



UNIVERSITÄT ZU LÜBECK

**Aus dem Institut für Neuro- und Bioinformatik
der Universität zu Lübeck
Direktor: Prof. Dr. rer. nat. Thomas Martinetz**

**Echtzeitfähiges Deep-Learning-Klassifikationssystem für die Flora im
Bioanbau**

Inauguraldissertation
zur
Erlangung der Doktorwürde
der Universität zu Lübeck

Aus der Sektion Informatik / Technik

vorgelegt von
Florian Johannes Knoll
aus Husum

Lübeck, 2020

1. Berichterstatter:

2. Berichterstatter:

Tag der mündlichen Prüfung:

Zum Druck genehmigt. Lübeck, den

I. Vorwort

Diese Dissertation entstand aus dem Bestreben der Westhof-Bio GmbH heraus, eine Lösung für das Jäten ihrer Felder zu finden. So trat der Westhof an die Fachhochschule Westküste (FHW) heran und es entstand eine ertragreiche Kooperation, welche durch Fördergelder aus dem Exzellenz- und Strukturbudget des Landes Schleswig-Holstein unterstützt wurde. Rasch wurde ebenfalls die Universität zu Lübeck als Betreuung meiner Dissertation gewonnen, wofür ich mich an dieser Stelle ganz herzlich für die intensive Unterstützung bei Prof. Dr.-Ing. Erhardt Barth bedanken möchte. Mein Dank gilt ebenfalls Prof. Dr.-Ing. Alfred Mertins und Prof. Dr. Philipp Rostalski für ihre Bereitschaft, sich als Prüfer zur Verfügung zu stellen.

Mein besonderer Dank gilt Prof. Dr.-Ing. Stephan Hußmann, welcher mir die ganzen Jahre jederzeit, auch während seiner Zeit als Dekan, unterstützend zur Seite stand. Das gesamte Projekt ist durch sein ausdauerndes Einwerben der finanziellen Ressourcen zustande gekommen und seine Hilfe bei meinen Veröffentlichungen der verschiedenen Paper und Journals sei hier besonders erwähnt. Ich danke dir!

Hervorheben möchte ich noch drei Studenten, welche zu geschätzten Mitarbeitern und Diskussionspartnern wurden. Häufig kristallisierten sich durch eben diese Diskussionen brauchbare Lösungsansätze heraus. Einer von ihnen ist Vitali Czymmek, welcher mit seiner Abschlussarbeit die erste Version des GPU-Codes entwickelte und später dem Projekt als weiterer Mitarbeiter beitrug. Weiterhin Leif Ole Harders, der im späteren Verlauf ebenfalls zum Projekt hinzutrat, sowie Sascha Poczhoski, der mit seiner Abschlussarbeit eines der ersten Convolutional Neural Networks („künstliche neuronale Netze“) im Projekt effizient trainierte. Auch allen anderen studentischen Hilfskräften sei hier nochmals gedankt.

Natürlich sei an dieser Stelle meine Dankbarkeit der FHW gegenüber erwähnt, welche durch die Bereitstellung der Infrastruktur vieles erleichterte. Ebenfalls soll den Mitarbeitern des Fachbereichs Technik für ihre stetige Diskussionsbereitschaft gedankt werden.

Dem Westhof, im Besonderen Rainer Carstens und Paul-Heinrich Dörscher, danke ich für die Bereitstellung der landwirtschaftlichen Ressourcen und für das mir entgegengebrachte Vertrauen, eine adäquate Lösung für ihre Problematik zu finden. Außerdem danke ich Herrn Carstens und Herrn Dörscher für die Chance das Projekt in der neu gegründeten Firma weiterzuentwickeln.

Im besonderen Maße möchte ich meiner Mutter Heinke Knoll und meiner Großmutter Uta Misdorf meinen Dank dafür aussprechen, dass sie jede Gelegenheit genutzt haben, meine Bildung zu fördern, so dass ich die Möglichkeit erhielt, eine Dissertation zu schreiben. Weiterhin danke ich meinem Bruder Fabian Dries und meiner Schwägerin Magdalena Dries, sowie Jan Misdorf und Pascal Tams für die Hilfe beim abschließenden Korrekturlesen. Ein weiterer Dank gebührt meinem Onkel Reiner Misdorf und meinem Bruder Tim Thor-Straten.

Zu guter Letzt möchte ich meiner liebevollen Frau Sonja Knoll einen großen Dank aussprechen. Ich werde Sonja nie vergessen, dass sie mir mit ihrer Zeit und Unterstützung trotz der Geburt unserer bezaubernden Tochter Clara Victoria stets die Freiräume erarbeitet hat, die ich für diese Dissertation benötigte.

Vielen Dank euch beiden.

Florian Johannes Knoll, 2020

Meiner Familie

II. Inhaltsverzeichnis

1.	Einleitung.....	6
1.1	Motivation	6
1.2	Stand der Forschung.....	7
1.3	Aufbau der Arbeit.....	8
1.4	Notationen	9
2.	Grundlagen.....	10
2.1	Forschungsumgebung Feld.....	10
2.1.1	Karottenanbau.....	10
2.1.2	Die Schwierigkeit der Forschungsumgebung	11
2.2	Künstliche Neuronale Netze	12
2.2.1	Allgemeiner Aufbau eines künstlichen neuronalen Netzes	12
2.2.2	Künstliches neuronales Netz als Schwellwert-Element.....	14
2.2.3	Trainieren von Netzen	19
2.2.4	Bilder als Eingabeinformation im MLP	26
2.2.5	Convolutional Neural Network.....	27
2.3	Statistische Methoden zur Untersuchung von Klassifikatoren	31
3.	Sensorauswahl und Vorverarbeitung.....	33
3.1	Einleitung.....	33
3.2	Grundkonzept.....	33
3.3	Die Vernichtungseinheit.....	34
3.4	Das Trägersystem der Sensoren.....	35
3.4.1	BoniRob	35
3.4.2	Sensorhalterung	36
3.5	Die untersuchten Sensoren und Konfigurationen.....	37
3.5.1	Bispektral-Kamera JAI.....	38
3.5.2	RGB Kamera.....	38
3.5.3	Lichtfeld Kamera.....	39
3.5.4	Cam Cube III.....	39
3.5.5	Argos 3D P100	40
3.5.6	Kinect II	40
3.5.7	LMI Gocator 2350.....	41
3.5.8	Fazit zu den Sensoren.....	42
3.6	Vorverarbeitung der Sensorwerte.....	43
3.6.1	Ground Truth Erstellung für die Segmentierung.....	43

3.6.2	Die verwendeten Auswertungen für die Segmentierung.....	46
3.6.3	Einfacher globaler Schwellwert auf dem grünen Kanal	48
3.6.4	Vegetationsindexverfahren	51
3.6.5	RGB Segmentierungsalgorithmus.....	56
3.6.6	HSV-Verfahren	62
3.6.7	Multilayer Perceptron pixelweise	71
3.6.8	Multilayer Perceptron flächenweise	80
3.6.9	Zusammenfassende Auswertung der Segmentierung	89
3.6.10	Fazit zu der Segmentierung.....	95
3.7	Fazit zur Sensorauswahl und Segmentierung	96
4.	Der Klassifikator.....	97
4.1	Einleitung.....	97
4.2	Ground Truth Erstellung und Aufteilung des Datensatzes für die Klassifikation	98
4.3	Grundkonzept zu den Klassifikatoren.....	101
4.4	Die verwendeten Auswertungen für die Pflanzenklassifikation	102
4.5	Die untersuchten Pflanzen-Klassifikatoren	103
4.5.1	Multilayer Perceptron-Netz zur Pflanzenklassifikation	103
4.5.2	Convolutional-Neural-Network zur pixelweisen Pflanzenklassifikation.....	109
4.5.3	GoogLeNet.....	116
4.5.4	Alex-Netz	119
4.5.5	Area-Netz.....	123
4.5.6	Auswertung der Klassifikatoren	133
4.5.7	Fazit zu den Klassifikatoren	137
4.6	Optimierung des Klassifikators.....	138
4.6.1	Berechnungsoptimierung.....	138
4.6.2	Hardware Optimierung / Umsetzung im FPGA als CPU	143
4.6.3	Hardware Optimierung / Umsetzung in der GPU.....	149
4.6.4	Optimierung der Netzwerkstruktur.....	155
4.6.5	Abschließende Auswertung zur Optimierung	161
4.6.6	Fazit zur Optimierung	174
4.7	Fazit zum Kapitel der Klassifikatoren.....	174
5.	Zusammenfassung und Ausblick	176
6.	Abbildungsverzeichnis	178
7.	Tabellenverzeichnis	184
8.	Literaturverzeichnis	186

