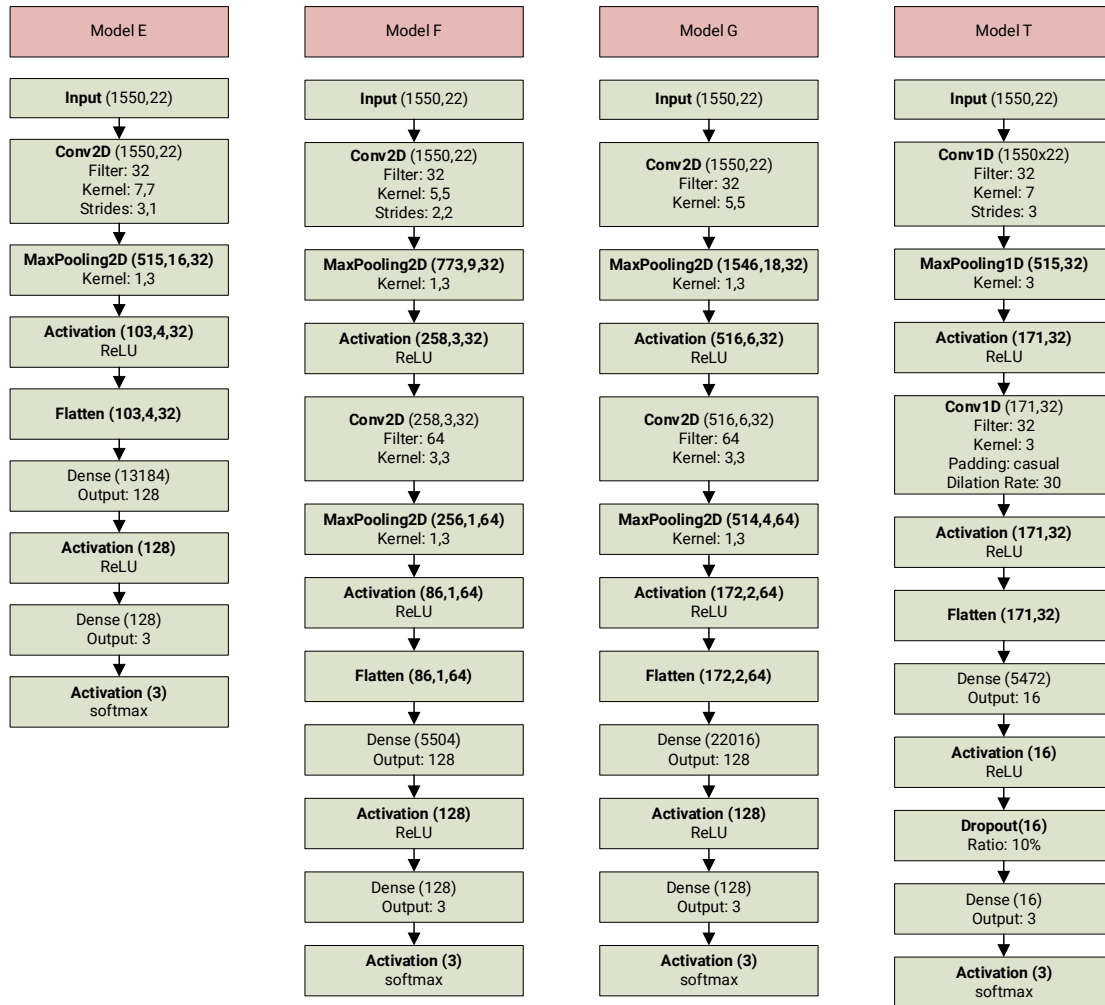


Abbildung 5-9: Modelle für Klassifikation nach den drei Klassen

Um mehrere Modelle und Parameter automatisiert zu evaluieren, wird ein Skript erstellt. Es können die Funktionspointer zu den einzelnen Modellerstellungsfunktionen eingetragen werden und die Modelle werden trainiert sowie gegen einen ausgewählten Datensatz geprüft. Die Evaluierung erfolgt in Repeated K Fold mit 3 Folds und zwei Wiederholungen. Die Metriken werden nach jedem Durchlauf errechnet und in Tabellen zusammengefasst.

Tabelle 18 zeigt die Ergebnisse mit dem in Abschnitt 0 beschriebenen Vorverarbeitungsverfahren.

Tabelle 18: Vergleich der Ergebnisse – Modelle für Klassifikation nach drei Klassen – Random Slicing

	Genau- igkeit Val. AVG.	Genau- igkeit Val. MAX.	Genau- igkeit Test AVG.	Genau- igkeit Test MAX.	Präzi- sion Val. MAX.	Präzi- sion Test AVG.	Präzi- sion Test MAX.	Recall Test MAX.	Recall Test AVG.	Test F1 AVG.	Test F1 MAX.
A	0.882	0.893	0.862	0.899	0.894	0.865	0.888	0.846	0.898	0.85	0.888
B	0.882	0.894	0.878	0.911	0.894	0.885	0.92	0.863	0.91	0.87	0.914
C	0.891	0.902	0.87	0.888	0.902	0.867	0.879	0.853	0.871	0.85	0.872
D	0.784	0.9	0.769	0.907	0.883	0.768	0.892	0.737	0.896	0.75	0.893
E	0.916	0.926	0.899	0.903	0.915	0.886	0.896	0.89	0.896	0.88	0.89
F	0.908	0.921	0.886	0.911	0.918	0.877	0.895	0.872	0.906	0.87	0.899
G	0.9	0.913	0.888	0.903	0.897	0.878	0.892	0.877	0.898	0.87	0.89
H	0.845	0.876	0.828	0.857	0.879	0.844	0.866	0.804	0.846	0.82	0.841
I	0.907	0.923	0.884	0.903	0.914	0.875	0.895	0.873	0.898	0.87	0.891
J	0.895	0.909	0.89	0.907	0.907	0.884	0.907	0.879	0.896	0.88	0.898
K	0.861	0.873	0.846	0.868	0.883	0.861	0.894	0.82	0.843	0.84	0.867

Tabelle 19 beinhaltet die Ergebnisse zu den gleichen Modellen, mit der optimierten „Random Slicing with Padding“ Vorverarbeitung (Abschnitt 5.2.3). Es ist ersichtlich, dass alle Modelle bessere Ergebnisse liefern und die einzelnen Klassen besser unterschieden werden.

Tabelle 19: Vergleich der Ergebnisse – Modelle für Klassifikation nach drei Klassen –Random Slicing with Padding

	Genau- igkeit Val. AVG.	Genau- igkeit Val. MAX.	Genau- igkeit Test AVG.	Genau- igkeit Test MAX.	Präzi- sion Val. MAX.	Präzi- sion Test AVG.	Präzi- sion Test MAX.	Recall Test MAX.	Recall Test AVG.	Test F1 AVG.	Test F1 MAX.
A	0.956	0.962	0.975	0.978	0.955	0.974	0.974	0.97	0.975	0.972	0.973
B	0.959	0.968	0.977	0.985	0.963	0.974	0.981	0.971	0.979	0.972	0.98
C	0.967	0.972	0.974	0.985	0.965	0.973	0.981	0.97	0.983	0.971	0.982
E	0.966	0.971	0.98	0.989	0.966	0.979	0.986	0.975	0.985	0.977	0.985
F	0.955	0.962	0.979	0.982	0.952	0.974	0.977	0.974	0.977	0.973	0.976
G	0.97	0.978	0.988	0.996	0.974	0.987	0.995	0.986	0.998	0.986	0.996
H	0.951	0.952	0.974	0.985	0.941	0.972	0.983	0.969	0.98	0.97	0.982
I	0.911	0.938	0.94	0.974	0.922	0.942	0.968	0.931	0.973	0.936	0.969
J	0.937	0.945	0.974	0.974	0.936	0.971	0.974	0.965	0.966	0.967	0.968
K	0.959	0.963	0.973	0.982	0.953	0.975	0.979	0.964	0.975	0.969	0.977
T	0.954	0.961	0.978	0.985	0.953	0.976	0.981	0.972	0.979	0.973	0.98

5.3.1.1 Modell F

Am Beispiel des Modell F werden der Trainingsverlauf und die internen Zustände des trainierten Netzes dargestellt. Die Early Stopping Kondition tritt nach 100 Epochen nicht ein, der Fehler zeigt weiterhin die Tendenz, dass es weiter minimiert werden kann und das Training wird durch Anzahl

der hinterlegten maximalen Epochen eingestellt. in Abbildung 5-10 ist ersichtlich wie sich der Fehler durch die Iterationen mindert und die Genauigkeit der Prädiktionen ansteigt.

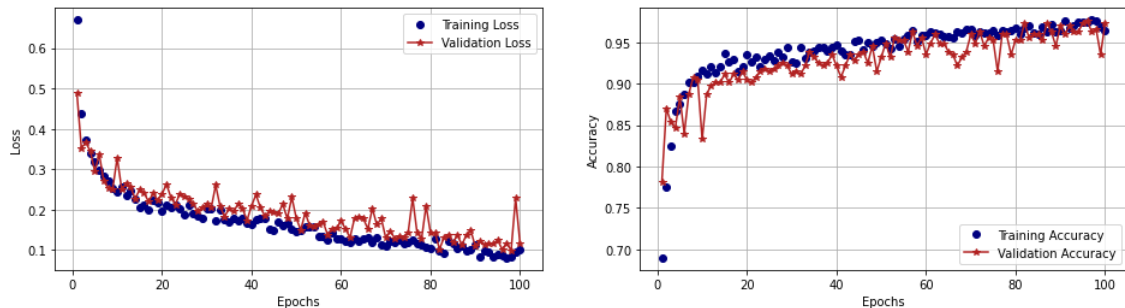


Abbildung 5-10: Trainingsverlauf Modell F

Nach dem Training des Netzes wird das Modell mit dem Testdatensatz evaluiert. Die Ergebnisse werden in der Tabelle 20 und in der Confusion Matrix Abbildung 5-11 zusammengefasst:

Tabelle 20: Ergebnisse Modell F

	Präzision	Recall	F1-Score	Support
Perfect	0.96	0.97	0.96	67
Too Early	0.98	1.00	0.99	151
Too Late	1.00	0.93	0.96	54
macro avg	0.98	0.97	0.97	272
weighted avg	0.98	0.98	0.98	272

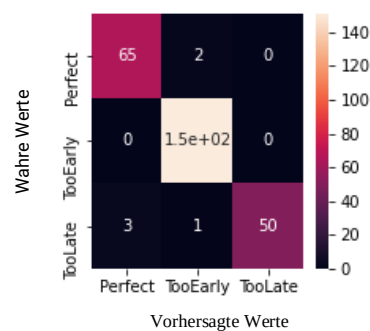


Abbildung 5-11: Modell F Confusion Matrix

Um einen Einblick ins Innere des Netzes zu gewinnen, werden Ausgangswerte der einzelnen Layers dargestellt. Hierzu wird eine Funktion geschrieben, welche diese Werte als Bild darstellt. Es werden 2D und auch 1D Layer visualisiert. Beim 1D Layer werden so lange 0 Werte am Ende des Arrays eingefügt bis eine Darstellung möglich wird. (Bsp. Das letzte Layer besteht aus 4 Pixel, in der Realität sind es 3).

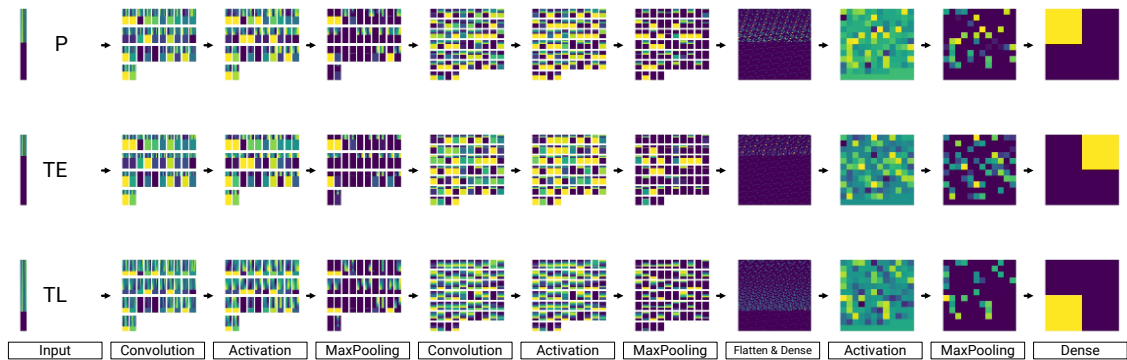


Abbildung 5-12: Modell F – Visualisierung Layer Outputs

Am Eingang des Netzes werden drei verschiedene Testfälle (P, TE, TL) zugeführt (Abbildung 5-12). Wie in der Abbildung ersichtlich ist, aktiviert sich immer ein anderer Ausgangs-Node, wenn eine abweichende Prädiktion ausgegeben wird. Bei den Convolutional Layer kann gut beobachtet werden, dass unterschiedliche Filter zu verschiedenen Eingangsdaten aktiviert werden.

5.4 Relevanz einzelner Sensoren

Um die Relevanz der Sensoren zu bestimmen wird ein Modell ausgewählt, welches bei variieren-der Veränderung der Eingangsdaten trainiert und evaluiert.

Das ausgewählte Modell verwendet 1D Operationen, damit die Anzahl an Sensordatenkanäle un-abhängig verändert werden kann und eine Auswertung trotzdem möglich ist. Bei 2D Operationen entstehen ebenfalls Zusammenhänge in der Achsrichtung b (Abbildung 2-9) der Daten womit die Reihenfolge der Kanäle auch einen Einfluss auf das Ergebnis hat. Bei 22 Kanälen existieren ins-gesamt mehr als 4 Millionen mögliche Variationen ohne Wiederholung. Ist die Reihenfolge der Kanäle von Bedeutung ist diese Zahl größer als drei Sextillionen.

Damit die Rechenzeit im umsetzbaren Rahmen bleibt, wurde die Vorgehensweise angewendet, dass Testfälle definiert werden, in welchen immer einer der Sensoren eliminiert wird (mit allen Kanälen die der Sensor liefert). Zusätzlich wurden Testfälle definiert in denen nur eine ausge-wählte Kombination an Sensoren verwendet wird.

Es wurde ein Skript erstellt in welchem diese Testfälle eingetragen werden und hintereinander abgearbeitet werden. Das Skript manipuliert die Sensordaten automatisiert und entfernt die ge-kennzeichneten Kanäle vor dem Training. Jede Auswahl wird insgesamt sechsmal evaluiert, durch zweifache Wiederholung von drei Splits. Die Metriken, Genauigkeit, Präzision, Recall und F1 Score werden an den Validierungsdaten ausgewertet. Tabelle 21 präsentiert die zusammenge-fassten Ergebnisse der Durchläufe.