1. Einleitung

1.1 Motivation

Für die Herstellung von biologischen Nahrungsmitteln dürfen keine Herbizide (Pflanzenschutzmittel) eingesetzt werden, weshalb das Beikraut (umgangssprachlich Unkraut) von Hand entfernt werden muss. Der Kooperationspartner, die Westhof-Bio GmbH besitzt allein für die Nutzpflanze der Karotte circa 170 ha. Damit diese Fläche in der Saison gejätet werden kann, werden hier über 100 Hilfsarbeiter benötigt. Die Kosten belaufen sich dabei auf über 170.000 €. Durch den wachsenden Lebensstandard herrscht jedoch ein Mangel an Hilfskräften für das Jäten. Damit Bioprodukte kein Luxusgut werden, soll das Jäten automatisiert und mechanisch erfolgen. Im Rahmen dieser Dissertation soll eine künstliche Intelligenz erforscht und entwickelt werden, die die Nutzpflanze vom Beikraut unterscheidet. Im Rahmen dieses Projektes wurde sich für die Karottenpflanze entschieden, da diese im kleinen Wachstumsstadium ein ähnliches Aussehen wie das Beikraut besitzt und dadurch eine der schwierigsten Pflanzenklassifikationen hinsichtlich des Untersuchungsgebietes darstellt. Auch werden Karotten zufällig durch Streuen ausgesät, sodass durch ihre Position keine Rückschlüsse auf die Identifikation gezogen werden können. Während der Arbeit wurden zwar auch andere Kulturen, wie zum Beispiel Spinat oder Rote Bete erfolgreich getestet, diese sind aber nicht Teil dieser Untersuchung. Die Testumgebungen sind Felder, die im normalen landwirtschaftlichen Betrieb involviert sind. Daher stellt die in dieser Arbeit erforschte Lösung keine Laborlösung dar, sondern wurde unter realistischen Bedingungen entwickelt und getestet. Dieser Umstand erschwerte jedoch auch die Forschungsarbeit. Denn am Anfang des Projektes existierten keine Daten oder Messaufbauten. Daher musste zunächst eine Datenbank erstellt werden, um mit der eigentlichen Klassifikatorforschung zu beginnen. Für das Erstellen der Datenbank eignen sich jedoch nur wenige Wochen im Jahr, da nur in diesen Wochen die Karotten auf dem Feld ausgebracht werden. Versuche, eine künstliche Versuchsumgebung im Labor zu schaffen, scheiterten. Jedes Mal wenn etwas unplanmäßig verlief, sei es ein defektes Bauteil am Sensor oder starker Regen, verkürzte sich die Messzeit. Auch musste während des Messens sichergestellt werden, dass genügend Messproben von einem speziellen Fall aufgenommen worden sind, da eine Rekonstruktion des Falles nach der Messsaison nicht möglich ist und für die Entwicklung einer Problemlösung dann eventuelle Datensätze fehlen, die erst in einem Jahr erneut erfasst hätten werden können.

Bei dem Projekt konnte wegen der begrenzten finanziellen Mittel im Wesentlichen nur ein Mitarbeiter (der Autor) beschäftigt werden. Die Ergebnisse und Untersuchungen dieser Dissertation wurden bereits in über 15 internationalen Veröffentlichungen präsentiert. Davon wurden sechs internationale Konferenzpaper, sowie drei Journals direkt von dem Autor publiziert. Weiter konnte ein internationales Buchkapitel veröffentlicht und die Erfolge der Dissertation in nationalen Medien präsentiert werden. Es wurden 23 Abschlussarbeiten vergeben, die vom Autor betreut wurden. Die 17 Bachelor- und sechs Masterthesen konnten jedoch nicht vollständig in diese Arbeit einfließen. Die verwendeten Arbeiten setzen dabei die Ideen und Lösungsansätze des Autors dieser Dissertation um. Abschließend sei noch erwähnt, dass drei Patente national [Patent #1, 2020], [Patent #2, 2020] und [Patent #3, 2020] sowie zwei Patente international [Patent #4, 2020] und [Patent #5, 2020], die die verschiedenen Aspekte dieser Arbeit beinhalten, eingereicht wurden und derzeit das allgemeine Prüfverfahren durchlaufen.

1.2 Stand der Forschung

Die Idee des maschinellen Jätens hat schon eine längere Tradition, wurde jedoch wegen der Komplexität bisher lediglich auf eine Unterscheidung von größeren Pflanzen beschränkt. In den 1990ern existierten bereits verschiedene Versuche zu diesem Thema. So hat die Universität Osnabrück beispielsweise eine Abschlussarbeit über die Unterscheidung zwischen Mais und Beikraut betreut [Klamt, 1998]. In der Dissertation von Horst Oebel wurde ebenfalls versucht, Pflanzen, unter anderem Mais und Zuckerrüben, auf landwirtschaftlichen Flächen zu unterscheiden [Oebel, 2006]. Auch eine Anwendung einer Wavelet-Analyse zur Klassifikation in Maiskulturen ist in [Kargar B. & Shirzadifar, 2013] beschrieben. Die Verwendung von Multi-Support Vector Machine, um Pflanzen in Gruppen einzuteilen und zu klassifizieren, wurde in [Wong, et al., 2014] veröffentlicht. Das automatische Jäten muss jedoch hiervon stark unterschieden werden. Es gibt verschiedene Ansätze für das Jäten sowohl zwischen den Reihen, als auch in der Reihe. Das Jäten zwischen den Reihen benötigt keine Unterscheidung zwischen verschiedenen Pflanzen, da hier jede Pflanze entfernt werden muss. Diese Dissertation beschäftigt sich dagegen mit dem Jäten in der Reihe, wobei ebenfalls eine Unterscheidung berücksichtigt werden muss. So existieren grobe Pflanzen wie Kohl und Zuckerrüben oder Pflanzen, die gezielt mit einem festen Abstand ausgebracht werden, wie zum Beispiel Mais. Anders als die Kultur der Karotte in dieser Untersuchung, kann die Unterscheidung von Kohl, Zuckerrüben oder Mais relativ einfach erfolgen.

Eine Veröffentlichung der Firma Bosch, die das Jäten in Karottenfeldern untersucht, ist kurz vor Beginn dieser Untersuchung erschienen [Haug, et al., 2014]. Eine weitere Entwicklung wurde jedoch später wieder eingestellt, da die Klassifikation nicht schnell genug erfolgte. Es wurde ein Randomforest Klassifikator eingesetzt, der für die Suche der verschiedenen Features bereits eine immense Rechenzeit beansprucht. Darüber hinaus wurden die Karotten in deutlich größerem Abstand gesät und haben damit auch weniger Realitätsnähe als in der hier vorliegenden Untersuchung.

Ein anderes Projekt, das zeitgleich und vollkommen unabhängig von dieser Dissertation präsentiert wurde, entstammt der Universität Rom. Das Projekt verfolgt einen ähnlichen Deep Learning Ansatz wie diese Untersuchung. Dabei werden jedoch Zuckerrüben als Nutzpflanze verwendet [Cicco, et al., 2017], [Potena, et al., 2017], welche größer als Karotten sind und dabei geordneter auf dem Feld liegen. Die veröffentlichten Klassifikationsergebnisse für die pixelweise Klassifikation zeigen eine Genauigkeit von circa 92%. Die benötigte Rechenzeit für ein Bild von 1296 x 966 Pixel liegt bei circa 2 - 5 Sekunden. Der Algorithmus klassifiziert dabei nicht jeden Pixel, sondern arbeitet in einem Raster von der Größe von 3 x 3 Pixel. Ein anderes vorgestelltes Verfahren klassifiziert bestimmte zusammenhängende Regionen. Hier ist die Genauigkeit zwischen 95 - 98% angegeben und die Verarbeitungszeit liegt zwischen 1 - 2 Sekunden pro Bild [Potena, et al., 2017].

Der Markt für Smart-Farming wächst stetig, so wird im Jahr 2018 eine Marktgröße von etwa sieben Milliarden Euro weltweit angegeben [statista.com, 2019]. Smart-Farming bedeutet in diesem Sinne nicht nur das Jäten von Unkraut, sondern beschreibt eine allgemeine digitale Revolution in der Landwirtschaft.

Diese Arbeit präsentiert die erfolgreiche Erforschung und Entwicklung einer Klassifikationsmethode für die Unterscheidung zwischen Nutzpflanze und Beikraut. Zunächst werden an entscheidenden Punkten verschiedene Lösungsansätze präsentiert und untersucht. Dafür wurden verschiedene statistische Verfahren verwendet, um die Auswertungen der verschiedenen Bestandteile dieser Arbeit mit all ihren Modifikationen aus verschiedenen Blickwinkeln zu betrachten. Aufgrund der Ergebnisse wird ein Ansatz ausgewählt und weiter entwickelt. Sie stellt damit eine Lösung dar, die über ein experimentelles Stadium hinausgeht. Die Dissertation beginnt, basierend auf der Masterarbeit des Autors [Knoll, 2014], mit einer allgemeinen Projektbeschreibung des Problems, des Trägersystems und der Kalibrierung zwischen Sensor- und Vernichtungseinheit, welche in einem IEEE Transaction Journal [Knoll, et al., 2016 a] veröffentlicht werden konnte. Anschließend wird mit der Sensorauswahl und Untersuchung des Messaufbaus fortgeführt, wobei die Ergebnisse auf der internationalen IEEE

Konferenz [Knoll, et al., 2016 b] präsentiert wurden. In dieser Dissertation werden außerdem neuentwickelte Segmentierungsalgorithmen vorgestellt, die im Elsevier Journal [Knoll, et al., 2018] veröffentlicht werden konnten und Bestandteil der nationalen Patente [Patent #1, 2020], [Patent #2, 2020], [Patent #3, 2020] und der internationalen Patente [Patent #4, 2020], [Patent #5, 2020] sind. Diese Algorithmen werden umfänglich mit dem State-of-the-Art NDVI-Verfahren [Haug, et al., 2014], [Matsushita, et al., 2007], [Oebel, 2006] verglichen. Die Ergebnisse zeigen, dass die neuen Segmentierungsalgorithmen nicht nur dieselben oder bessere Ergebnisse aufzeigen, sondern auch mit herkömmlichen RGB-Kameras arbeiten. Das NDVI-Verfahren benötigt eine Bisppektralkamera, die neben einem RGB-Bild auch ein Bild im nahen Infrarotbereich benötigt. Als Klassifikatoren wurden in dieser Arbeit verschiedene Deep-Learning-Netzwerke konstruiert und untersucht. Es stellte sich heraus, dass sie bessere Ergebnisse aufweisen als andere Klassifikatoren in diesem Bereich [Szegedy, et al., 2015], [Michaels, et al., 2015], [Haug, et al., 2014] und [Krizhevsky, et al., 2012]. Auch diese Ergebnisse konnten im Elsevier Journal [Knoll, et al., 2018] vorgestellt werden. Ein großer Erfolg dieser Arbeit ist die zu erreichende Klassifikationsgeschwindigkeit, die durch eine neue Verarbeitungsarchitektur ermöglicht wird. Diese Architektur wurde dabei unter anderem im Elsevier Journal [Knoll, et al., 2019] präsentiert. Die Verarbeitungsstruktur ist in einer eigens entworfenen CPU, sowie als GPU-Umsetzung aufgebaut und untersucht worden. Die Genauigkeitsergebnisse konnten durch ein in dieser Arbeit vorgestelltes angepasstes Lernen der Netzwerke abschließend noch verbessert werden. Die Klassifikatoren und die Verfahren zur Geschwindigkeitsoptimierung sind ebenfalls Bestandteil der nationalen Patente [Patent #1, 2020], [Patent #2, 2020], [Patent #3, 2020] und der internationalen Patente [Patent #4, 2020], [Patent #5, 2020].

1.3 Aufbau der Arbeit

Der Aufbau dieser Dissertation und die realen Entwicklungsschritte sind nicht immer identisch gewesen. Aus didaktischen Überlegungen wurde die Arbeit so angeordnet, wie nachfolgend präsentiert.

Die Dissertation besitzt neben dem Grundlagenkapitel zwei Hauptkapitel. Jedes Hauptkapitel unterteilt sich wieder in zwei Bereiche. Dabei wird zu jedem Bereich eine kurze Einleitung gegeben, in der die grundlegenden Fragen für diesen Bereich erklärt werden. Weiter folgen am Ende eine Gegenüberstellung aller untersuchten Varianten sowie ein kurzes Fazit.

Hauptkapitel I: Sensorauswahl und Vorverarbeitung

Der erste Bereich des Kapitels beschäftigt sich kurz mit dem Trägersystem sowie einer Evaluation und Auswahl eines Sensorsystems für die Klassifikation. Im anschließenden zweiten Bereich folgt eine Untersuchung verschiedener Ansätze zur Segmentierung der Pflanzen von dem Hintergrund.

Hauptkapitel II: Der Klassifikator

Im ersten Bereich werden verschiedene Klassifikatoren auf Basis der Deep Learning Netzwerke für die Pflanzenklassifikation untersucht. Im zweiten Bereich werden Verbesserungen der Klassifikation entwickelt. Dies erfolgt einerseits durch eine Optimierung der ausgewählten Klassifikatoren, die eine echtzeitfähige Verarbeitung gewährleisten soll, sowie andererseits durch eine Anpassung des selbstlernenden Netzwerkes.

Die Dissertation endet abschließend mit einer Zusammenfassung und einem Ausblick.

1.4 Notationen

Es handelt sich bei dieser Arbeit um eine deutschsprachige Dissertation. Trotzdem kann es vorkommen, dass englische Begriffe verwendet werden, obwohl eine deutsche Übersetzung existiert. Dies hat den Grund, dass in der technischen Welt vor allem englische Begriffe verwendet werden. Es wäre zwar möglich, zum Beispiel für den Begriff Convolution Layer die deutsche Übersetzung Faltungsschicht zu verwenden, jedoch erhöht dies nicht das Verständnis des Lesers.

Außerdem verwendet der Autor Fußnoten, um die Arbeit mit weiteren Erläuterungen und Erklärungen zu versehen. Es werden jedoch auch vereinzelt Fußnoten verwendet, um direkter auf eine Quelle einzugehen.

2. Grundlagen

2.1 Forschungsumgebung Feld

Für dieses Forschungsumfeld existierten bisher keine Datengrundlagen sowie Methoden zur Generierung dieser Grundlagen. Um die Erforschung eines Klassifikators durchzuführen, müssen daher die Daten erst aufgenommen und aufbereitet werden. Das Sammeln der Daten stellt demnach einen wichtigen Bestandteil dieser Arbeit dar und wird entsprechend im späteren Verlauf dieser Ausarbeitung genauer beschrieben. Zum besseren Verständnis soll hier kurz erläutert werden wie der Anbau von Karotten in diesem Projekt vollzogen wird. Die Datenaufzeichnung geschah dabei auf tatsächlich genutzten landwirtschaftlichen Flächen. Die dadurch entstandenen Probleme sollen anschließend beschrieben werden.