Tabelle 21: Vergleich der Ergebnisse – Relevanz einzelner Sensoren

	Genau- igkeit AVG.	Genau- igkeit MAX.	PREC. AVG.	PREC. MAX.	RECALL. AVG.	RECALL. MAX.	F1 AVG.	F1 MAX.
Alle Sensoren	0.951	0.968	0.938	0.960	0.943	0.963	0.940	0.961
Kein SHT31 (Rohrsensor Temp. und Humid.)	0.947	0.960	0.937	0.952	0.939	0.953	0.937	0.952
Kein MLX (Passive IR und Ambient Temp.)	0.943	0.956	0.928	0.943	0.936	0.947	0.931	0.945
Kein PT1000A	0.950	0.955	0.939	0.949	0.943	0.950	0.940	0.946
Kein PT1000B	0.950	0.967	0.938	0.961	0.941	0.960	0.939	0.961
Keine HT Luftfeuchtig- keitssensoren	0.949	0.963	0.937	0.954	0.941	0.960	0.938	0.956
Kein Farbsensor	0.952	0.965	0.937	0.954	0.946	0.960	0.941	0.957
Kein SGP30 (Rohrsensor Ethanol und H2)	0.946	0.958	0.934	0.949	0.937	0.952	0.935	0.947
Keine Waage	0.952	0.965	0.940	0.955	0.945	0.959	0.942	0.956
Keine Temp. A0	0.950	0.972	0.939	0.962	0.943	0.971	0.940	0.966
Keine Temp. A1	0.951	0.971	0.939	0.963	0.944	0.967	0.941	0.965
Keine Temp. A2	0.950	0.959	0.937	0.949	0.942	0.958	0.939	0.952
Keine Temp. A3	0.952	0.969	0.941	0.961	0.947	0.970	0.943	0.965
SELECTION_A	0.952	0.959	0.942	0.950	0.942	0.952	0.941	0.950
SELECTION_B	0.950	0.958	0.937	0.948	0.944	0.951	0.940	0.949
SELECTION_C	0.944	0.953	0.934	0.947	0.935	0.954	0.934	0.949
SELECTION_D	0.951	0.963	0.939	0.952	0.944	0.958	0.941	0.955
SELECTION_E	0.919	0.939	0.906	0.930	0.898	0.926	0.901	0.927
SELECTION_F	0.947	0.960	0.937	0.952	0.942	0.957	0.939	0.954
SELECTION_G	0.916	0.922	0.899	0.906	0.902	0.904	0.898	0.903
SELECTION_H	0.934	0.947	0.921	0.932	0.923	0.939	0.921	0.935
Alle Sensoren - Zweiter Durchlauf	0.944	0.954	0.933	0.944	0.935	0.950	0.933	0.946
Nur Farbsensor	0.911	0.925	0.894	0.915	0.901	0.916	0.897	0.915
Nur MLX* (Passive IR und Ambient Temp)	0.923	0.932	0.908	0.915	0.912	0.919	0.909	0.914
Nur PT1000A	0.874	0.890	0.851	0.868	0.846	0.859	0.848	0.863
Nur PT1000B	0.914	0.920	0.895	0.907	0.899	0.906	0.895	0.905
SELECTION_S	0.948	0.954	0.936	0.951	0.941	0.952	0.938	0.946

Die Selektionen bestehen aus den folgenden Sensoren:

- **A:** Rohrsensor, PT1000s, HTA und HTB Luftfeuchtigkeit, Temperatur Signale A0-A3
- **B:** Rohrsensor (ohne eCO₂, Abs. Humidität und TVOC Kanal), PT1000s, HTA und HTB Luftfeuchtigkeit, Temperatur Signale A0-A3
- **C:** IR Thermometer, Farbsensor, Waage Temperatur Signale A0-A3
- **D:** Alle Sensoren (Rohrsensor ohne eCO₂, Abs. Humidität und TVOC Kanal)
- **E:** PT1000s, HTA und HTB Luftfeuchtigkeit
- **F:** Rohrsensor ohne Abs. Humidität
- **G:** PT1000s und der SHT31 aus dem Rohrsensor (Luftfeuchtigkeit und Temperatur)
- **H:** IR Thermometer und B + G Kanäle des Farbsensors
- **S:** Alle Sensoren außer die folgenden Kanäle:
 - Temperatur Signale A0, A1, A3
 - alle Kanäle des Farbsensors
 - Waage

Bei dem Versuch wird ersichtlich, dass viele Kombinationen sehr ähnliche Ergebnisse liefern. Die Waage und der Farbsensor führen nicht zu einer deutlichen Verbesserung des Ergebnisses, jedoch kann ein Netz, welches anhand der Daten eines Farbsensors trainiert wurde, genauere Vorhersagen treffen als die Daten eines einzigen Temperatursensors.

Die Daten des PT1000 Sensors über dem Heizelement sind relevanter als die Daten des PT1000 Sensors im Flansch.

Wird ausschließlich der Rohrsensor verwendet, kann trotzdem eine maximale Genauigkeit von 96.0% erreicht werden. Der Vergleich zwischen A und B zeigt, dass die kalkulierten Kanäle des Gassensors keine redundanten Informationen beinhalten, sondern zum besseren Ergebnis beitragen können.

Für Selektion **S** werden die Kanäle entnommen, die einzeln zur Verbesserung der Ergebnisse führen (**Kein A0, A1 und A3** liefern bessere Ergebnisse als **alle Sensoren**, auch **keine Waage** sowie **kein Farbsensor** Testfall zeigen eine Verbesserung (Testfall - Tabelle 21)). Der Test **S** zeigt, dass es keine Verbesserung im Vergleich zu **allen Sensoren** bringt, was die Aussagekraft dieses Vergleiches in Frage stellt. Es sollten unterschiedliche Modelle mit mehr Testfällen ausgewertet werden, um weitere Aussagen zu der Relevanz der Sensoren mit dieser Methode zu erhalten. Anhang A3 zeigen die Ergebnisse von einem zweiten Durchlauf des Tests.

5.5 Random Forests

Random Forest [73] ist ein verbreitetes Machine Learning Verfahren für Klassifikationsaufgaben. Es werden zwei Random Forests Modelle aufgestellt und mit einem CNN gegengeprüft. Für beide Fällen werden identische Trainings- und Testdaten verwendet. Die Anzahl an Ästimatoren bestimmt die Anzahl an parallel eingesetzten Entscheidungsbäumen. Abbildung 5-13 zeigt einen Entscheidungsbaum des ersten Modells mit der Tiefe von fünf.

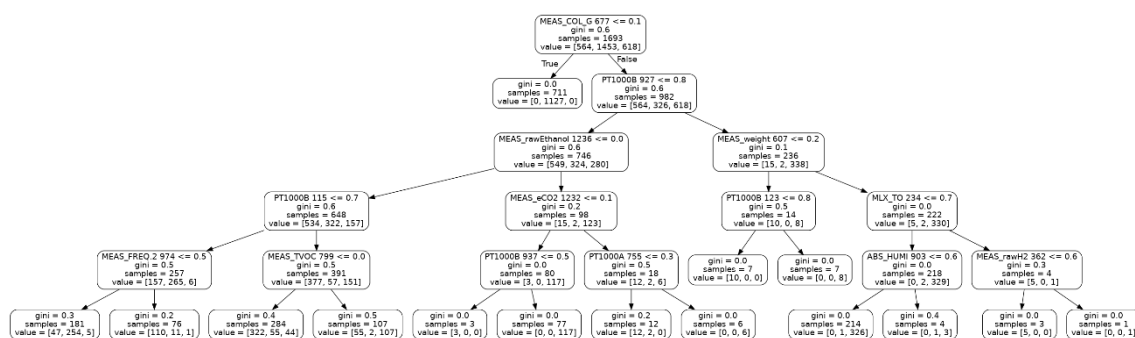


Abbildung 5-13: Random Forests Modell F16, eines der Entscheidungsbäume

Die Parameter und Ergebnisse der Random Forest Klassifizierers werden in der Tabelle zusammengefasst.

Tabelle 22: Random Forests Modelle

Modellname	Ästimatoren	Tiefe (MAX.)	Genauigkeit AVG. Validierung	Genauigkeit Test
F16	16	5	0.932	0.945
F32	32	50	0.989	0.993

An dem gleichen Datensatz wird das Modell GB (Anhang A5) 75 Epochen lang trainiert. Das CNN Modell erreicht ebenfalls eine Genauigkeit von 98.9% an dem Validierungsdatensatz und 99.3% bei dem Testdatensatz. Die Confusion Matrizen der drei Modelle (I. – CNN GB, II. RF32, III. RF 16) werden nebeneinander dargestellt.

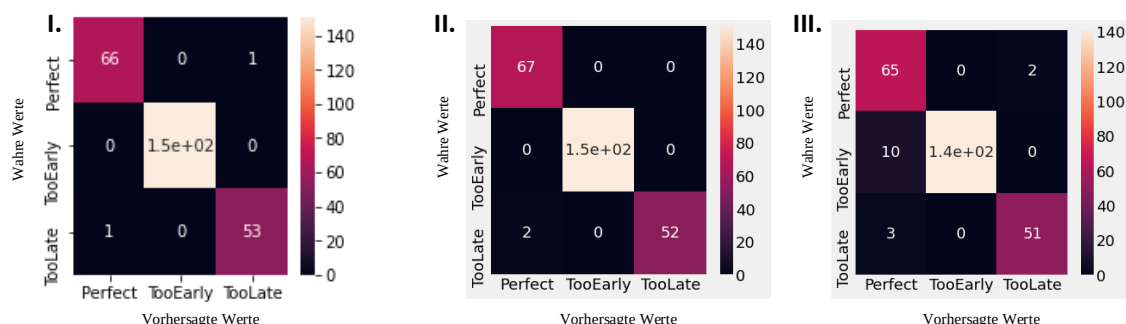


Abbildung 5-14: Confusion Matrix der Modelle GB (I.), RF32(II.) und RF16 (III.)

Für das Random Forests Verfahren existiert eine etablierte Methode [61] (Mean Decrease of Impurity) um die Relevanz der Eingangsvariablen zu analysieren. Diese Methode ist in scikit-learn implementiert [60] und wird auf beiden Random Forests Modellen (RF16 und RF32) ausgeführt. Die Ergebnisse für die ersten 50 Variablen sind im Anhang A4 ersichtlich. Die Ergebnisse zeigen bei beiden Modellen, dass die Daten in der Zeitspanne 67s – 69s eine wichtige Rolle für die Entscheidungen haben. Das Signal des analogen Temperatursensors A3 und des PT1000B tragen viel zum Ergebnis bei. Der Farbsensor sowie der Rohrsensor bei dem RF32 Modell haben einen großen Anteil an den Vorhersagen.

5.6 Toast Decoder

Inspiziert vom neuronalen Netzwerktypen Generative Adversarial Networks [59] und Autoencoder im Anwendungsbereich von Bildsynthese sowie Bildkompression [58] wurde ein Toast Decoder erstellt. Für dieses Experiment liefern die vorliegenden Sensordaten den Input des Netzwerkes und die Bilder von den fertigen Toasts das am Output des Netzes erwünschte Ergebnis.

5.6.1 Preprocessing

Die ersten 246 Toasts wurden in einem Fotozelt unter konstanten Beleuchtungsbedingungen fotografiert. Jedes Bild beinhaltet 49 Toasts. Diese werden mehrmals beidseitig fotografiert. Zu den einzelnen Bildern wird vermerkt zu welcher Messnummer sie erfasst worden sind.

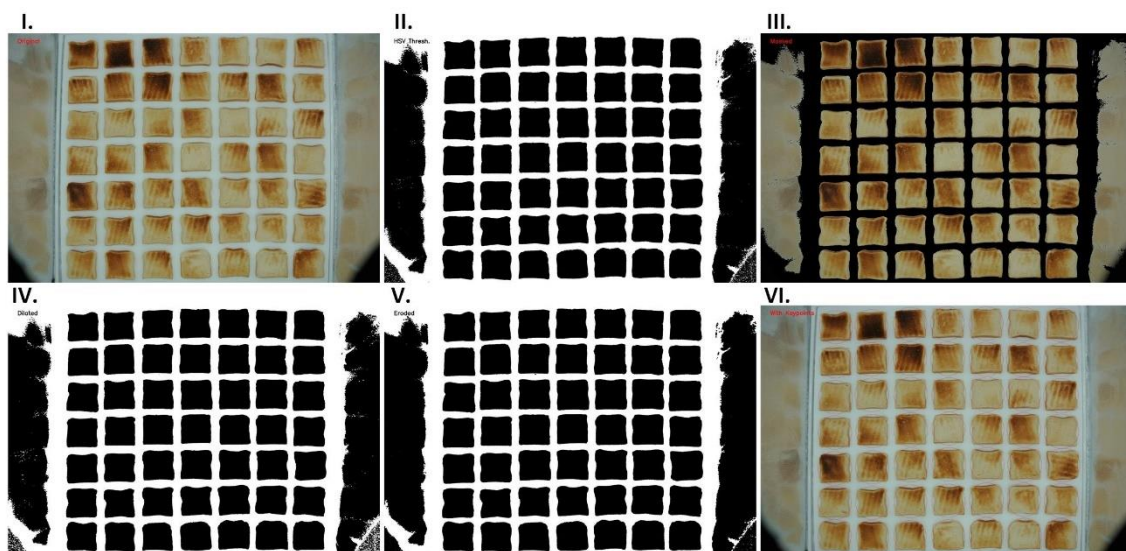


Abbildung 5-15: Bilverarbeitungsschritte der Toastbilder

Die Bilder werden von einem Skript automatisiert geladen und die folgenden Bildverarbeitungsschritte werden mit OpenCV durchgeführt:

- II. Durch HSV-Schwellwert Bildung die einzelnen Toasts segmentieren
- III. Den segmentierten Bereich maskieren
- IV. und V. Durch Dilation und Erosion die Fläche schließen
- VI. Schlüsselpunkte der einzelnen Toasts durch Binary Blob Detektion bestimmen.
- VII. Um die Schlüsselpunkten ein Rechteck bestimmen und dieses als einzelne Toastbilder exportieren.

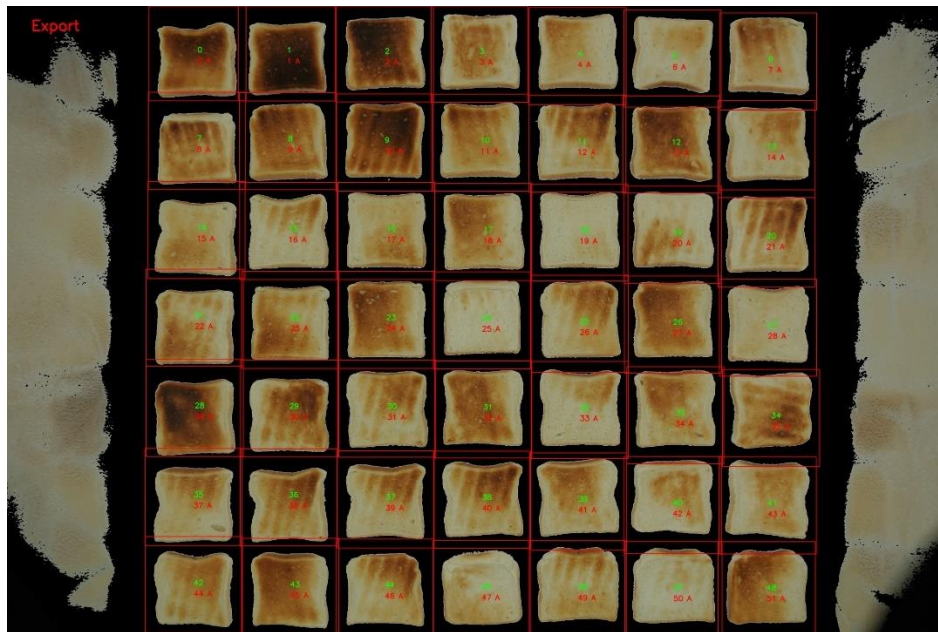


Abbildung 5-16: Zuordnung der Binary Blobs zu Messungen und Brot Seiten

Anschließend werden die Bilder einzeln mit einem Filename, welcher die Zuordnung zu der ursprünglichen Messung ermöglicht, gespeichert.

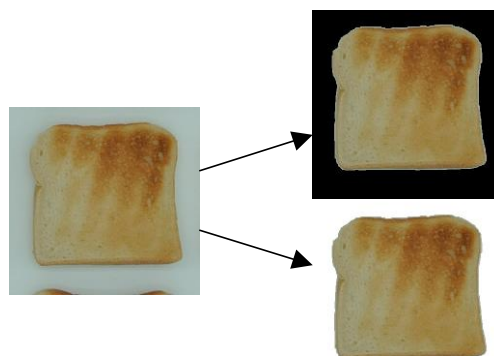


Abbildung 5-17: Nachbearbeitung einzelner Toastbilder um zwei Datensätze mit schwarzem und weißem Hintergrund zu erhalten

Von einem zweiten Skript können die Einzelbilder von dem Hintergrund befreit und in einen schwarzen und weißen Hintergrund exportiert werden (Abbildung 5-17). So entstehen insgesamt 1519 Bilder, welche 245 Messungen zugeordnet sind. Das Vorhandensein mehrerer Bilder, die eine etwas andere Scharfstellung sowie Positionierung aufweisen (es wurde kein Stativ für die Aufnahmen verwendet) bereichert die Anzahl an Trainingsdaten.