2.1.1 Karottenanbau

Die Forschungs- und Messumgebungen sind Felder des Kooperationspartners Westhof-Bio GmbH (im Folgenden als Westhof benannt) und sind dabei in die landwirtschaftliche Bewirtschaftung integriert. Der Westhof baut die Bio-Karotten auf Erdwällen wie in Abbildung 1 an. Das Anbauen geschieht dabei in folgenden Schritten. Zuerst werden Erdwälle angehäuft, welche als trapezförmig beschrieben werden können. Ihre Höhe liegt bei 15 cm und die obere Breite bei circa 20 cm. Die Wälle haben einen Abstand von etwa 0,75 m von Mittelpunkt zu Mittelpunkt. Diese Angaben sind jedoch keine exakten Werte, da diese durch die Bodenbeschaffenheiten und die Anbaufahrzeuge variieren können. Es folgt anschließend das Ausbringen der Karottensamen. Diese werden auf der oberen Kante des Trapezes in einem 12 cm breiten Streifen gestreut und stellen damit ein zufälliges Muster dar. Auf einem Meter Walllänge befinden sich dabei circa 120 Karottensamen, welche nach etwa 10 Tagen zu sprießen beginnen. Um das Vorkommen von Beikraut (Unkraut) zu verringern und alle Pflanzen in etwa dasselbe Wachstumsstadium zu bringen, werden alle bis dahin gewachsenen Pflanzen mit einer Flamme entfernt. Nach einigen Wochen erfolgt ein manuelles Jäten, indem eine Gruppe von Hilfsarbeitern das Beikraut mit den Händen vom Feld entfernt.

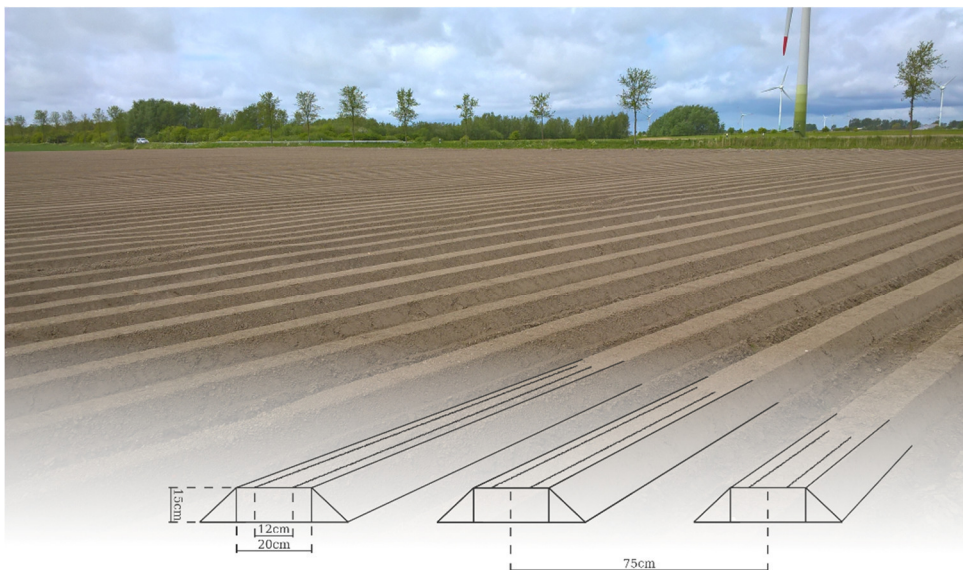


Abbildung 1: Darstellung eines Karottenfeldes kurz vor dem Auflaufen der Pflanzen

2.1.2 Die Schwierigkeit der Forschungsumgebung

Die Forschungsumgebung als reales Areal bietet einige Vor- sowie Nachteile, welche im Folgenden kurz beschrieben werden.

Ein Vorteil der Messumgebung ist, dass es sich um ein reales, landwirtschaftlich genutztes Feld handelt, welches die Chance bietet, keine Laborlösung, sondern eine reell nutzbare Lösung zu finden. Diese Lösung ist somit direkt am Feld erforscht und erprobt worden, wodurch Problematiken im natürlichen Raum unmittelbar erfasst und gelöst werden konnten.

Durch diese Messumgebung ergeben sich jedoch auch verschiedene Problematiken, welche nachfolgend diskutiert werden.

Eines der größten Probleme stellt das Zeitfenster in dem gemessen werden kann dar, da die Karotten nur in einem Zeitraum von 2-3 Monaten im Jahr angebaut werden. Dadurch, dass nicht jedes Wachstumsstadium der Pflanzen für das Projekt interessant ist, kürzt sich die gesamte Zeit auf wenige Wochen im Jahr, wodurch jede Messung im Projekt genau geplant und in sich abgeschlossen sein muss. Für auftretende Problemstellungen beziehungsweise Klassifikationsfälle müssen in wenigen Wochen ausreichend Daten gewonnen werden, um diese im Zeitraum zwischen den Feldversuchen lösen zu können. Wenn nach der Messung beispielsweise bekannt wird, dass die Datenmenge nicht ausreicht, muss ein Jahr gewartet werden, um weitere Daten zu erheben. Im nächsten Jahr müssen dann zudem mehr Daten erhoben werden, um die bestehende Problematik zu lösen, da die Messungen von Jahr zu Jahr, beispielsweise durch wechselnde Klimabedingungen oder unterschiedliches Pflanzenwachstum nicht reproduzierbar sind. Durch die sich ständig ändernden Umweltbedingungen, beispielsweise starker Regen oder große Hitze, ergeben sich jedes Jahr immer wieder erneut andere Messreihen. So kam es, dass von einem Jahr auf das andere eine komplett andere Pflanze die Mehrheit auf dem Feld stellte, oder dass sich der Hintergrund massiv änderte, weshalb die Datengrundlage einen wachsenden Prozess darstellt, welcher einer ständigen Veränderung unterliegt. Mit jedem Jahr kommen neue Aspekte auf, die in die Datengrundlage einfließen müssen. Durch die raue Umgebung muss die gesamte Technik mit den verschiedenen Wettereinflüssen wie extremer Sonneneinstrahlung, Regen und Staub zurechtkommen. Ein weiteres Problem ist, dass diese Felder in den natürlichen landwirtschaftlichen Betrieb eingebunden waren, wodurch während der Messphasen auch Rücksicht auf die Arbeitsprozesse des Betriebes genommen werden musste.

Am Anfang des Projekts bestanden keine Datenbanken oder Daten aus anderen Projekten mit denen experimentiert werden konnte. Alle hier verwendeten Daten für die Erforschung und Entwicklung wurden unter diesen Bedingungen selbst gesammelt und ausgewertet.

2.2 Künstliche Neuronale Netze

Das künstliche neuronale Netz (KNN) ist der Versuch, die neuronalen Strukturen des Gehirns mathematisch so zu beschreiben, dass es möglich ist, dieses in einer Recheneinheit nachzubilden. Es handelt sich dabei um Strukturen, die mittels eines Lernalgorithmus auf eine Aufgabe trainiert werden können. Im nachfolgenden Kapitel wird eine eigene Zusammenfassung über den Aufbau eines Künstlichen Neuronalen Netzes geboten, welche die Ergebnisse und Erläuterungen aus den Publikationen von [Zell, 1994], [Kruse, et al., 2011], [Lämmel & Cleve, 2008] und [Wartala, 2018] wiedergibt. Diese hat dabei keinen Anspruch auf Vollständigkeit und soll lediglich einen hilfreichen Einblick geben, um den weiterführenden Teil der Arbeit verstehen zu können.

2.2.1 Allgemeiner Aufbau eines künstlichen neuronalen Netzes

Der Hauptbestandteil des künstlichen neuronalen Netzes bildet das Zusammenspiel zwischen den Gewichten und den Neuronen, wie in Abbildung 2 dargestellt. Dieses Zusammenspiel findet sich auch in der Natur wieder, jedoch ist es bei einem künstlichen neuronalen Netz stark idealisiert. Der Aufbau eines natürlichen neuronalen Netzes, wie dem Gehirn soll hier nicht weiter erläutert werden, ein genauerer Einblick wird jedoch in der Publikation von [Zell, 1994] geboten.

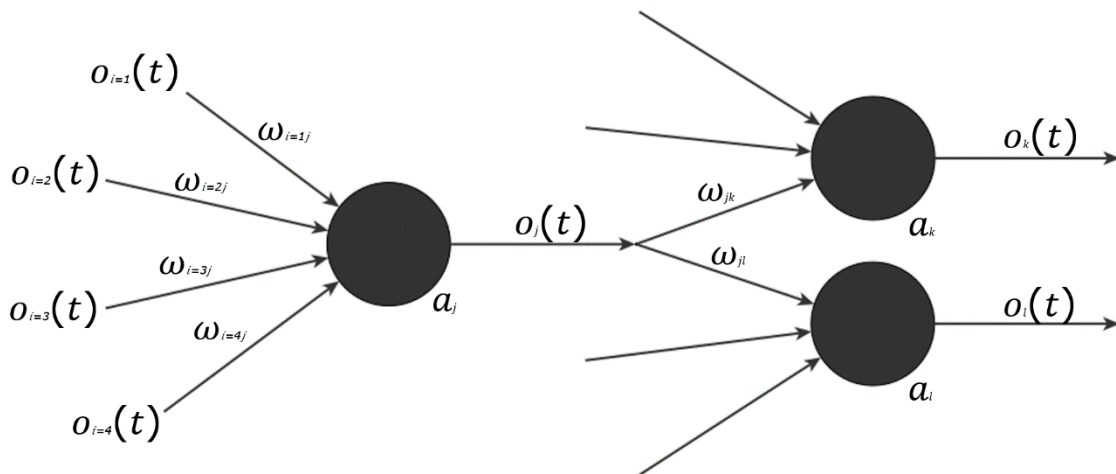


Abbildung 2: Arbeitsweise von Gewichten und Neuronen in einem künstlichen neuronalen Netz
inspiriert aus: [Zell, 1994]

Wie in Abbildung 2 zu sehen ist, gelangen die verschiedenen Signale $o_i(t)$ über ihre jeweiligen Gewichte ω_{ij} zu den Neuronen a_j . Die Indexe der Gewichte geben dabei an, zwischen welchen Neuronen dieses Gewicht liegt. Der erste Index ist das Sende-Neuron, der zweite Index ist das Empfangsneuron. Jedes Neuron besitzt seinen einzigartigen Index und die Signale $o_i(t)$ werden mit dem Index gekennzeichnet, von welchem Neuron das Signal kommt.

Ein Neuron selbst kann dabei als eine Abbildung eines N-dimensionalen reellen Raumes auf eine reelle skalare Größe betrachtet werden [Lämmel & Cleve, 2008]:

$$\mathbb{R}^N \mapsto \mathbb{R}. \quad (1)$$

Der erste Schritt der Signalverarbeitung eines Neurons erfolgt durch eine Multiplikation eines Signals $o_i(t)$ mit seinem individuellen Gewicht ω_{ij} . Anschließend summiert das Neuron diese Eingangssignale zu seiner Netzeingabe $net_j(t)$ auf. Dies kann mathematisch wie folgt beschrieben werden:

$$net_j(t) = \sum_{i=0}^N o_i(t) \cdot \omega_{ij}. \quad (2)$$

Danach wird das aufsummierte Signal auf eine Aktivierungsfunktion $f_{act}(\cdot)$ des Neurons gegeben und das Aktivierungssignal erzeugt. Die Aktivierungsfunktion kann dabei zum Beispiel eine lineare Funktion, einen Schwellwert oder eine Funktion mit einem sigmoiden Charakter haben. Einige Beispiele sind in Abbildung 3 dargestellt.

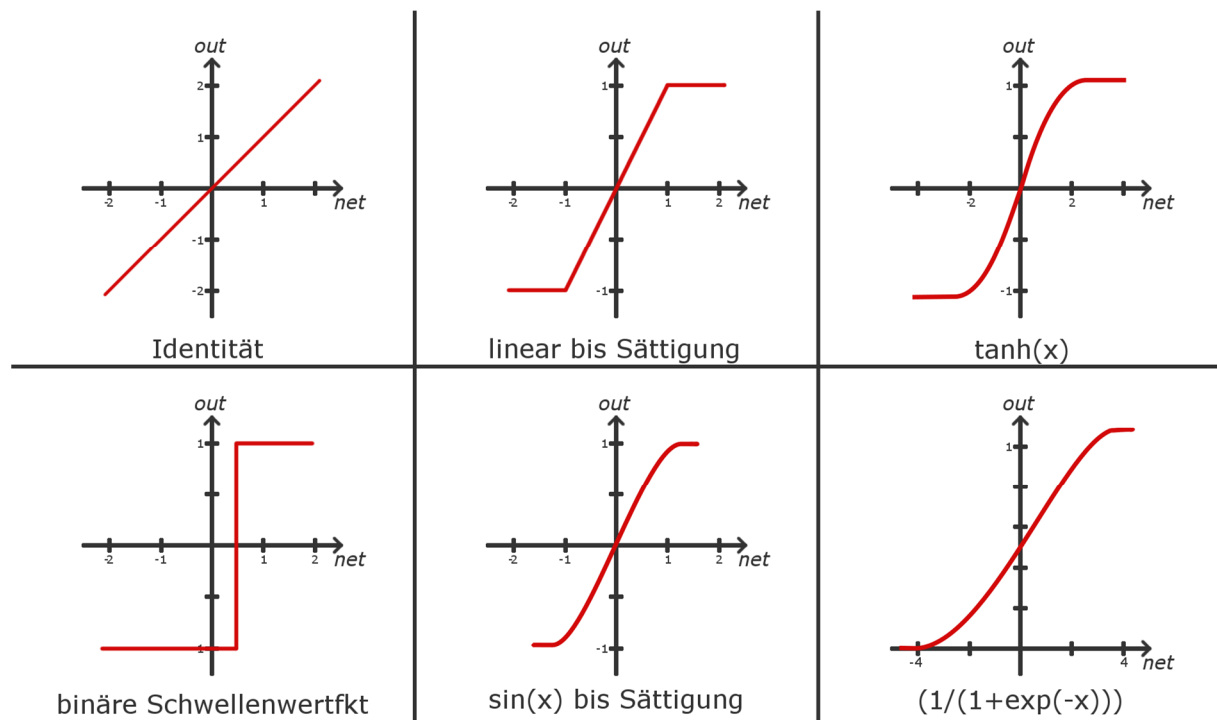


Abbildung 3: Aktivierungsfunktionen eines Neurons inspiriert aus: [Zell, 1994]

Häufig verwendete Funktionen sind zum Beispiel der Tangenshyperbolicus und die Sigmoid-Funktion, da diese dem natürlichen Original sehr nahekommen. Die Aktivierungsfunktion besitzt als Parameter die Netzeingabe $net_j(t)$, sowie die Schwellwertgröße θ_j .

$$a_j(t) = f_{act}(net_j(t), \theta_j). \quad (3)$$

Abschließend wird das Aktivierungssignal auf eine weitere Funktion, die Ausgabefunktion f_{out} , gegeben. Die Ausgabefunktion kann verwendet werden, um die Ausgabe $o_j(t)$ eines Neurons zu beschränken. Auch diese Funktion setzt sich aus zwei Parametern, dem Aktivierungssignal und einem Schwellwert ϑ_j zusammen, wie in Formel (4) zu sehen ist. Durch die häufig verwendeten Aktivierungsfunktionen, die bereits eine Beschränkung besitzen, wird in der Ausgabefunktion meistens die Identität verwendet.

$$o_j(t) = f_{out}(a_j(t), \vartheta_j). \quad (4)$$

Das Ausgangssignal gelangt durch weitere Gewichte erneut zu einem anderen Neuron, bis es das gesamte Netz durchlaufen hat. Das Modell des einzelnen Neurons soll dabei simpel gehalten werden. Die eigentliche Komplexität und die Stärke eines KNN entstehen in einem neuronalen Netz in seiner Netzstruktur. Dadurch, dass jedes Neuron für sich arbeiten kann, ist die massive Parallelisierung des

gesamten Netzwerks in Hinblick auf Geschwindigkeit und Informationsredundanz ein weiterer Vorteil. Das „Wissen“ eines Netzes wird dabei in den jeweiligen Gewichten und Schwellwerten der Aktivierungsfunktion gespeichert.