5.6.2 Modell

Es wird ein CNN aufgebaut welches am Ausgang 64x64x3 Werte liefert, die einem Bild entsprechen sollen. Am Eingang liegen die Messdaten wie bei der Klassifizierung vor. Der Fehler wird als MSE zurückgeführt. Das Ziel ist die dimensional Unterschiede durch das Netz zu überbrücken:

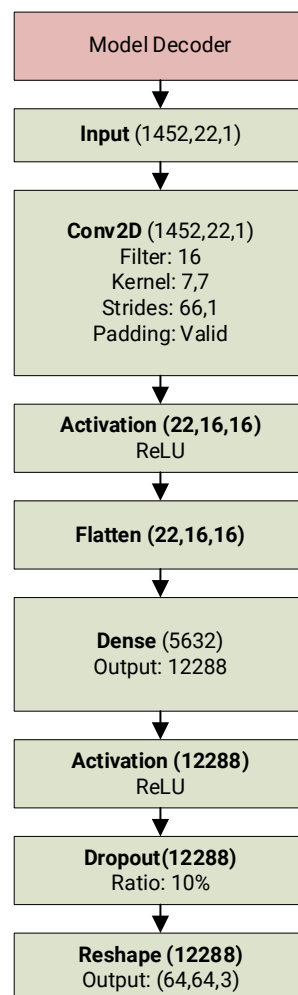


Abbildung 5-18: CNN – Toast Decoder

Das Modell weist mehr als 68 Millionen Parameter auf, die während dem Training optimiert werden. Die Vorhersagen des Netzes nach dem Training Abbildung 5-19 zeigen eindeutig dunklere Bilder bei P und TL eingestuften Messdaten wie bei TE Messungen. Es ist ersichtlich, dass die Streifenmuster die durch den Toaster verursacht werden von dem Netz ebenfalls reproduziert werden.



Abbildung 5-19: Vorhersagen des Toastdecoders in Vergleich zu tatsächlichen Brotzustände

Es wird ein auch zweites Netz getestet, bei welchem Upsampling Layer verwendet werden und welches nur über 49352 Parameter verfügt. Das Ergebnis ist in Anhang A ersichtlich.

5.7 Live Predictor

Um die entwickelten Netze im Echtbetrieb testen zu können, wurde ein Live Predictor entwickelt. Die PC-Software (Toaster Tool) war bereits in der Lage, durch die vorgestellte Funktion – Live Data - die Sensordaten während der Messung zu empfangen. Weiters kann das Toaster Tool die Daten im CSV Format an einen File Deskriptor schreiben. Das Training der neuronalen Netze erfolgt in Python Umgebung und zwischen den beiden Welten soll eine Verbindung hergestellt werden. Vom Toaster Tool werden zwei Named Pipes (Unidirektionale FIFOs) angelegt in welchen die Kommunikation zwischen der C++ Anwendung und dem Python Skript stattfindet. Die NN Modelle können nach dem Training in Tensorflow mit den optimierten Gewichten gespeichert und exportiert werden. Das Python Skript wird von der C++ Anwendung in einem neuen Konsolenfenster gestartet und schreibt seine Prozess ID in ein separates File. Anhand dem PID kann die C++ Anwendung den Python Prozess beim Schließen der Anwendung ebenfalls schließen. Ist das Python Skript fertig mit dem Laden des Modells gibt dieser Auskunft an die C++ Anwendung über seinen Pipe, dass er bereit ist Daten zu empfangen. Die C++ Anwendung empfängt die vom Toaster als UDP Multicast Nachrichten versendete 100ms Log Einträge. In diesem Fall wird die Funktion,

welche Daten ins CSV schreibt oder über GNU Plot darstellt an einen neuen File Deskriptor gerichtet (File Deskriptor der Pipes zum Python Skript). Die Sensordaten werden geparkt als wären diese Zeilen eines CSV Files und an die Python Anwendung gesendet.

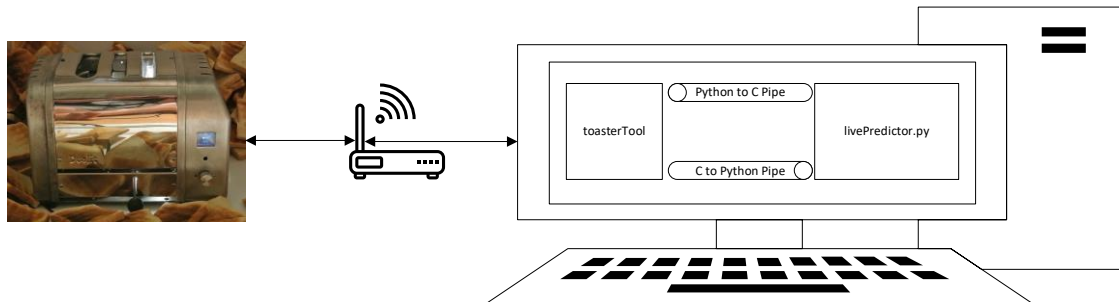


Abbildung 5-20: Live Predictor

Neben dem Modell, welches mit den trainierten Gewichten gespeichert wird, wird die Skalierung der Daten während dem Preprocessing der Sensordaten gespeichert. Der Live Predictor akkumuliert diese Zeilen an Sensordaten und erstellt für jede Vorhersage ein Null-gepaddeten, skalierten Datensatz und führt diesen dem Netz zu. Durch die Pipe wird der Predictor von der C++ Anwendung zurückgesetzt, wenn vom Toaster länger als 1 Sekunde keine Daten mehr empfangen werden. Aus den Versuchen des Kapitels ist bekannt, dass das System nicht immer mit 100% Genauigkeit entscheiden kann, ob die aktuell gezeigte Zeitscheibe dem wahren Zustand entspricht. Würde ein Mensch seinen Mitmenschen um die Uhrzeit fragen und eine offensichtlich falsche Antwort erhalten, würde dieser die Antwort ignorieren. Diese Überlegung wurde auf den LivePredictor so übertragen, dass TL Detektionen so lange übersprungen werden, bis wenigstens ein P erkannt wird. Wenn der Fragende nun weiß, dass sein Mitmensch zu 97% die richtige Uhrzeit sagt, kann seine Rückmeldung trotzdem mit Vorsicht wahrgenommen werden.

Dementsprechend gibt der Live Predictor nur dann den Stop, wenn wenigstens x aus P und TL Prädiktionen erfolgt sind. Diese Zahl x könnte auch als Bestimmungsmethode für den Bräunungsgrad verwendet werden, da die Tests mit dem Prädiktor zeigen, dass mehrere (ca. 20 – 30) P Meldungen vor dem ersten TL erfolgen. In diesem Sinne kann auch noch ein Modell, mit wesentlich mehr Ungenauigkeit, eine Verbesserung an Wiederholungsgenauigkeit des Toasters bringen. Das Stop Kommando wird vom Live Predictor an die C++ Anwendung gegeben, die an den Toaster eine UDP Nachricht adressiert. Die Messung wird gestoppt und der Toast wird ausgegeben. Der Source Code für den Python Teil Live Predictor ist im Anhang A angegeben.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung der Ergebnisse

Es wurde ein erfolgreiches modulares Messsystem geschaffen, um die notwendigen Sensordaten für die gezielte Analyse der Arbeit, einzusammeln. Das System hat störungslos über 300 Messungen aufgenommen und übertragen.

Dieses System, bestehend aus Software Komponenten, wurde aufgebaut, um die aufgenommenen Daten zu bearbeiten und Machine Learning Algorithmen wie Deep Learning oder Random Forest einzusetzen.

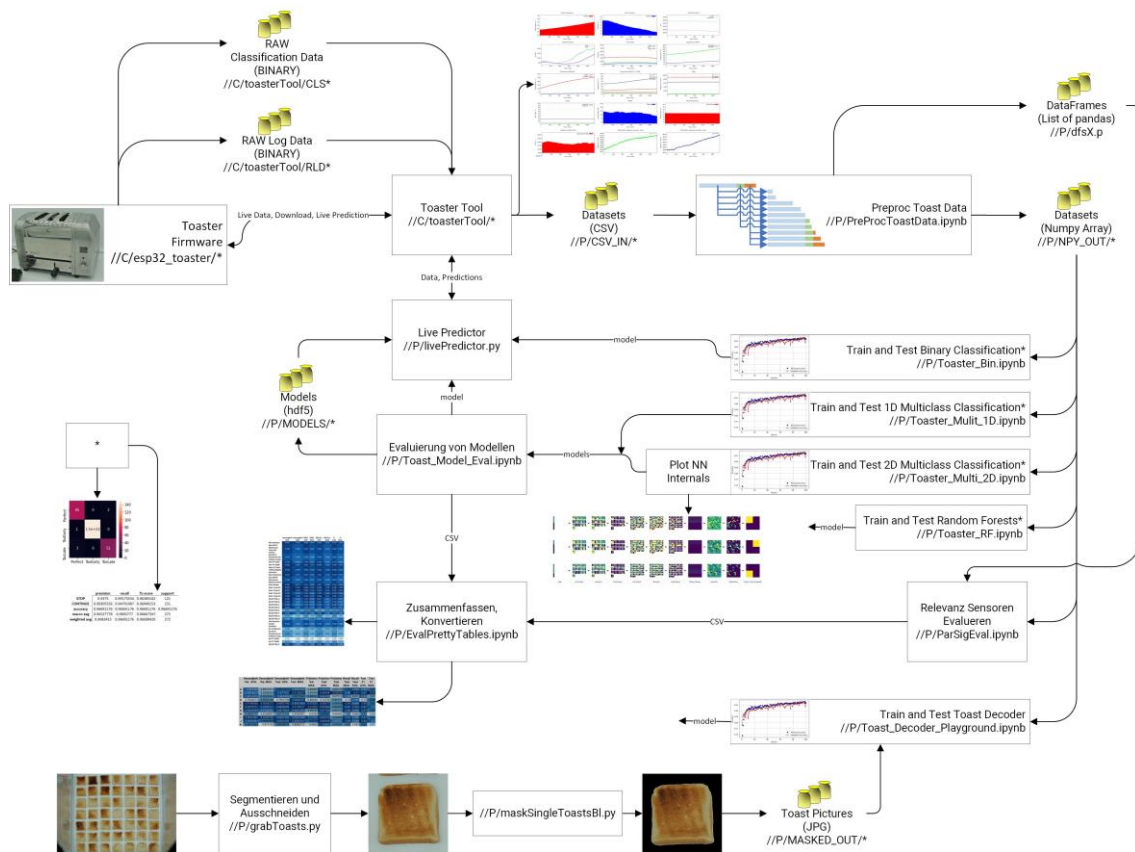


Abbildung 6-1: Messsystem

Jeder rechteckige Block (Abbildung 6-1) stellt ein Programm oder ein Skript dar. Die entstehenden Daten werden mit Icons gekennzeichnet. Die Namen der Skripte und Programmen werden mit Pfadangabe in der digitalen Anlage angeführt.

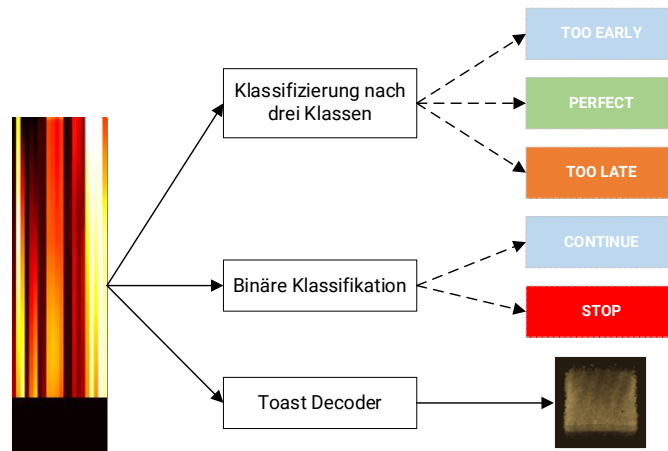


Abbildung 6-2: Aufgestellte Modelle für Prädiktionen basierend auf Sensordaten eines Toasters

Mit diesem System konnten zwei Arten von Klassifikationsmodellen Abbildung 6-2 sowie ein dritter Modelltyp für Generierung von Bildern aufgestellt werden. Die Modelle können trainiert, evaluiert und verglichen werden. Es sind Methoden für Bestimmung der Relevanz von Sensoren (Feature importance) getestet worden. Es wurde festgestellt, dass für eine Testgenauigkeit von 96% die Daten des Rohrsensor Moduls ausreichend wären. Weiterhin zeigten die Versuche, dass ein neurales Netz in Kombination mit jedem hier verwendeten Sensor funktionieren kann und aus alles lernen kann.

Es sind auch Modelle für das Klassifikationsproblem mit wenigen Parametern (Modell AL – 8998) mit guter Genauigkeit von 98% bei dem Testdatensatz getestet worden und auch weitere rechenaufwändigere Modelle mit noch besseren Ergebnissen. Das Klassifizierungsproblem kann auch durch Random Forests mit sehr guten Ergebnissen (99.3%) gelöst werden.

Ist es möglich die Wiederholungsgenauigkeit von Toaster durch Einsatz von neurale Netzwerken zu optimieren, wie der Titel verspricht? Um diese Frage nicht nur mit Messergebnissen an Testdaten zu beantworten, wurde der Live Predictor erstellt und der Prototyp wurde mit zwei auf dem Markt erhältlichen, modernen Toaster verglichen:

- BOSCH TAT6A001 (Langschlitztoaster mit Thermostat und Brotzentrierung)
- Breville / Sage Smart Toast STA825BAL2EEU1 (the Smart Toast, motorisierter zwei Scheiben Toaster mit Brotzentrierung)



Abbildung 6-3: Bosch, Sage/Breville, Prototyp mit LivePredictor auf Dualit Basis

Es werden hintereinander in jeder Maschine drei Toasts geröstet. Der Bosch Toaster wird auf Röststufe 2, der Breville / Sage Toaster wird für einen Durchlauf auf Stufe 2, bei dem zweiten Durchlauf auf Stufe 3 betrieben. Die Modelle T und F werden in den Live Predictor geladen. Abbildung 6-4 zeigt das Ergebnis des Tests:

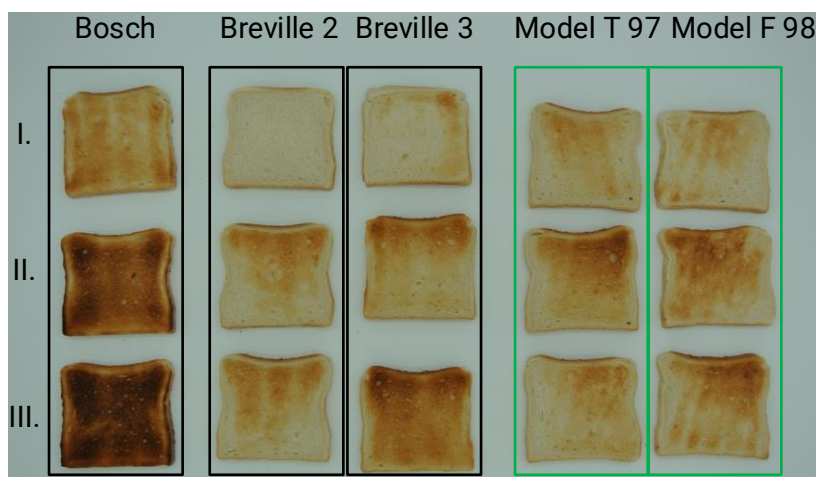


Abbildung 6-4: Test der Wiederholungsgenauigkeit

Alle Toaster mindern die Röstzeit nach dem ersten Toast:

Tabelle 23: Toastzeiten

Toast Nr.	Breville 2 / s	Breville 3 / s	Bosch / s	Modell T / s	Modell F /s
1	113	140	121	111	112
2	81	101	104	89	93
3	81	99	97	73	76

Der erste Toast. Bei der Einstellung wo der Sage den zweiten Toast ideal erstellt, bleibt der erste Toast ungeröstet. Der Live Predictor produziert ein Ergebnis welches für manche Konsumenten

zu hell ist, jedoch während dem Aufnehmen der Testdaten als Perfekt getitelt wurden, da diese die am Anfang festgelegte Definition erfüllen. Beim zweiten Toast verbrennt der Toast in dem Bosch Toaster obwohl die Röstzeit deutlich gekürzt wird. Der Breville sowie der Live Predictor liefern ein gutes Ergebnis. An der Stelle ist der Vorteil von der Brotzentrierung beider modernen Toaster gegenüber dem Dualit (Prototyp) ersichtlich, da sie ein wesentlich gleichmäßigeres Röstbild produzieren.

Anhand dieses Tests, basierend auf wenigen Aufnahmen kann gezeigt werden, dass neurale Netze eindeutig ein Optimierungspotential für die Wiederholungsgenauigkeit beinhalten. Zusammenfassend kann die Aussage getroffen werden, dass Machine Learning beim elektrischen Toaster zur Verbesserung gegenüber Toastzeitregelung von bestehenden Modellen führen kann.

6.2 Errata

Während der Fertigstellung der Arbeit wurde festgestellt, dass die Daten des Luftfeuchtigkeit Sensors im Flansch (F1) in den CSV Files Fehler beinhalten. Ein Programmfehler im Toaster Tool verursacht diese, während dem Down Sampeln dieser Daten von 100ms auf 10ms und betrifft immer die erste Messung in einer Messreihe. Die Netze dessen Ergebnisse hier angeführt worden sind, sind mit diesem Fehler in den Daten trainiert worden.

Der Farbsensor liefert in den ersten Sekunden einer Messung keine richtigen Werte, was am zu späten Einschalten der Beleuchtung und der zu langen Integrationszeit liegt. Der Fehler wurde erst entdeckt, als schon zahlreiche Messungen erfolgt sind, deswegen wurde das Verhalten nicht mehr angepasst. Die weiteren Daten des Sensors sind richtig und an dem Verlauf der RGB-Kanäle ist die Verdunkelung feststellbar.

Gelegentlich schaltet der Prototyp nur mit erneutem unter Strom setzen ein, dies wurde noch nicht näher analysiert. Mögliche Fehlerquellen sind Brown Out Detektion und die Dauer bis zum Erreichen der Versorgungsspannung.

Die in der digitalen Anlage inkludierten Schaltpläne zeigen oft nicht die finalen Werte für passive Bauteile wie Kondensatoren und Widerstände.

6.3 Ausblick

Eine Erweiterung der Messplattform wäre denkbar, welche den Test von Machine Learning Modellen im Prototyp ohne externe Abhängigkeiten (Server und WLAN-Verbindung) ermöglicht. Somit könnte ein Computing Node (z.B. Nvidia Jetson Nano) mit UART / RS485 Anbindung an das Mainboard des Toasters eingebaut werden. Der darauf befindliche Live Predictor sollte fernwartbar mit neuen Modellen versorgt werden können.

Käme es zur Anwendung von Machine Learning bei einem realen Toaster, muss die notwendige Hardware in dem Gerät für eine Klassifikation nicht vorhanden sein, da der Bediener in der Regel ein Smartphone besitzt welches genügend Rechenkapazität zur Verfügung stellt.

Stört den Anwender nicht, dass Sensordaten des Toasters seinen PAN / LAN verlassen, kann eine Lösung basierend auf einem Remote Server (Cloud) vorgestellt werden. Diese erhält die Daten von tausenden Toastern, wodurch sie mit Prädiktionen versorgt werden. Je mehr klassifizierte Daten vorliegen, desto weiter könnte das System trainiert und verbessert werden.

Die entstehenden Rechenaufwände bei der Anwendung eines neuronalen Netzes wurden in dieser Arbeit nicht behandelt (welche Rechenaufwände genau entstehen und wie sie sich Requantisieren lassen). Diese sind aber relevant für eine Umsetzung auf ressourcenarmen Plattformen wie Mikrocontroller. Die Evaluierung von Tensorflow Lite, CMSIS-NN oder Eigenimplementierungen für diesen Zweck könnten zukünftige Themenbereiche sein.

Es handelt sich hier vorerst um ein Proof Of Concept mit nur einem Brottyp. Es könnte untersucht werden, ob anhand den ersten 40-50 Sekunden des Toastvorgangs basierend auf Sensordaten mit Machine Learning feststellbar ist, welcher Brottyp gerade geröstet wird. Dann könnte der Toaster die weiteren Prädiktionen aus dem, zu der Charakteristik des Brotes, passenden Netz beziehen.

Das Verfahren, wie die Testdaten für einen Brottyp ideal aufgenommen werden, benötigt eine überdachte systematische Vorgehensweise. Zusätzlich soll erkannt werden, ob ein Brot zuvor schon getoastet wurde. Hierzu müssen neue Trainingsdaten mit der entsprechenden Klassifizierung aufgenommen werden. Weitere extreme Brotzustände wie Tiefgefrorenes, Ausgetrocknet etc. soll ein idealer Toaster ebenfalls abbilden können, hierzu werden ebenfalls weitere Messdaten benötigt.

Das Aufnehmen der Messdaten und die Klassifizierung sollte möglichst automatisiert ohne menschlichen Unsicherheitsfaktor erfolgen können, beispielsweise durch das Aufnehmen von Bilddaten der Toastoberfläche vor, nach und während dem Röstvorgang.

Die Erarbeitung der relevanten Messfällen, die für die ideale Abbildung von Brot -Toaster Kombination Kennlinien zwingend benötigt werden, könnte die notwendige Anzahl an Messungen reduzieren.

Das Messsystem bietet noch Entwicklungspotential an Hardware- und Softwareseite. Es können weitere Sensormodule entstehen und verschiedene Ausbaustufen des Mainboards. Gepaart mit einer flexiblen Software die einen beliebigen Messaufbau ohne Programmieraufwand konfigurieren lässt, könnte als Prototypisierungs-Plattform von Machine Learning Anwendungen dienen.

7 Literatur- und Quellenverzeichnis

- [1]P. Werbos, "Backpropagation through time: what it does and how to do it", *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550-1560, 1990.
- [2]O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015.
- [3]"2019 recent trends in GPU price per FLOPS", *AI Impacts*, 2020. [Online]. Available: <https://aiimpacts.org/2019-recent-trends-in-gpu-price-per-flops/>. [Accessed: 24- Jul- 2020]
- [4]"NVIDIA cuDNN", *NVIDIA Developer*, 2020. [Online]. Available: <https://developer.nvidia.com/cudnn>. [Accessed: 24- Jul- 2020]
- [5]J. Loderbauer, *Das Bäckerbuch*. Hamburg: Büchner, Handwerk und Technik, 2006.
- [6]"Maillard-Reaktion", *Chemie.de*, 2020. [Online]. Available: <https://www.chemie.de/lexikon/Maillard-Reaktion.html>. [Accessed: 24- Jul- 2020]
- [7]E. Artman, *Toasters 1909-1960*. Atglen, PA: Schiffer Pub., 1996.
- [8]*ESP32 Series Datasheet*. Espressif, 2020 [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Accessed: 25- Jul- 2020]
- [9]Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard and L. Jackel, "Back-propagation Applied to Handwritten Zip Code Recognition", *Neural Computation*, vol. 1, no. 4, pp. 541-551, 1989.
- [10]A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2012.

- [11]C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [12]M. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks", *Computer Vision – ECCV 2014*, pp. 818-833, 2014.
- [13]K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [14]S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size", *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015.
- [15]"sklearn.preprocessing.MinMaxScaler – scikit-learn 0.23.1 documentation", *Scikit-learn.org*, 2020. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>. [Accessed: 25- Jul- 2020]
- [16]X. Glorot, A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks", in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315-323.
- [17]Z. Zhang, Z. Yang, Y. Sun, Y. Wu and Y. Xing, "Lenet-5 Convolution Neural Network with Mish Activation Function and Fixed Memory Step Gradient Descent Method", *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing*, 2019.
- [18]W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115-133, 1943.
- [19]D. Kingma and J. Ba, "Adam; A Method for Stochastic Optimization", *International Conference on Learning Representation*, vol. 14126980, 2015.
- [20]I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. MIT Press, 2016.

- [21]T. Mitchell, *Machine learning*. New York: McGraw Hill, 2017.
- [22]M. Zeiler, "ADADELTA: An adaptive learning rate method.", *CoRR*, vol. 12125701, 2012.
- [23]T. Dozat, *Incorporating Nesterov Momentum into Adam*. ICLR Workshop, 2016 [Online]. Available: <https://openreview.net/pdf?id=OM0jvwB8jlp57ZJjtNEZ>. [Accessed: 25- Jul- 2020]
- [24]J. Duchi, E. Hazan and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", *Journal of Machine Learning Research*, pp. 2121-2159, 2011.
- [25]"sklearn.model_selection.KFold — scikit-learn 0.23.1 documentation", *Scikit-learn.org*, 2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html. [Accessed: 25- Jul- 2020]
- [26]G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303-314, 1989.
- [27]A. Karpathy, "CS231n Convolutional Neural Networks for Visual Recognition", *Cs231n.github.io*, 2020. [Online]. Available: <https://cs231n.github.io/>. [Accessed: 25- Jul- 2020]
- [28]V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning", vol. 160307285, 2016.
- [29]M. Singh, V. Pondenkandath, B. Zhou, P. Lukowicz and M. Liwicki, "Transforming sensor data to the image domain for deep learning — An application to footstep detection", *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017.
- [30]J. Schmidhuber, "Deep Learning", *Scholarpedia*, vol. 10, no. 11, p. 32832, 2015.
- [31]G. Huang, Z. Liu, L. Van Der Maaten and K. Weinberger, "Densely Connected Convolutional Networks", *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [32]S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning", *CoRR*, vol. 181112808, 2018 [Online]. Available: <http://arxiv.org/abs/1811.12808>. [Accessed: 25- Jul- 2020]

- [33]N. Srivastava, G. Hinton, A. Krizhevsky and I. Sutskever, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.
- [34]"{{metadataController.pageTitle}}", *Subscription.packtpub.com*, 2020. [Online]. Available: https://subscription.packtpub.com/book/big_data_and_business_intelligenDidier. [Accessed: 25- Jul- 2020]
- [35]D. Henrion and J. Lasserre, "GloptiPoly", *ACM Transactions on Mathematical Software (TOMS)*, vol. 29, no. 2, pp. 165-194, 2003.
- [36]J. Weng, N. Ahuja and T. Huang, "Learning recognition and segmentation of 3-D objects from 2-D images", 1993 (4th) *International Conference on Computer Vision*.
- [37]J. Schmidhuber, "Who Invented Backpropagation?", *People.idsia.ch*, 2014. [Online]. Available: <http://people.idsia.ch/~juergen/who-invented-backpropagation.html>. [Accessed: 25- Jul- 2020]
- [38]K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", *CoRR*, vol. 150201852, 2015.
- [39]S. Hirano, J. Brubaker, D. Patterson and G. Hayes, "Detecting cooking state with gas sensors during dry cooking", *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13*, 2013.
- [40]Cs.cornell.edu, 2013. [Online]. Available: https://www.cs.cornell.edu/courses/cs1114/2013sp/sections/S06_convolution.pdf. [Accessed: 25- Jul- 2020]
- [41]C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning", *CoRR*, vol. 181103378, 2018.
- [42]H. Sultan, N. Salem and W. Al-Atabany, "Multi-Classification of Brain Tumor Images Using Deep Neural Network", *IEEE Access*, vol. 7, pp. 69215-69225, 2019.

- [43]T. Deng, S. Wang and M. Zhu, "Dual-wavelength optical sensor for measuring the surface area concentration and the volume concentration of aerosols", *Sensors and Actuators B: Chemical*, vol. 236, pp. 334-342, 2016.
- [44]S. Yan, T. Deng, W. Xu and S. Wang, *Selecting an Optimal Set of Scattering Angles and Wavelengths for Practical Photoelectric Smoke Detector*. 2017 [Online]. Available: <http://nts.uni-duisburg-essen.de/aube/aube17/paper/Session%2002/S2P5%20Yan%20et%20al.pdf>. [Accessed: 25- Jul- 2020]
- [45]*Relative Humidity Sensor Technical Data*, 7th ed. Humirel, 2002 [Online]. Available: <https://www.parallax.com/sites/default/files/downloads/27920-Humidity-Sensor-Datasheet.pdf>. [Accessed: 25- Jul- 2020]
- [46]C. de Godzinsky, "Mains Zero-Crossing Detector | DEXTREL", *DEXTREL*, 2020. [Online]. Available: <https://dextrel.net/design-ideas-2/mains-zero-crossing-detector.html>. [Accessed: 25- Jul- 2020]
- [47]H. Schröcke and K. Weiner, *Mineralogie*. .
- [48]D. Cole, E. Browning and F. Schroeder, *Encyclopedia of modern everyday inventions*. .
- [49]K. Gupta and N. Gupta, *Advanced electrical and electronics materials*. .
- [50]T. Hegborn, *Integrating Electrical Heating Elements in Product Design*, 1st ed. CRC Press, 1997.
- [51]M. Newborough, W. Batty and S. Probert, "Design improvements for the ubiquitous electric toaster", *Applied Energy*, vol. 27, no. 1, pp. 1-52, 1987.
- [52]M. Newborough and S. Probert, "Enhancing the heat-transfer performances of conventional open-topped closed-sided toasters", *Applied Energy*, vol. 29, no. 1, pp. 37-55, 1988.
- [53]"Maillard-Reaktion", *Chemie.de*, 2020. [Online]. Available: <https://www.chemie.de/lexikon/Maillard-Reaktion.html>. [Accessed: 26- Jul- 2020]

- [54]"Acrylamide", *Food.gov.uk*, 2018. [Online]. Available: <https://www.food.gov.uk/print/pdf/node/281>. [Accessed: 26- Jul- 2020]
- [55]Y. Xu, B. Cui, R. Ran, Y. Liu, H. Chen, G. Kai and J. Shi, "Risk assessment, formation, and mitigation of dietary acrylamide: Current status and future prospects", *Food and Chemical Toxicology*, vol. 69, pp. 1-12, 2014.
- [56]J. Veerbeek, "www.toastermuseum.com - TOASTER COLLECTION VEERBECK - Companies A-Z", *Toastermuseum.com*, 2020. [Online]. Available: http://www.toaster-museum.com/scripts/toastercollection/col_companies01.html. [Accessed: 26- Jul- 2020]
- [57]D. Aachen, "Rowenta, flatbed toaster (grill) "E 5107""", *Designundklassiker.de*, 2020. [Online]. Available: <https://www.designundklassiker.de/technics/toasters/Rowenta-flatbed-toaster-grill-E-5107::8128.html?language=en>. [Accessed: 26- Jul- 2020]
- [58]Z. Cheng, H. Sun, M. Takeuchi and J. Katto, "Deep Convolutional AutoEncoder-based Lossy Image Compression", *2018 Picture Coding Symposium (PCS)*, 2018.
- [59]I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Brngio, "Generative Adversarial Networks", *ArXiv*, vol. 14062661, 2014.
- [60]"1.11. Ensemble methods — scikit-learn 0.23.1 documentation", *Scikit-learn.org*, 2020. [Online]. Available: <https://scikit-learn.org/stable/modules/ensemble.html#random-forest-feature-importance>. [Accessed: 26- Jul- 2020]
- [61]G. Louppe, L. Wehenkel, A. Suter and P. Geurts, "Understanding variable importances in forests of randomized trees", *Advances in Neural Information Processing Systems*, vol. 26, 2013.
- [62]J. Hu, L. Shen and G. Sun, "Squeeze-and-Excitation Networks", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.
- [63]C. Shorten and T. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning", *Journal of Big Data*, vol. 6, no. 1, 2019.

[64]P. Horowitz and W. Hill, *The art of electronics*, 3rd ed. Cambridge University, 2016, p. Lecture 8.