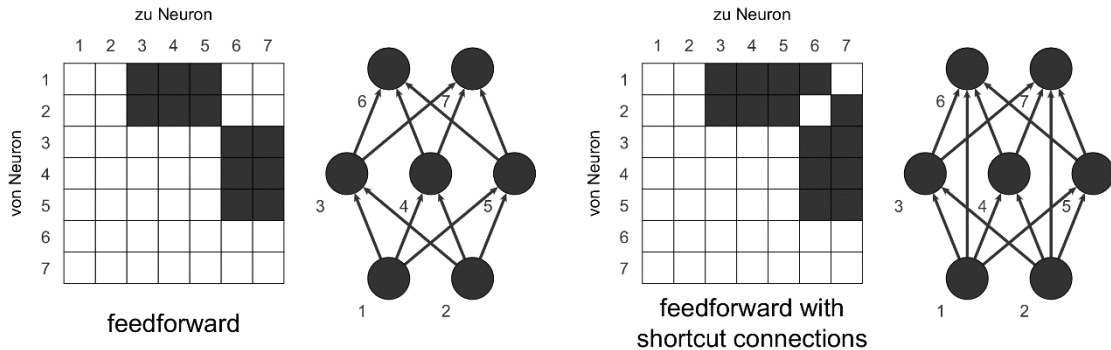


Abbildung 4: Beispiele der verwendeten Netztopografie inspiriert aus: [Zell, 1994]

Die Netzstruktur kann in drei Segmente eingeteilt werden. Die erste ist die Eingabeschicht (Input-Layer), über welche die Information in das Netzwerk gelangt. Hier findet, sofern dies nötig ist, eine Vorverarbeitung in Form einer Skalierung statt. In der verdeckten Schicht (Hidden-Layer) erfolgt die Informationsverarbeitung, deren Informationen entsprechend über die Ausgangsschicht (Output-Layer) aus dem Netzwerk gelangen. Dabei besitzen die hier verwendeten Netze eine „feedforward“ bzw. „feedforward mit shortcut connection“ Struktur, wie in Abbildung 4.

2.2.2 Künstliches neuronales Netz als Schwellwert-Element

Wird lediglich ein einzelnes Neuron betrachtet, kann dieses als ein Schwellwert-Element betrachtet werden, sofern es sich bei seiner Aktivierungsfunktion um eine Rechteckfunktion handelt. Das heißt, dass das zeitliche Ausgangssignal $out(t)$ den Wert 1 annimmt, sofern die Summe der Eingangssignale $x_i(t)$ multipliziert mit seinem Gewicht ω_i größer oder gleich eines Schwellwerts θ ist. Sollte der Schwellwert nicht erreicht werden, wird der Wert 0 ausgegeben. Abbildung 5 zeigt die schematische Darstellung eines Schwellwertelementes.

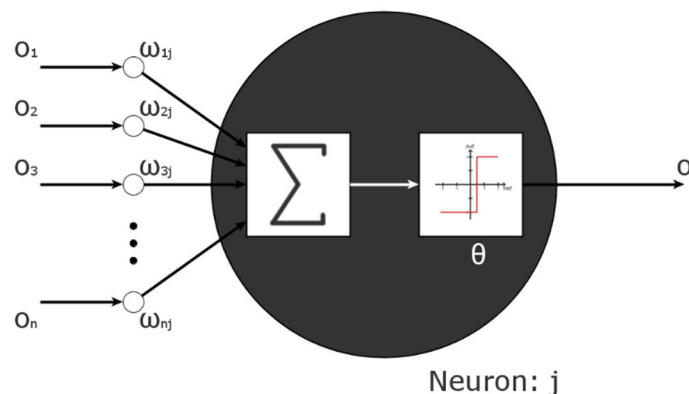


Abbildung 5: Schematische Darstellung eines Schwellwertelementes inspiriert aus: [Zell, 1994]

Mathematisch ergibt sich der Zusammenhang wie in der Formel (5) dargestellt.

$$out(t) = \begin{cases} 1, & \text{wenn } \sum_{i=0}^N \omega_i \cdot x_i(t) \geq \theta \\ 0, & \text{sonst} \end{cases} \quad (5)$$

Dieser kann auch als Skalarprodukt von Vektoren geschrieben werden, wie in Formel (6).

$$\vec{x} \cdot \vec{\omega} \geq \theta \Leftrightarrow \begin{pmatrix} x_0 \\ \vdots \\ x_N \end{pmatrix} \cdot \begin{pmatrix} \omega_0 \\ \vdots \\ \omega_N \end{pmatrix} \geq \theta. \quad (6)$$

Entsprechend kann der Zusammenhang verwendet werden, um zwei Mengen mit N Parametern zu trennen. Im Folgenden soll ein zweidimensionaler Fall betrachtet werden, der bereits von Kruse [Kruse, et al., 2011] erläutert wurde. In der Abbildung 6 sollen die weißen und schwarzen Kreise mit der roten Trennlinie voneinander separiert werden. Die Parameter x_0 und x_1 erlauben dabei eine eindeutige Lokalisierung der einzelnen Kreise in einem zweidimensionalen Raum.

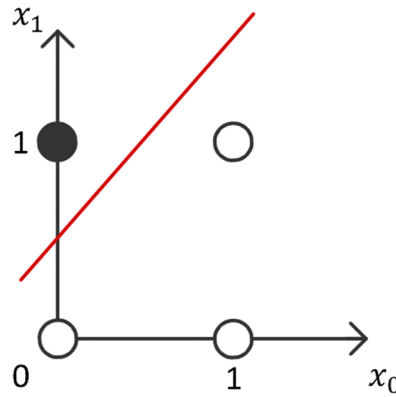


Abbildung 6: Anwendungsbeispiel eines Schwellenwertelements auf die Separierung zweier Mengen inspiriert aus: [Kruse, et al., 2011]

Zur Bestimmung des Schwellenwertelements wird die Geradengleichung benötigt, die die Trennlinie beschreibt. Die allgemeine Form einer Geradengleichung ist

$$x_1 = m \cdot x_0 + b, \quad (7)$$

mit m als Steigung und b als Ordinatenabschnitt. Für die in Abbildung 6 dargestellte Gerade, die durch die Punkte $P1 (0/0,5)$ und $P2 (1/1,5)$ verläuft, ergibt sich für die Steigung

$$m = \frac{\Delta x_1}{\Delta x_0} = \frac{1,5 - 0,5}{1 - 0} = 1 \quad (8)$$

und für den Ordinatenabschnitt

$$b = 0,5. \quad (9)$$

Damit ergibt sich für die Geradengleichung

$$x_1 = m \cdot x_0 + b \Leftrightarrow x_1 = 1 \cdot x_0 + \frac{1}{2}. \quad (10)$$

Durch Umformung der Gleichung ergibt sich die Parameterdarstellung

$$2 \cdot x_1 - 2 \cdot x_0 = 1. \quad (11)$$

Wird die Berechnung des Schwellenwertes aus Formel (5) erneut unter der Annahme betrachtet, dass nur diejenigen Parameter-Paare zulässig sind, die auf der roten Trennlinie aus Abbildung 6 liegen sollen, vereinfacht sich die Gleichung zu

$$\sum_{i=0}^N \omega_i \cdot x_i(t) = \theta. \quad (12)$$

Eine Gegenüberstellung von den Formeln (11) und (12) erlaubt es, die Gewichte und den Schwellwert durch Ablesen zu bestimmen:

$$\begin{aligned} \omega_0 &= -2 \\ \omega_1 &= 2 \\ \theta &= 1. \end{aligned} \quad (13)$$

Damit ergibt sich das Schwellwertelement, welches in Abbildung 7 gesucht wurde. Der Wert innerhalb des dargestellten Neurons gibt dabei seinen Schwellenwert θ an.