[65]T. Instruments, *TLC555 LinCMOS Timer*. Texas Instruments, 2020 [Online]. Available: https://www.ti.com/lit/ds/sym-link/tlc555.pdf?ts=1595366750588&ref_url=https%25253A%25252F%25252Fwww.google.com%25252F. [Accessed: 26- Jul- 2020]

[66]"B+B Thermo-Technik | Feuchtesensor kapazitiv KFS330-MIN | Messtechnik online", <https://shop.bb-sensors.com/>, 2020. [Online]. Available: <https://shop.bb-sensors.com/Feuchtemesstechnik/Feuchtesensoren/Feuchtesensor-kapazitiv-KFS330-MIN.html>. [Accessed: 26-Jul- 2020]

[67]*Kapazitiver Feuchtsensor KFS140-D*. BB Sensors, 2016 [Online]. Available: https://shop.bb-sensors.com/out/media/Datenblatt_Kapazitiver_Feuchtsensor_KFS140-D.pdf. [Accessed: 26-Jul- 2020]

[68]*PCA9615 2-channel multipoint Fast-mode Plus differential I2C-bus buffer with hot-swap logic*. NXP, 2016 [Online]. Available: <https://www.nxp.com/docs/en/data-sheet/PCA9615.pdf>. [Accessed: 26- Jul- 2020]

[69]*TCS3472 Color Light-to-Digital Converter with IR Filter*. AMS, 2020 [Online]. Available: https://ams.com/documents/20143/36005/TCS3472_DS000390_3-00.pdf/6fe47e15-e32f-7fa7-03cb-22935da44b26. [Accessed: 26- Jul- 2020]

[70]"Bashaw Fire Hall Museum - Town of Bashaw", *Town of Bashaw*, 2020. [Online]. Available: <http://townofbashaw.com/community/arts-culture/bashaw-fire-hall-museum/>. [Accessed: 26-Jul- 2020]

[71]"tf.keras.callbacks.EarlyStopping | TensorFlow Core v2.2.0", *TensorFlow*, 2020. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping. [Accessed: 26- Jul- 2020]

[72]H. Isleroglu and S. Beyhan, "Prediction of baking quality using machine learning based intelligent models", *Heat and Mass Transfer*, vol. 56, no. 7, pp. 2045-2055, 2020.

[73]L. Breiman, *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

8 Abkürzungsverzeichnis

EFSA - European Food Safety Authority

SPI Serial Peripheral Interface

WLAN Wireless Local Area Network

LDO Low-dropout regulator

POSIX Portable Operating System Interface

SOC System on a chip

ASIC Application-specific integrated circuit

FSM Finite State Machine

IOT Internet of Things

UART Universal Asynchronous Receiver-Transmitter

DMA Direct Memory Access

ZCD Zero Crossing Detector

CSV Comma Separated Values

MSE Mean Squared Error

SGD Stochastic Gradient Descent

GFLOPS (Giga) Floating Point Operations (per) Second

CUDNN CUDA Deep Neural Network

FDM Fused Deposition Modeling

DLP Digital Light Processing

ISM Industrial, Scientific and Medical

CNN Convolutional Neural Network

NN (Artificial) Neural Network

PAN Personal Area Network

SSR Solid State Relay

9 Abbildungsverzeichnis

Abbildung 1-1: General Electric D-12 - Percher und Elekthermax VP-3 – Pincher [56] ...	2
Abbildung 1-2: Copeman Turnover [US1108552A].....	3
Abbildung 1-3: Heatmaster No.692 Dropdown [56] und Flatbed [57].....	3
Abbildung 1-4: Universal E 947 [56] Turner	4
Abbildung 1-5: Universal E 9422 - Slider[56], Landers, Frary & Clark, circa 1930s. – Tipper[70].....	4
Abbildung 1-6: Toast-O-Lator Model G - Drive Through, Strites Pop Up [56]	5
Abbildung 1-7: Sunbeam Radiant Toaster [US2459170A].....	6
Abbildung 2-1: Umfeld dieser Arbeit, angelehnt an [20]	11
Abbildung 2-2: Perceptron dessen Ausgang in eine Aktivierungsfunktion geführt wird	12
Abbildung 2-3: Separierbarkeit [34].....	13
Abbildung 2-4: Multi Layer Perceptron	14
Abbildung 2-5: Six Hump Camel Back Function [35], SGD Iterative Darstellung [20]	15
Abbildung 2-6: Verfolgung der Gradienten	15
Abbildung 2-7: Single Perceptron Netzwerk – Backpropagation	17
Abbildung 2-8: Cross Validation mit K-Fold.....	19
Abbildung 2-9: Convolution Visualisierung.....	20
Abbildung 2-10: Convolution 1D Array.....	21
Abbildung 2-11: Convolution – Stride und Dilation.....	21
Abbildung 2-12: Rectified Linear Unit – ReLU [42].....	22

Abbildung 2-13: Max Pooling	23
Abbildung 2-14: Links – Standard Neural Netzwerk, Rechts – Dropout Neural Netzwerk [33].....	24
Abbildung 2-15: Iterationen - Epochs.....	24
Abbildung 2-16: Confusion Matrix	25
Abbildung 3-1: Mainboard	28
Abbildung 3-2: Übersicht aus dem Blickwinkel des Mikrocontrollers	29
Abbildung 3-3: Spannungsversorgung Mainboard.....	29
Abbildung 3-4: Spannungsversorgung - Planes	30
Abbildung 3-5: Programmierschnittstelle.....	30
Abbildung 3-6: I2CD Port Schematik.....	31
Abbildung 3-7: I2CD Signal Selektion.....	32
Abbildung 3-8: GPIO und PWM Expansion	32
Abbildung 3-9: Halbbrücke Schematik	33
Abbildung 3-10: Ausgang mit N Kanal MOSFET	33
Abbildung 3-11: Halbbrücke Schematik	33
Abbildung 3-12: Analog Schnittstelle für drei Leiter PT1000 Messung	34
Abbildung 3-13: Versorgungsplatine Platzierung der Bauteile, Anschlüsse	35
Abbildung 3-14: Versorgungsplatine Schematik.....	36
Abbildung 3-15: Kühlungslösung Leistungselektronik	36
Abbildung 3-16: ADC Modul	37
Abbildung 3-17: RJ45 T Splitter.....	37

Abbildung 3-18: Wägemesszellen Modul	38
Abbildung 3-19: Gas Sensor Blockschaltbild	39
Abbildung 3-20: Rohrsensor Modul	39
Abbildung 3-21: Rohrsensor PCB.....	40
Abbildung 3-22: ADC Modul	41
Abbildung 3-23: Schematik Kapazität – Frequenz Konverter	42
Abbildung 3-24: Schematik Licht – Frequenz Konverter	42
Abbildung 3-25: IR Temperatur Sensor.....	43
Abbildung 3-26: Farbsensor Modul.....	44
Abbildung 3-27: TCS34725 Blockschaltbild [69].....	44
Abbildung 3-28: LED Modul	45
Abbildung 3-29: Schematik LED Modul	45
Abbildung 3-30: Encoder Modul	46
Abbildung 3-31: Encoder Schematik.....	46
Abbildung 4-1: Dualt 2+1 Slice Toaster	48
Abbildung 4-2: Explosionszeichnung des Prototyps	49
Abbildung 4-3: Pressglimmerplatte mit Löchern für optischen Sensoren	50
Abbildung 4-4: Obere Ansicht – fertiger Prototyp.....	51
Abbildung 4-5: Seitenansicht - Waage	52
Abbildung 4-6: Platzierung der Temperatur Sensoren	53
Abbildung 4-7: Platzierung der Sensoren in der Lüfterstrecke	54
Abbildung 4-8: Seitenansicht des fertigen Prototyps	55

Abbildung 4-9: Grundplatte des fertigen Prototyps	56
Abbildung 4-10: Erster Prototyp des Kurvenlüftergehäuses und mehrere Versuche von Lüfterrädern	56
Abbildung 4-11: Anordnung der Komponenten des Rauchsensors	57
Abbildung 4-12: Messablauf Rauchsensor	57
Abbildung 4-13: Fertiger Prototyp	58
Abbildung 4-14: Aufteilung der Aufgaben zwischen den Kernen	59
Abbildung 4-15: Vereinfachter Ablauf DAQ	60
Abbildung 4-16: Toaster GUI zeigt den Verbindungsstatus und den Button um den Toast abzusenden.....	63
Abbildung 4-17: Ansteuerung Heizer und Einschalttaste für den Download Mode	63
Abbildung 4-18: Hauptbildschirm des Toasters	64
Abbildung 4-19: Anzeige der Sensordaten	65
Abbildung 4-20: Ablauf Diagramm – Heater Control.....	65
Abbildung 4-21: Temperatur Regler Auswahl Menü.....	66
Abbildung 4-22: Toaster Tool Programm Funktionalität.....	66
Abbildung 5-1: Bewertung.....	68
Abbildung 5-2: Messsystem wird auf Anfangstemperatur gekühlt.....	71
Abbildung 5-3: Messung Nr. 49.....	72
Abbildung 5-4: Verfahren zur Generierung von Messdaten.	73
Abbildung 5-5: Sensordaten- Darstellung in der Bild Domain	74
Abbildung 5-6: Optimiertes Verfahren zur Generierung von Messdaten.....	75
Abbildung 5-7: Modelle für binäre Klassifikation	78

Abbildung 5-8: Confusion Matrix – Binäre Klassifikation	79
Abbildung 5-9: Modelle für Klassifikation nach den drei Klassen	81
Abbildung 5-10: Trainingsverlauf Modell F.....	83
Abbildung 5-11: Modell F Confusion Matrix.....	83
Abbildung 5-12: Model F – Visualisierung Layer Outputs	84
Abbildung 5-13: Random Forests Modell F16, eines der Entscheidungsbäume.....	87
Abbildung 5-14: Confusion Matrix der Modelle GB (I.), RF32(II.) und RF16 (III.)	87
Abbildung 5-15: Bilverarbeitungsschritte der Toastbilder.....	88
Abbildung 5-16: Zuordnung der Binary Blobs zu Messungen und Brot Seiten	89
Abbildung 5-17: Nachbearbeitung einzelner Toastbilder um zwei Datensätze mit schwarzem und weißem Hintergrund zu erhalten.....	89
Abbildung 5-18: CNN – Toast Decoder	90
Abbildung 5-19: Vorhersagen des Toastdecoders in Vergleich zu tatsächlichen Brotzustände	91
Abbildung 5-20: Live Predictor	92
Abbildung 6-1: Messsystem.....	93
Abbildung 6-2: Aufgestellte Modelle für Prädiktionen basierend auf Sensordaten eines Toasters.....	94
Abbildung 6-3: Bosch, Sage/Breville, Prototyp mit LivePredictor auf Dualit Basis	95
Abbildung 6-4: Test der Wiederholungsgenauigkeit.....	95
Abbildung 0-1: Explosionszeichnung - Front Links.....	XXVIII
Abbildung 0-2: Finaler Prototyp.....	XXIX
Abbildung 0-3: Finaler Prototyp – Mainboard Verkabelt.....	XXIX

Abbildung 0-4: Finaler Prototyp, Montage Encoder und Display Modul.....	XXIX
Abbildung 0-5: IR Thermometer Modul.....	XXX
Abbildung 0-6: Ein früher Prototyp mit DC Motor	XXXI
Abbildung 0-7: Ein früher Prototyp mit DC Motor und Silikon Schläuche mit Lüfterstrecken Segmente	XXXII
Abbildung 0-8: Wägemesszelle im Eingebauten Zustand.....	XXXIII
Abbildung 0-9: Inbetriebnahme	XXXIV
Abbildung 0-10: Module des Messsystems	XXXV
Abbildung 0-11: Farbsensor Modul.....	XXXV
Abbildung 0-12: Prototypen Lüfterstrecke	XXXVI
Abbildung 0-1: Modell GB	XLI
Abbildung 0-2: Model D.....	XLII
Abbildung 0-3: Model B.....	XLIII
Abbildung 0-4: Model C.....	XLIV
Abbildung 0-1: Vorhersagen - Toast Decoder Upsampling Modell.....	XLV

10 Tabellenverzeichnis

Tabelle 2-1: NAND Wahrheitstabelle.....	13
Tabelle 2: HT Feuchtigkeitssensoren des Messsystems [66][67]	41
Tabelle 3: Übersicht Sensor Kanäle des Messsystems.....	58
Tabelle 4: Metadaten Einer Messung Teil 1	69
Tabelle 5: Metadaten Einer Messung Teil 2	69
Tabelle 6: Rohdaten einer Messung Teil 1	70
Tabelle 7: Rohdaten einer Messung Teil 2	70
Tabelle 8: Rohdaten einer Messung Teil 3	70
Tabelle 9: Binäre Klassifikation Auswahl aus den getesteten Modellen	74
Tabelle 10: Binäre Klassifikation Auswahl aus den getesteten Modellen	75
Tabelle 11: Binäre Klassifikation Auswahl aus den getesteten Modellen	76
Tabelle 12: Binäre Klassifikation Auswahl aus den getesteten Modellen	78
Tabelle 13: Ergebnisse Model A.....	79
Tabelle 14: Ergebnisse Model AL.....	79
Tabelle 15: Ergebnisse Model C.....	79
Tabelle 16: Ergebnisse Model TB.....	79
Tabelle 17: Multiclass - Auswahl aus den getesteten Modellen.....	80
Tabelle 18: Vergleich der Ergebnisse – Modelle für Klassifikation nach drei Klassen – Random Slicing.....	82
Tabelle 19: Vergleich der Ergebnisse – Modelle für Klassifikation nach drei Klassen – Random Slicing with Padding.....	82

Tabelle 20: Ergebnisse Model F	83
Tabelle 21: Vergleich der Ergebnisse – Relevanz einzelner Sensoren.....	85
Tabelle 22: Random Forests Modelle	87
Tabelle 23: Toastzeiten	95
Tabelle 24: Relevanz Sensoren zweiter Durchlauf.....	XXXVIII
Tabelle 25: RF32 Relevanz Sensoren.....	XXXIX
Tabelle 26: RF16 Relevanz Sensoren.....	XL