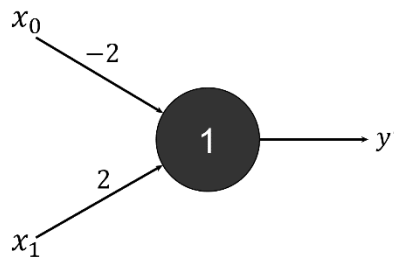


Abbildung 7: Darstellung des berechneten Schwellwertelements zum Separieren zweier Mengen
inspiriert aus: [Kruse, et al., 2011]

Der aus der Parameterdarstellung abzulesende Normalenvektor $\vec{n} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$ zeigt in die Richtung, in der das Schwellwertelement die 1 als Ausgabe erzeugt. In diesem Fall würde jede Kombination der Parameter x_0 und x_1 oberhalb oder auf der roten Trennlinie den Ausgangswert 1 hervorrufen.

Mit diesen Schwellwertelementen können nur lineare separierbare Mengen erzeugt werden. Sollte eine Trennung der weißen und schwarzen Kreise wie in Abbildung 8 erfolgen, reicht es nicht aus, nur ein einzelnes Neuron zu verwenden. Es handelt sich bei dem Problem um das logische Exklusiv-Oder (XOR) [Kruse, et al., 2011].

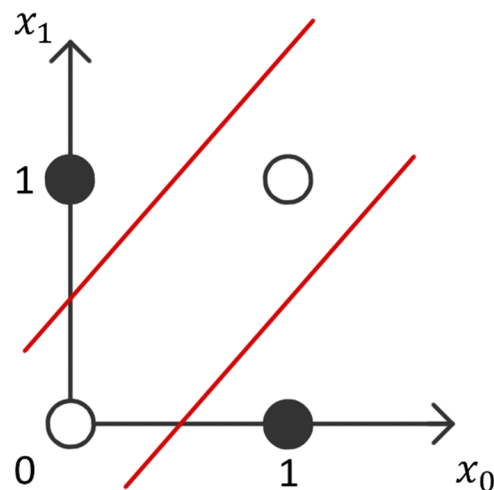


Abbildung 8: Nicht linear separierbare Mengen inspiriert aus: [Kruse, et al., 2011]

Eine Lösung des Problems ist es, die separierbaren Mengen einzeln zu betrachten. Für den schwarzen Kreis auf dem Punkt P (0/1) existiert bereits eine Lösung, wie sie in Abbildung 7 dargestellt ist. Für den Kreis auf dem Punkt P (1/0) kann eine analoge Lösung gefunden werden. Hierfür ergibt sich die Parameterdarstellung

$$-2 \cdot x_1 + 2 \cdot x_0 = 1. \quad (14)$$

Aus der Parameterdarstellung kann erneut das Schwellenwertelement abgelesen werden, wie es in Abbildung 9 dargestellt ist.

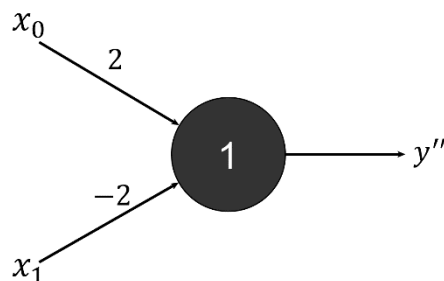


Abbildung 9: Schwellenwertelement für die Separierung des Kreises unten rechts inspiriert aus: [Kruse, et al., 2011]

Anschließend müssen die Schwellenwertelemente aus Abbildung 7 und Abbildung 9 lediglich mit einer Booleschen Operation verknüpft werden:

$$y = y' \vee y''. \quad (15)$$

In diesem Fall wäre es das logische Oder. Diese Operation kann mit einem Schwellenwertelement erzeugt werden, wie in Abbildung 10 dargestellt.

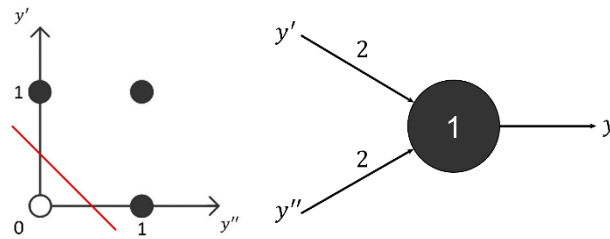


Abbildung 10: Umsetzung der Booleschen Operation oder als Schwellenwertelement (links). Darstellung der zu separierenden Mengen (rechts), Darstellung des Schwellenwertelementes inspiriert aus: [Kruse, et al., 2011]

Durch das Zusammensetzen der Einzellösungen erfolgt das in Abbildung 11 dargestellte Netz. Die Informationen x_0 und x_1 gelangen über die Eingabeschicht (graue Neuronen) in das Netz. Es folgt eine verdeckte Schicht aus dem grünen und dem blauen Neuron. Das grüne Neuron separiert den schwarzen Kreis unten rechts und das blaue Neuron den schwarzen Kreis oben links. Die logische Oder-Verknüpfung erfolgt über das rote Neuron, das auch die Ausgabeschicht darstellt.

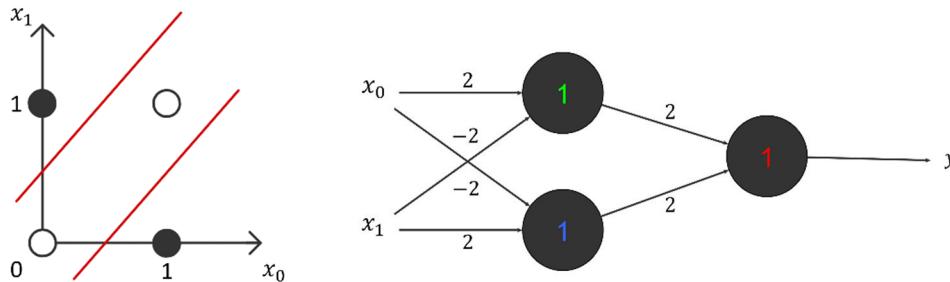


Abbildung 11: Zusammengesetztes Schwellenwertelement zur Separierung nicht linear separabler Mengen inspiriert aus: [Kruse, et al., 2011]

Mit diesen Netzen aus Schwellenwertelementen können auch komplexere Mengen voneinander getrennt werden. Das Netz aus Abbildung 11 besteht im Wesentlichen aus einer Eingabeschicht, einer verdeckten Schicht und einer Ausgabeschicht. Durch das Erweitern der verdeckten Schicht können komplexere Strukturen zur Segmentierung verwendet werden. Ein Beispiel ist in Abbildung 12.