Anhang A1

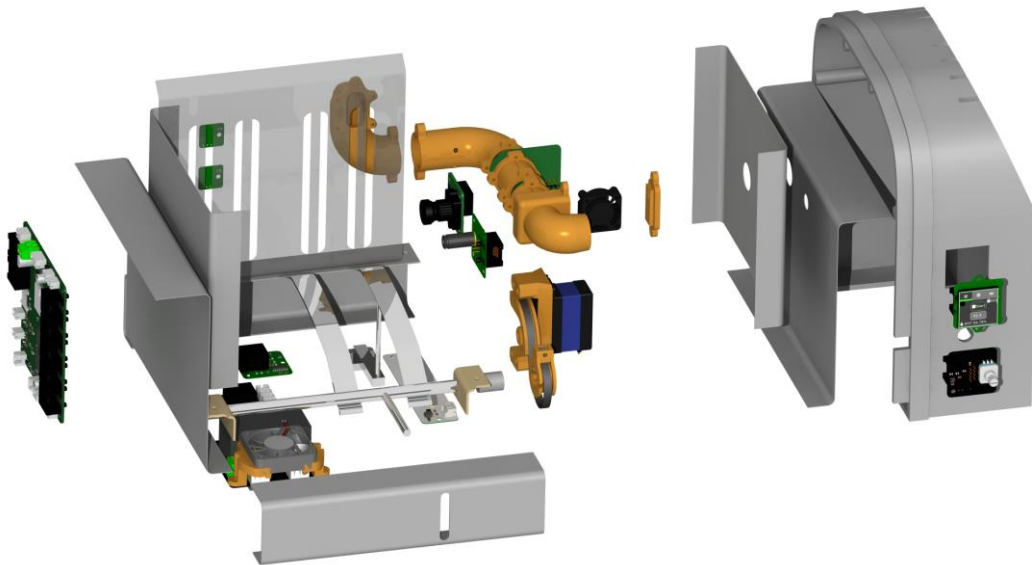


Abbildung 0-1: Explosionszeichnung - Front Links

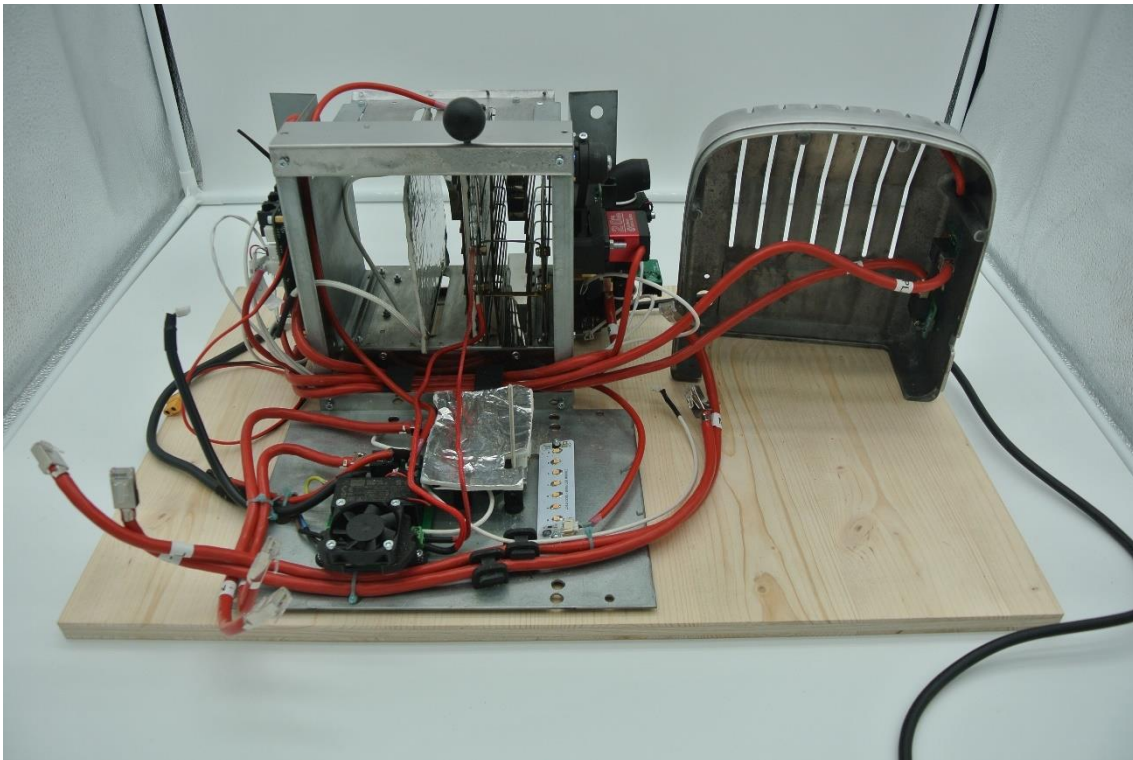


Abbildung 0-2: Finaler Prototyp

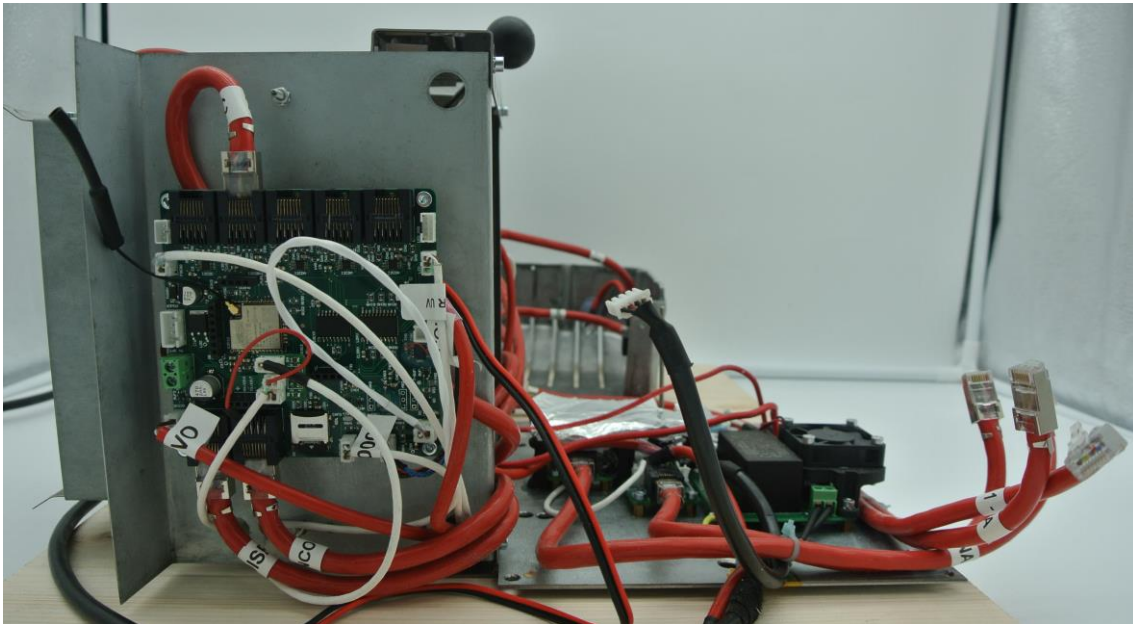


Abbildung 0-3: Finaler Prototyp – Mainboard Verkabelt



Abbildung 0-4: Finaler Prototyp, Montage Encoder und Display Modul

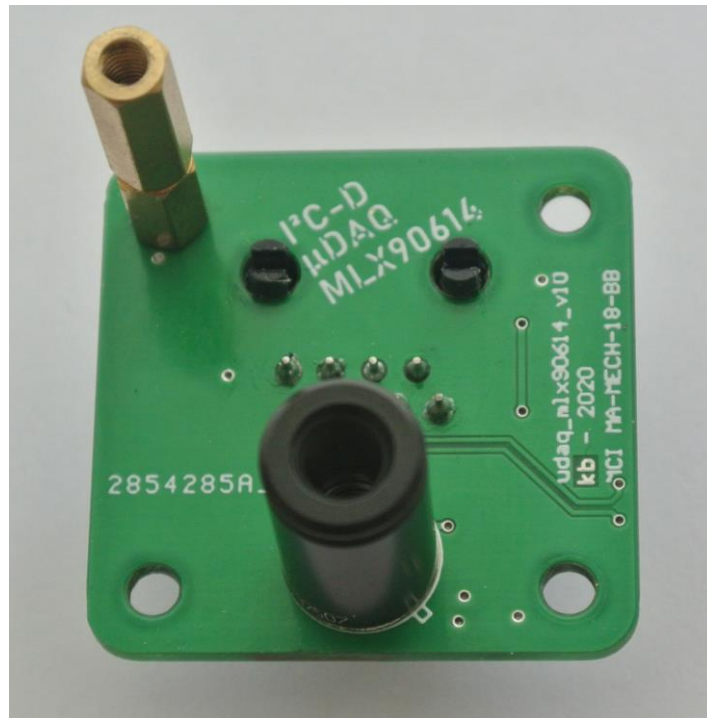


Abbildung 0-5: IR Thermometer Modul

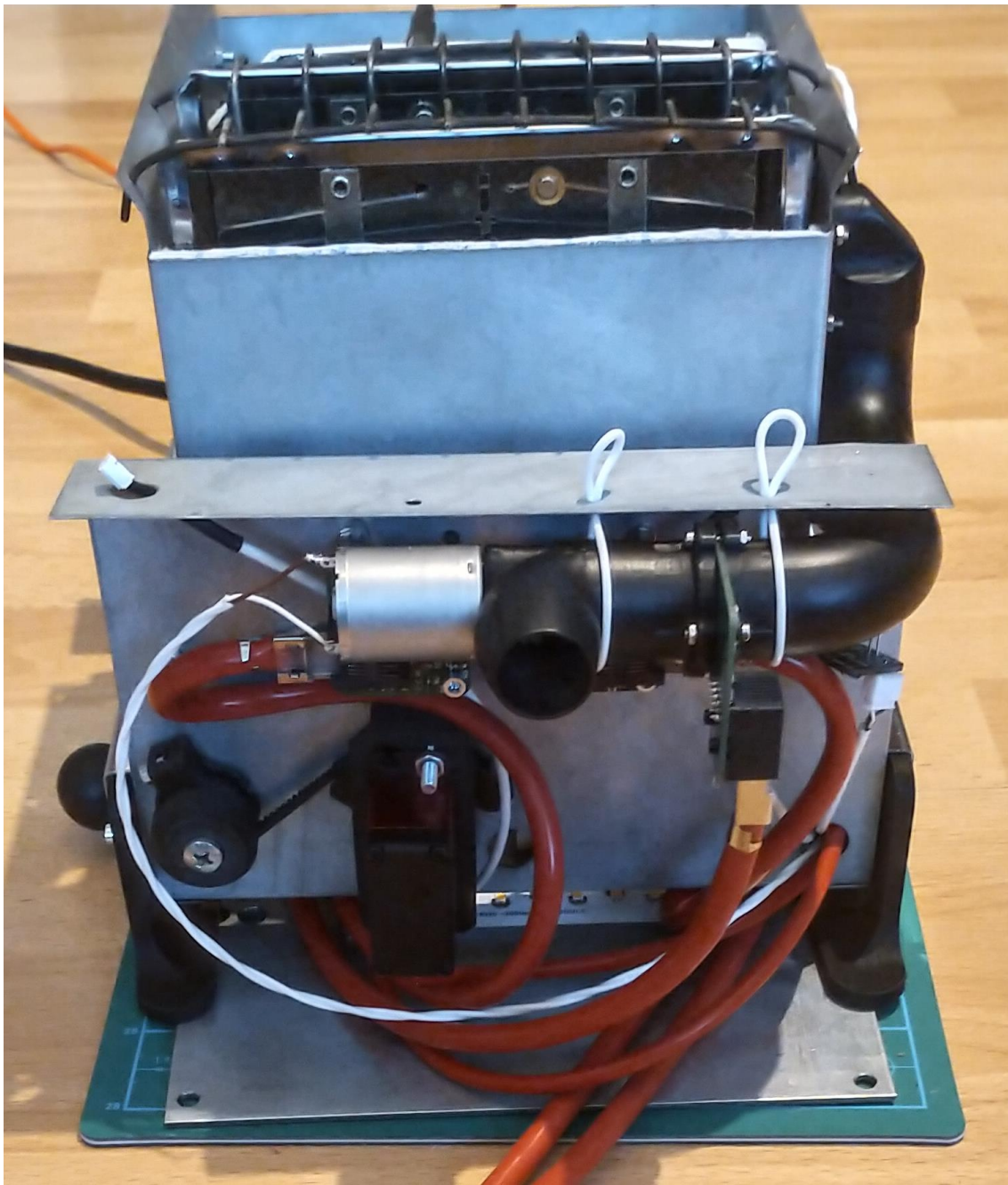


Abbildung 0-6: Ein früher Prototyp mit DC Motor

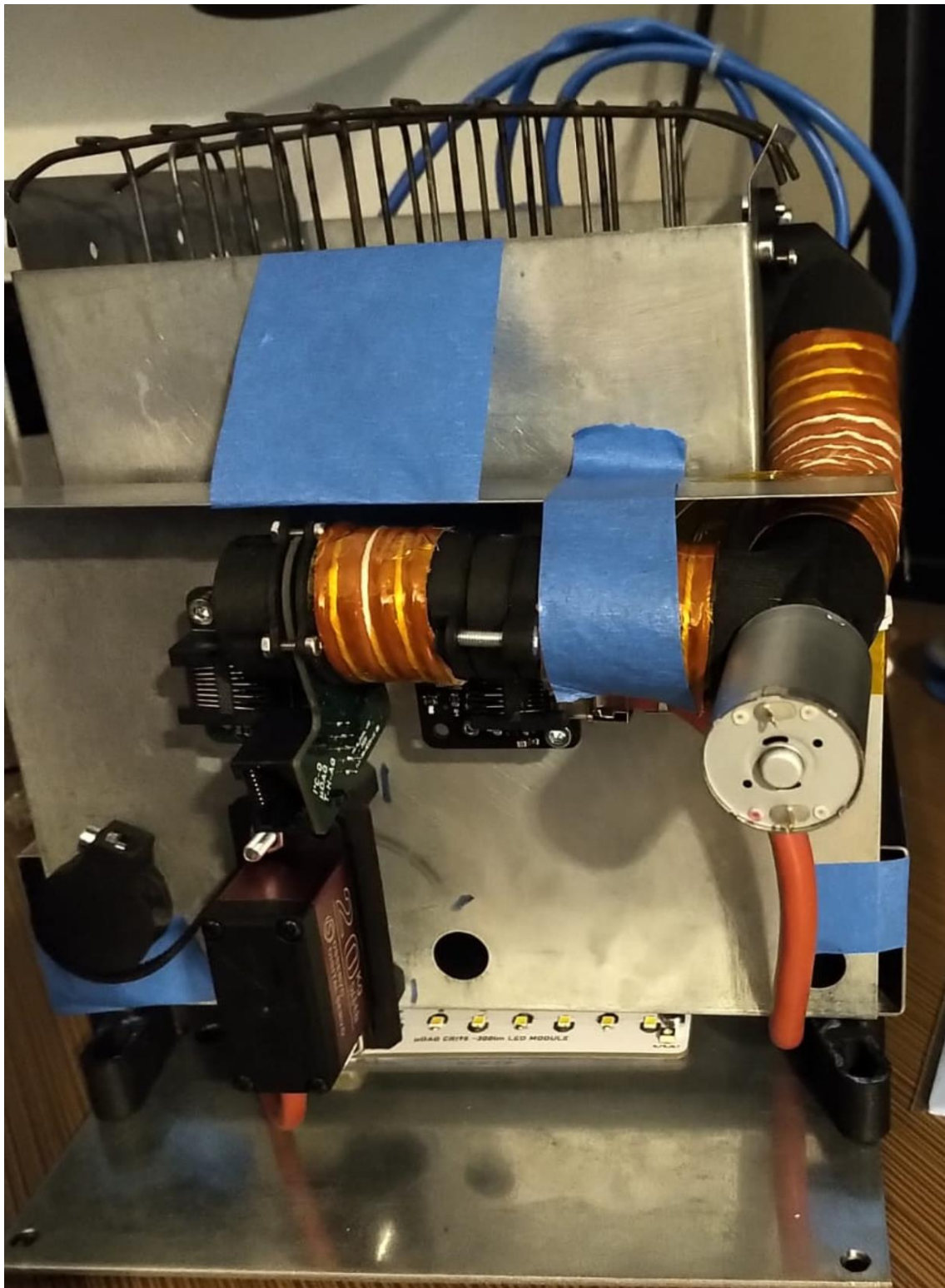


Abbildung 0-7: Ein früher Prototyp mit DC Motor und Silikon Schläuche mit Lüfterstrecken Segmente

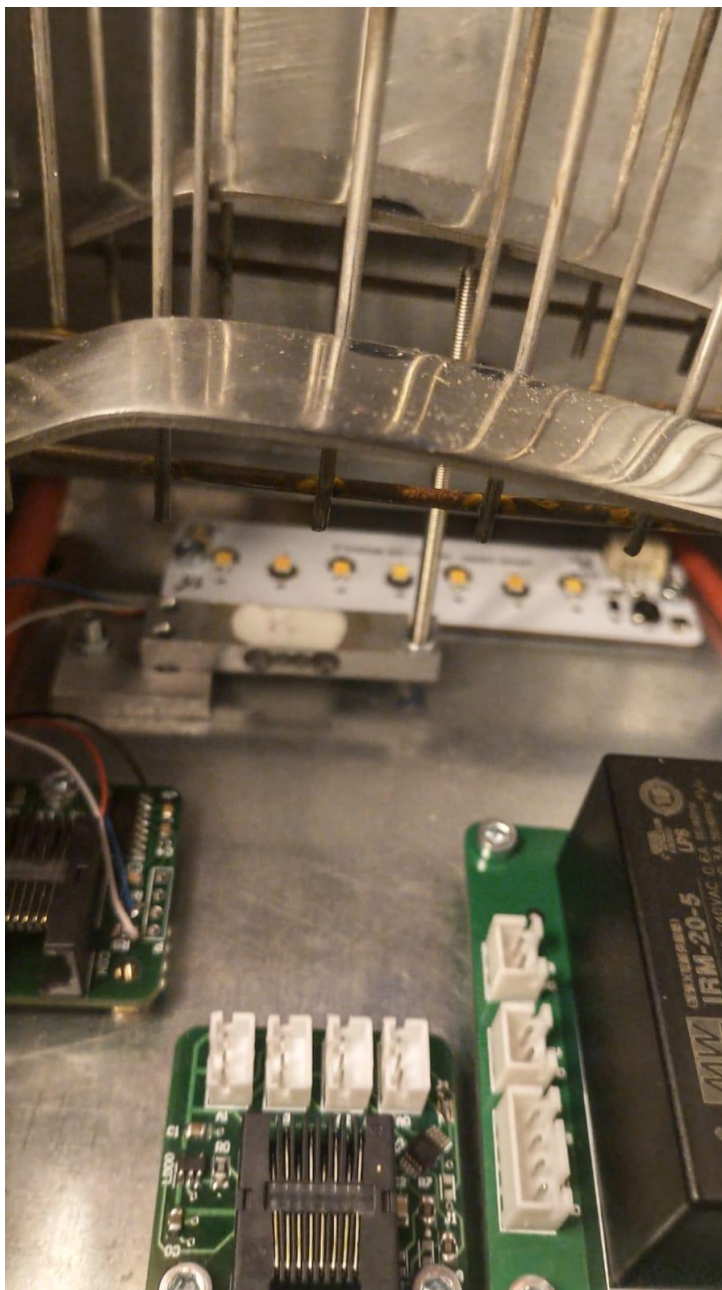


Abbildung 0-8: Wägemesszelle im Eingebauten Zustand

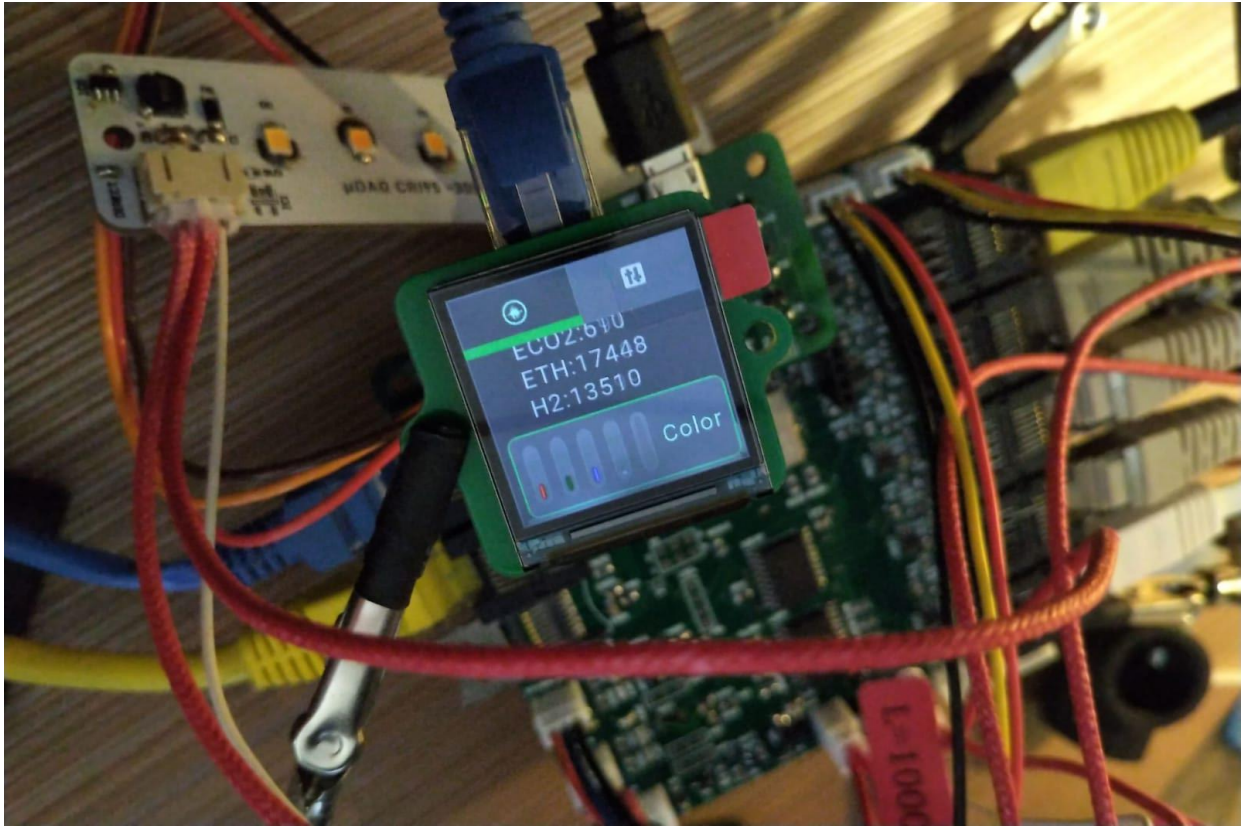


Abbildung 0-9: Inbetriebnahme

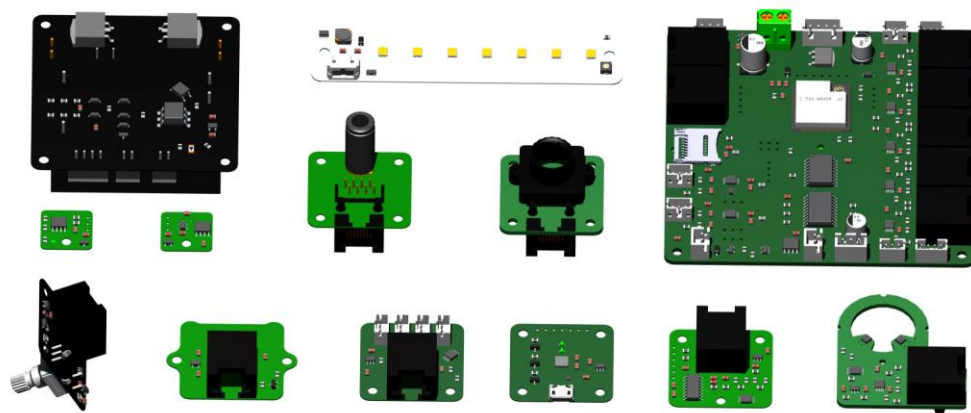


Abbildung 0-10: Module des Messsystems

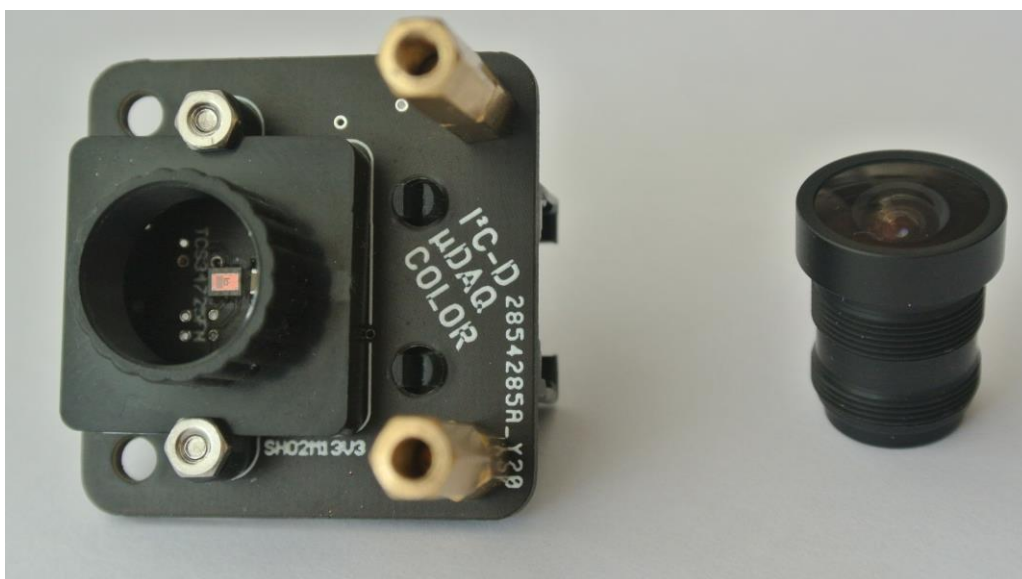


Abbildung 0-11: Farbsensor Modul



Abbildung 0-12: Prototypen Lüfterstrecke

Anhang A2

Weitere mechanische Modifikationen:

- Aufbohren von bestehenden Nieten
- Bearbeitung bestehenden und neuen Blechen, Löcher und Aussparungen für Verschraubungen, Optische Sensoren, Montage.
- Anfertigung einer neuen Aluminium Grundplatte welche genügende Steifigkeit für den Aufbau einer Waage erlaubt, sowie gleichzeitig als Kühl Körper für die Leistungselektronik dient.
- Glimmer Platten mit Bohrungen versehen.
- Heizelemente fixiert befestigen, hierzu Anfertigung von Bronze kantenbleche, Durchbohrung des Heizelement Halterung Käfigs und direkte Vernietung auf das Gehäuse. (Die originale Befestigung lässt viel Bewegung für den Heizelementen Rahmen zu, was dazu führt dass die optische Sensoren den Sichtfeld verlieren)
- Thermische Isolierung der Heizelemente, sowie Grundplatte durch Keramikfasermatten mit Alubeschichtung
- Konstruktion und Prototypisierung für Kabelbefestigungen und Führungen
- Fixierte Positionierung in länglicher Richtung des Blechs der die optische Sensoren halten. Hierzu wird eine Verstrebung durch Gewinde Stange angebracht.

Anhang A3

Tabelle 24: Relevanz Sensoren zweiter Durchlauf

	Genauig- keit AVG.	Genauig- keit MAX.	PREC. AVG.	PREC. MAX.	RECALL. AVG.	RECALL. MAX.	F1 AVG.	F1 MAX.
Alle Sensoren	0.952	0.968	0.942	0.963	0.946	0.963	0.943	0.962
Kein SHT31 (Rohrsensor Temp und Humid.)	0.944	0.959	0.937	0.951	0.934	0.952	0.935	0.951
Kein MLX* (Passive IR und Ambient Temp)	0.947	0.965	0.935	0.958	0.939	0.965	0.937	0.961
Kein PT1000A	0.954	0.963	0.944	0.951	0.948	0.957	0.945	0.953
Kein PT1000B	0.952	0.964	0.941	0.955	0.945	0.959	0.943	0.956
Keine HT feste Luft- feuchtigkeitssensoren	0.950	0.959	0.938	0.954	0.943	0.956	0.940	0.952
Kein Farbsensor	0.953	0.976	0.941	0.968	0.945	0.974	0.942	0.970
Kein SGP30 (Rohrsensor Ethanol und H2)	0.944	0.963	0.934	0.951	0.937	0.958	0.934	0.954
Keine Waage	0.955	0.971	0.947	0.965	0.947	0.965	0.946	0.965
Keine Temp A0	0.952	0.963	0.943	0.960	0.942	0.955	0.942	0.958
Keine Temp A1	0.947	0.963	0.938	0.954	0.938	0.952	0.937	0.952
Keine Temp A2	0.956	0.970	0.947	0.968	0.952	0.966	0.949	0.966
Keine Temp A3	0.946	0.966	0.935	0.959	0.938	0.961	0.936	0.959
SELECTION_A	0.948	0.964	0.935	0.955	0.941	0.961	0.937	0.957
SELECTION_B	0.949	0.960	0.936	0.945	0.940	0.951	0.938	0.948
SELECTION_C	0.950	0.961	0.940	0.955	0.942	0.963	0.940	0.958
SELECTION_D	0.944	0.958	0.933	0.949	0.933	0.953	0.933	0.951
SELECTION_E	0.921	0.932	0.905	0.920	0.904	0.918	0.903	0.916
SELECTION_F	0.945	0.963	0.933	0.956	0.938	0.959	0.935	0.957
SELECTION_G	0.915	0.928	0.901	0.908	0.900	0.915	0.899	0.910
SELECTION_H	0.940	0.958	0.926	0.950	0.931	0.951	0.928	0.951
Alle Sensoren - Zweiter Durchlauf	0.953	0.962	0.943	0.953	0.945	0.955	0.943	0.953
Nur Farbsensor	0.908	0.921	0.889	0.910	0.893	0.912	0.890	0.911
Nur MLX* (Passive IR und Ambient Temp)	0.926	0.932	0.913	0.920	0.912	0.921	0.912	0.919
Nur PT1000A	0.876	0.888	0.856	0.867	0.849	0.873	0.850	0.868
Nur PT1000B	0.913	0.920	0.894	0.905	0.897	0.904	0.894	0.902
SELECTION_S	0.951	0.969	0.938	0.960	0.944	0.963	0.940	0.961

Anhang A4

Tabelle 25: RF32 Relevanz Sensoren

Kanal	t/100ms	Relevanz
MEAS_A_3	675	0.024
PT1000A	673	0.013
MEAS_FREQ.1	683	0.013
MEAS_FREQ.2	698	0.013
MLX_TA	671	0.012
PT1000B	671	0.012
MLX_TA	676	0.012
MEAS_A_3	676	0.012
MEAS_COL_G	677	0.012
MLX_TA	678	0.012
MEAS_rawEthanol	678	0.012
ABS_HUMI	678	0.012
PT1000A	679	0.012
MEAS_A_0	679	0.012
SHT_T	680	0.012
MEAS_rawEthanol	680	0.012
PT1000A	681	0.012
PT1000B	681	0.012
MEAS_COL_R	681	0.012
SHT_T	682	0.012
MEAS_A_2	682	0.012
MEAS_COL_R	684	0.012
MLX_TA	685	0.012
PT1000B	686	0.012
MEAS_A_3	686	0.012
PT1000B	693	0.012
MEAS_rawEthanol	697	0.012
MEAS_A_0	698	0.012
SHT_H	699	0.012
PT1000B	700	0.012
MEAS_A_1	698	0.011
MEAS_A_3	885	0.007
PT1000B	927	0.006
MEAS_A_3	893	0.005
PT1000B	923	0.005
MEAS_A_3	760	0.004
MEAS_A_3	793	0.004
MEAS_A_3	796	0.004
MEAS_A_3	808	0.004
MEAS_A_3	840	0.004
MEAS_A_3	844	0.004
MEAS_A_3	851	0.004
MEAS_FREQ.1	860	0.004
MEAS_A_3	865	0.004
MEAS_A_3	868	0.004
MEAS_A_3	880	0.004
MEAS_A_3	881	0.004
MEAS_A_3	882	0.004
PT1000B	905	0.004
PT1000B	912	0.004
Summe		0.469

Tabelle 26: RF16 Relevanz Sensoren

Kanal	t/100ms	Relevanz
MEAS_COL_R	681	0.035
MEAS_FREQ.2	698	0.034
MLX_TA	671	0.033
MEAS_FREQ.1	683	0.033
MEAS_A_3	676	0.032
MEAS_rawEthanol	680	0.032
PT1000B	671	0.031
MEAS_COL_G	677	0.031
MEAS_A_0	679	0.031
SHT_T	680	0.031
PT1000B	681	0.031
MLX_TA	685	0.031
MEAS_A_0	698	0.031
PT1000B	700	0.031
MEAS_A_3	675	0.03
MEAS_A_2	682	0.03
MEAS_A_3	885	0.019
MEAS_A_3	893	0.013
PT1000B	927	0.011
MEAS_A_3	840	0.01
MEAS_A_3	880	0.01
PT1000B	905	0.01
PT1000B	923	0.01
MEAS_A_3	760	0.009
MEAS_A_3	844	0.009
MEAS_A_1	899	0.009
MEAS_FREQ.1	927	0.009
PT1000B	929	0.009
PT1000B	950	0.009
MEAS_A_1	808	0.008
PT1000B	998	0.008
MEAS_A_0	55	0.006
MEAS_A_0	601	0.006
MEAS_A_3	849	0.006
PT1000B	1208	0.006
MEAS_weight	1213	0.006
MLX_TA	1214	0.006
MEAS_weight	1292	0.006
SHT_T	3	0.005
MEAS_A_0	6	0.005
PT1000B	20	0.005
PT1000B	115	0.005
MEAS_A_0	206	0.005
PT1000A	382	0.005
MLX_TA	402	0.005
MEAS_FREQ.2	523	0.005
MLX_TO	577	0.005
PT1000B	844	0.005
PT1000B	861	0.005
MEAS_FREQ.2	974	0.005
Summe		0.762

Anhang A5

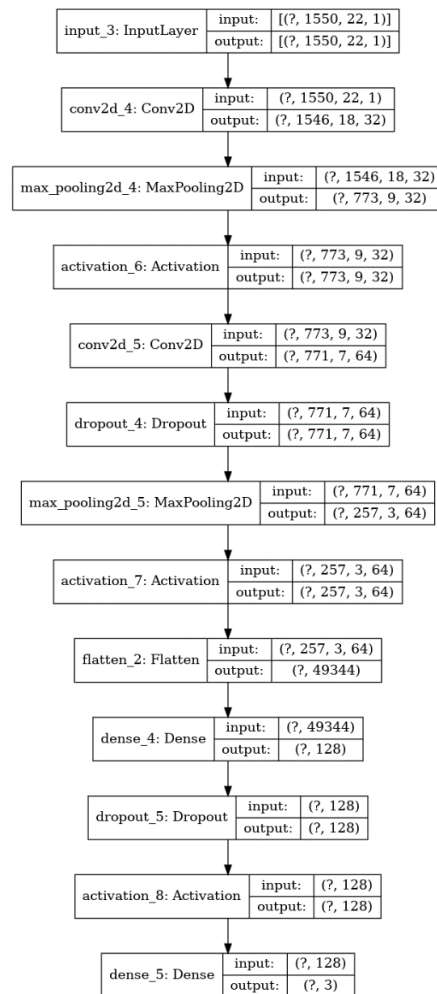


Abbildung 0-1: Modell GB

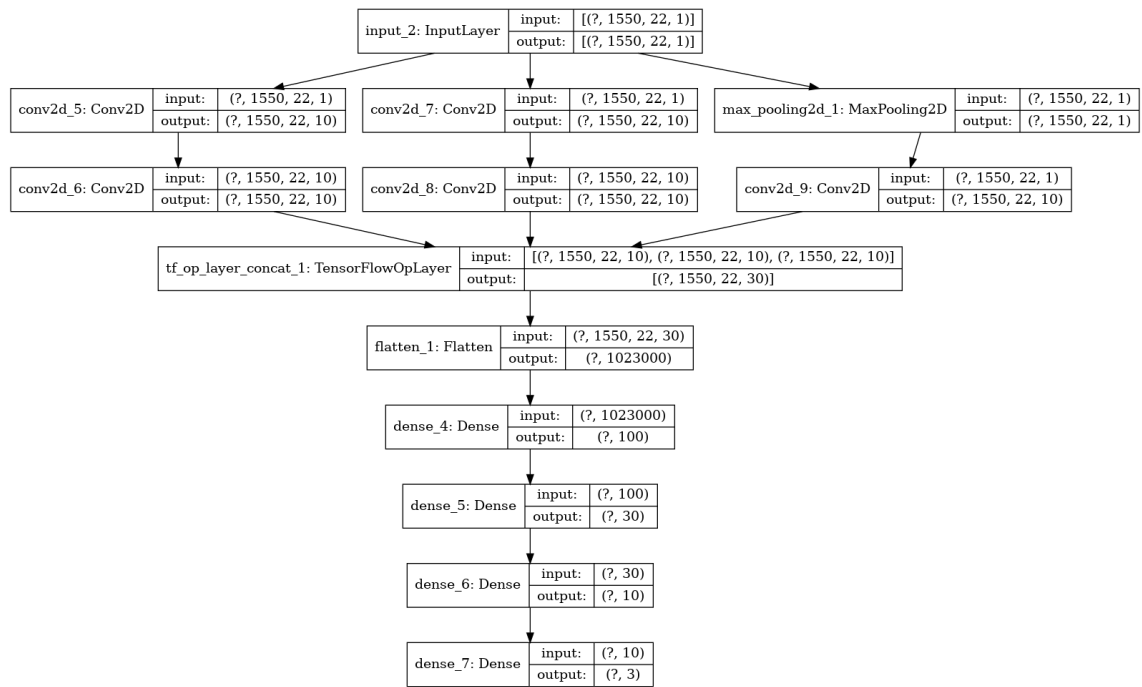


Abbildung 0-2: Model D

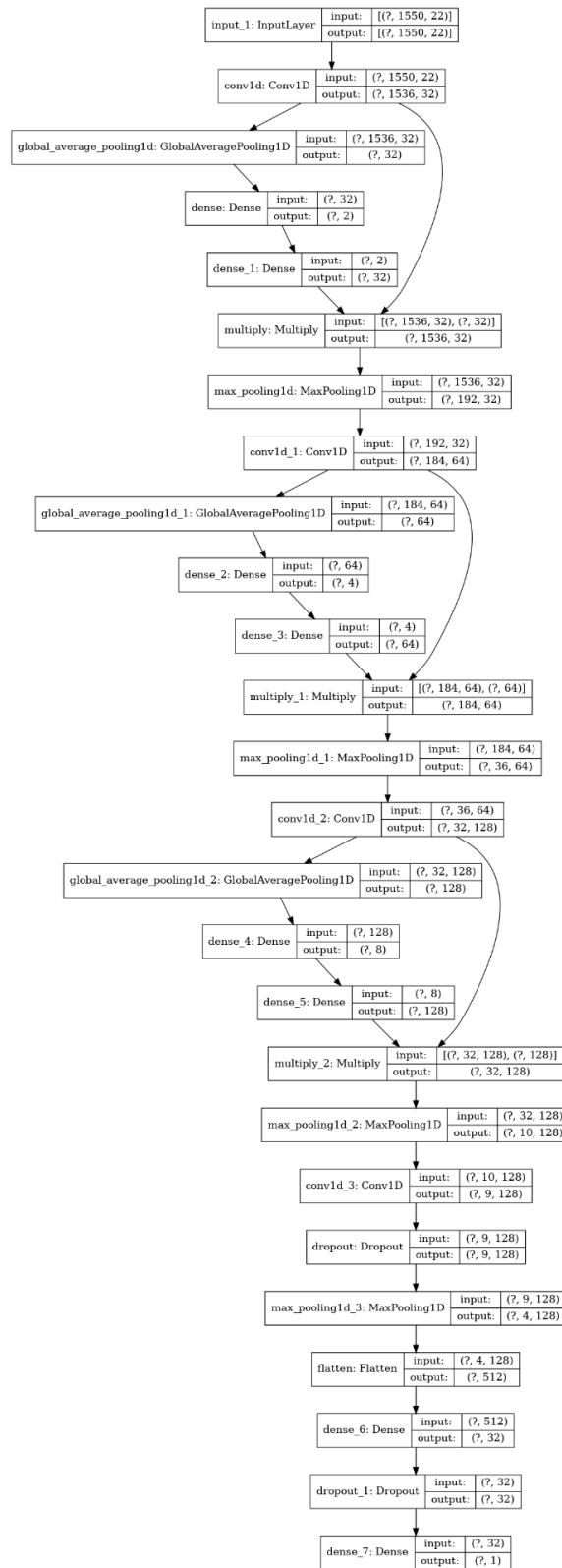


Abbildung 0-3: Model B

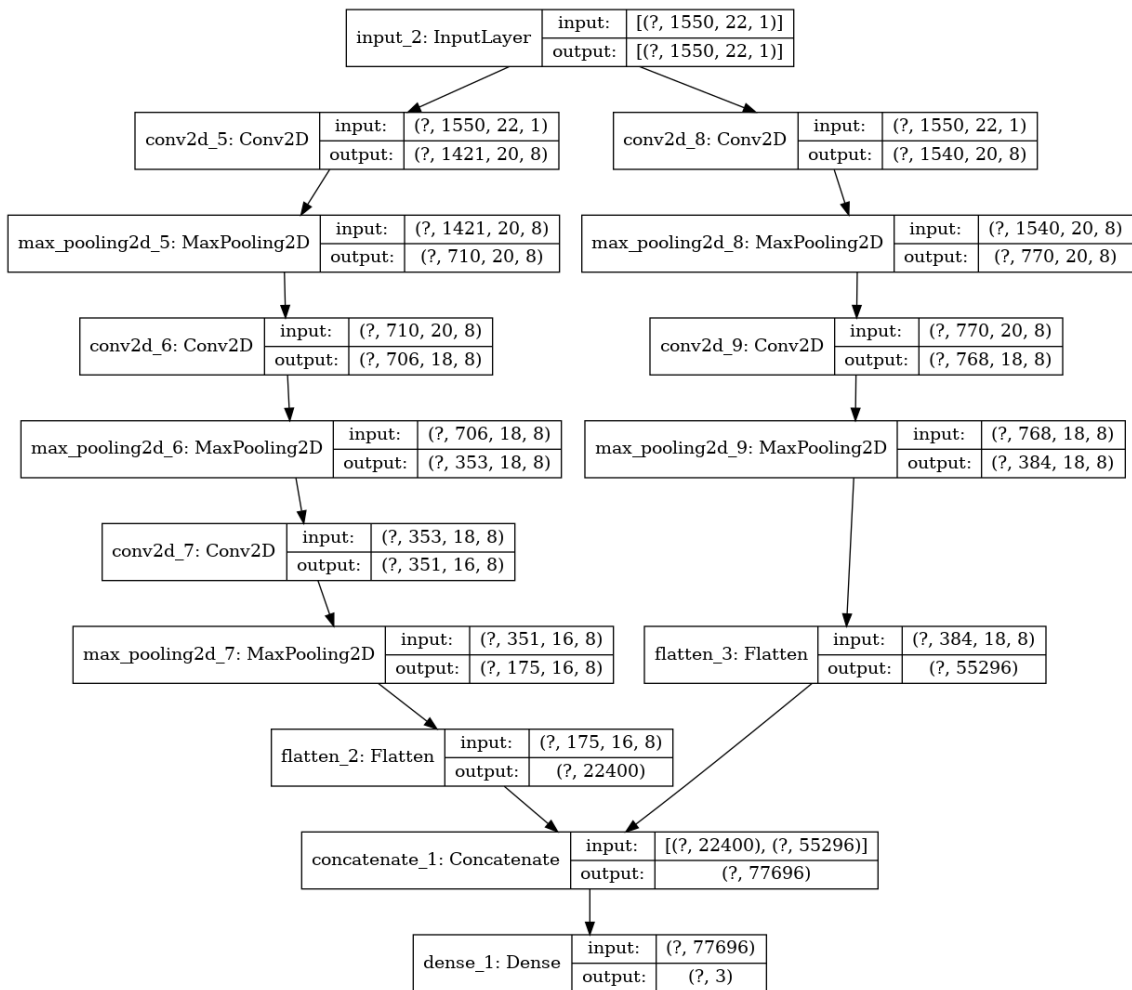


Abbildung 0-4: Model C

Anhang A6

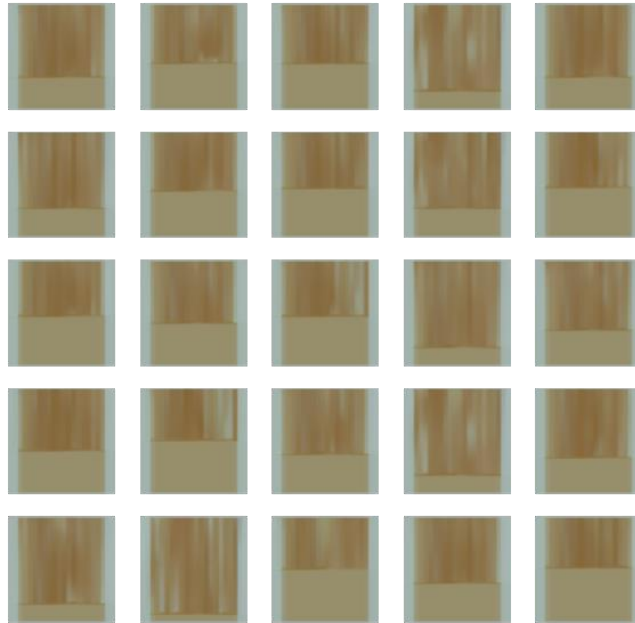


Abbildung 0-1: Vorhersagen - Toast Decoder Upsampling Modell

Anhang A7

Anhang A7-1

```
void updateWeights(float* m, float* b, float* x, float* y, int n, float lr)
{
    float dm = 0, db = 0;
    for (int i=0; i < n; i++)
    {
        dm += -2 * x[i] * (y[i] - (*m * x[i] + *b) );
        db += -2 * ( y[i] - ( *m * x[i] + *b));
    }
    *m -= (dm / n) * lr;
    *b -= (db / n) * lr;
}
```

Anhang A7-2

```
#!/usr/bin/env python
import sys; sys.path.insert(0, "/usr/local/lib/python3.8/dist-packages")
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
import io
import os
```

```

import numpy as np
import math
import tensorflow
import os
import errno
import sys

def writePidFile():
    pid = str(os.getpid())
    f = open('/tmp/livePredictor.pid', 'w')
    f.write(pid)
    f.close()

PERFECT = 0
TOO_EARLY = 1
TOO_LATE = 2

PREDICTIONS = ["PERFECT", "TOO_EARLY", "TOO_LATE"]
zeroPaddedLength = 1550
FIFO_IN = "/tmp/cToPython"
FIFO_OUT = "/tmp/pythonToC"
RESET = "R\n"
LOADED = "L\n"

writePidFile()
model = tensorflow.keras.models.load_model('/home/b/ws_python/livePredictModel_MULTI_SPARSE.hdf5')
model.summary()

columnsInDataSet = ['time', 'SHT_T', 'SHT_H', 'MLX_TA', 'MLX_TO', 'PT1000A', 'PT1000B', 'MEAS_FREQ', 'MEAS_FREQ.1',
'MEAS_FREQ.2', 'MEAS_COL_R', 'MEAS_COL_G',
'MEAS_COL_B', 'MEAS_COL_C', 'MEAS_TVOC', 'MEAS_eCO2', 'MEAS_rawH2', 'MEAS_rawEthanol', 'MEAS_weight', 'MEAS_A_0',
'MEAS_A_1', 'MEAS_A_2',
'MEAS_A_3', 'MEAS_BOARD', 'MEAS_UV', 'MEAS_IR', 'ABS_HUMI', 'Classification']

columnsToDrop = ['time', 'MEAS_FREQ', 'MEAS_IR', 'MEAS_UV', 'MEAS_BOARD', 'Classification']
columnNamesNotToNormalize = []

df = pd.DataFrame( columns=columnsInDataSet)
df.head()

scaler = joblib.load('/home/b/ws_python/tdata_scaler.gz')
pipeOut = open( FIFO_OUT, "w" )

pipeOut.write(LOADED)
pipeOut.flush()

predsInSessCnt = 0
predsSinceStartCnt = 0
perfectArrivedCnt = 0
tooLateArrivedCnt = 0

with open(FIFO_IN) as pipeIn:
    for line in pipeIn:
        if len(line) == len(RESET):
            if line == RESET:
                perfectArrivedCnt = 0
                tooLateArrivedCnt = 0
                predsInSessCnt = 0
                print("RESET!")
                df = df[0:0]
            else:
                values = [i for i in line.strip().split(',') ]
                sys.stdout.flush()
                df.loc[len(df)] = values
                dfToPred = df.drop(columns=columnsToDrop)
                columnNamesToNormalize = [x for x in list(dfToPred) if x not in columnNamesNotToNormalize ]
                dfToPred[columnNamesToNormalize] = scaler.transform ( dfToPred[columnNamesToNormalize] )
                zeroLineCntToAdd = zeroPaddedLength - dfToPred.shape[0]
                frameOfZeros = pd.DataFrame( zeroLineCntToAdd*[0]*dfToPred.shape[1], columns=dfToPred.columns)
                dfToPred = dfToPred.append(frameOfZeros)
                xToPred = dfToPred.values
                prediction = np.round(model.predict(np.array( [xToPred,] )))
                prediction = np.argmax(prediction)
                #pipeOut.write( str(prediction) + "\n" )

                if prediction == PERFECT:
                    perfectArrivedCnt = perfectArrivedCnt+1
                elif prediction == TOO_LATE:
                    if perfectArrivedCnt > 0:
                        tooLateArrivedCnt = tooLateArrivedCnt+1

                if perfectArrivedCnt > 10 or (perfectArrivedCnt+tooLateArrivedCnt) > 10:
                    pipeOut.write( str(1) + "\n" )

                predsInSessCnt = predsInSessCnt + 1
                predsSinceStartCnt = predsSinceStartCnt + 1
                #print(line)
                if predsSinceStartCnt > 0:
                    print('\r', end="")
                print ( predsInSessCnt, "-th Prediction :", PREDICTIONS[prediction] )

```

```
pipeOut.flush()  
print("Dead Predictor")
```