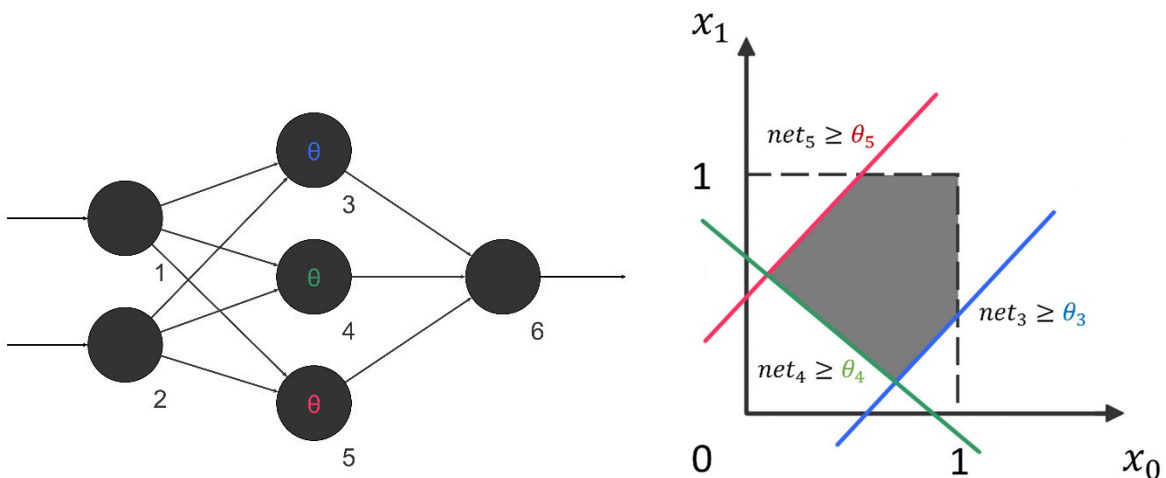


Abbildung 12: Schwellenwertelement zur Trennung komplexerer Mengen inspiriert aus: [Zell, 1994]

Die Neuronen 1 und 2 bilden erneut die Eingabeschicht, über die die Informationen zum Separieren in das Netz gelangen. Jedes Neuron der versteckten Schicht (Neuronen 3, 4 und 5) bildet jeweils eine Trennlinie aus Abbildung 12 b ab. Die Ausgaben der drei Neuronen geben binär an, ob ein Eingabewert oberhalb oder unterhalb der Trennlinie liegt. Mit der Verknüpfung dieser Information kann das Ausgabeneuron 6 bestimmen, ob ein Eingabewert innerhalb des zu separierenden Bereiches liegt oder nicht. Die Ausgabe erfolgt dann ebenfalls als binärer Wert. Diese Art der neuronalen Netze wird in der Literatur als multilayer perceptron (nachfolgend MLP) bezeichnet, auch wenn das Perzeptron kein einheitlich verwendeter Begriff ist [Lämmel & Cleve, 2008].

2.2.3 Trainieren von Netzen

Ein Vorteil der neuronalen Netze ist ihre Lernfähigkeit. Auch die Schwellenwertelemente können mittels Lernalgorithmen auf ihre Arbeit angepasst werden. Es wird dabei zwischen verschiedenen Arten von Lernverfahren unterschieden. So gibt es das überwachte Lernen, das bestärkende Lernen und das unüberwachte Lernen. Beim überwachten Lernen wird ein Eingabemuster mit seinem bekannten Ausgabemuster auf das Netz gegeben und überprüft, wie exakt die gewünschte Ausgabe berechnet wurde. Der Vorgang wird solange wiederholt, bis der Fehler des Netzes minimal wird. Es stellt das schnellste der Lernverfahren dar, entspricht jedoch nicht dem natürlichen Vorbild. Das bestärkende Lernen analysiert hingegen nicht den Grad der Richtigkeit einer Ausgabe. Es wird lediglich die Information verwendet, ob ein Ergebnis richtig oder falsch ist. Beim unüberwachten Lernen versucht das Netz selbständig, ähnliche Muster in passenden Strukturen abzubilden. Es ist das Verfahren, das dem natürlichen Vorbild am ehesten entspricht, ist jedoch auch nicht für alle Anwendungsfälle geeignet. Ein Anwendungsfall wäre unter anderem das Training von selbstorganisierten Karten [Zell, 1994].

Hier wird überwiegend das überwachte Lernen verwendet, da genau bekannt ist, welches Eingabemuster welches Ergebnis erzeugen soll und dieses Verfahren das schnellste ist. Ein künstliches neuronales Netz speichert sein „Wissen“ unter anderem in seinen Gewichten. Beim Lernen müssen daher diese Variablen angepasst werden. Ein Eingabemuster würde daher über die Gewichte ein Ergebnis erzeugen und aus dem Ergebnis kann durch einen Vergleich mit dem bekannten Soll-Ergebnis ein Fehler berechnet werden. Der Fehler kann als Funktion der Gewichte aufgefasst werden:

$$E(\Omega) = E(\omega_1 \dots \omega_N). \quad (16)$$

Die Gewichtsmatrix Ω hat dabei die Größe N^2 . Der Wert, der in der i. Zeile und j. Spalte in der Matrix steht, ist das Gewicht, das die Verbindung zwischen dem i. und j. Neuron darstellt.

$$\Omega := \begin{pmatrix} \omega_{11} & \cdots & \omega_{1N} \\ \vdots & \ddots & \vdots \\ \omega_{N1} & \cdots & \omega_{NN} \end{pmatrix}. \quad (17)$$

Werden alle Fehler aufsummiert, welche jedes Trainingsmuster bei der jeweiligen Gewichtspermutation erzeugt, kann dies wie in Abbildung 13 dargestellt werden.

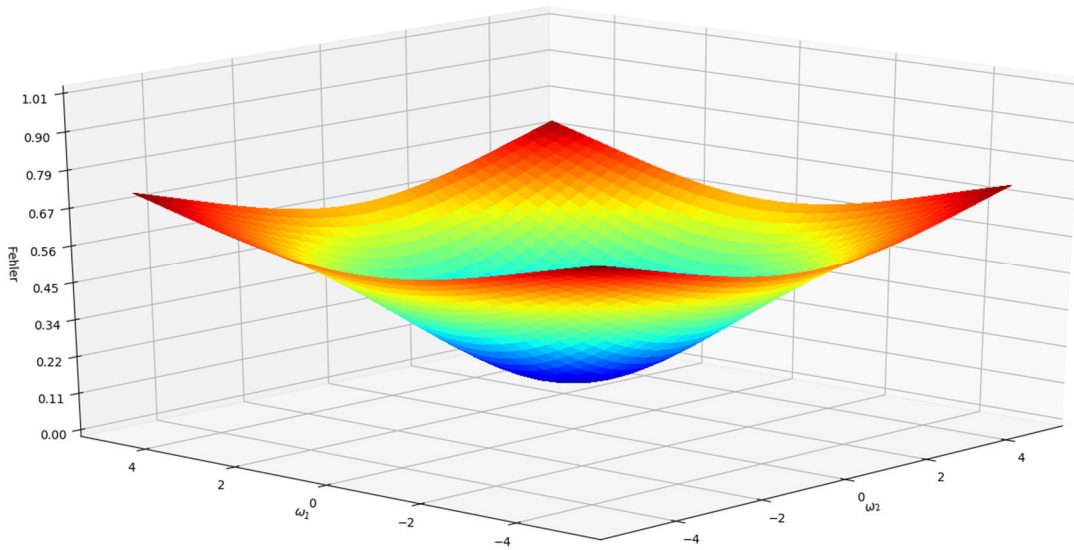


Abbildung 13: Darstellung des Fehlers eines Netzes bei der Variation von zwei Gewichten inspiriert aus: [Zell, 1994]

Das Ziel eines Lernalgorithmus ist es, die Einstellung der Gewichte zu finden, in dem der Fehler minimal wird [Lämmel & Cleve, 2008]. Ein künstliches neuronales Netz besteht jedoch meistens nicht nur aus zwei Gewichten, auch sind die Gewichte nicht die einzigen Parameter, die zum Lernen verwendet werden können. Auch die Schwellwerte der Aktivierungsfunktionen können unter anderem „Wissen“ enthalten. Dadurch, dass hier eine Vielzahl von Permutationen zustande kommt, ist es derzeit nicht möglich, die Permutation zu berechnen, in der das globale Minimum (der Punkt, an dem der Fehler des neuronalen Netzes am geringsten ist) erreicht wird. Helfen kann hier ein Gradientenabstiegsverfahren. Der Gradientenoperator kann durch die Ableitung der Fehlerfunktion die Richtung bestimmen, in der der Fehler kleiner wird. Dadurch ist dann auch bekannt, in welche Richtung die einzelnen Gewichte verändert werden müssen. Einer dieser Algorithmen ist das Backpropagation-Verfahren, das im Folgenden hergeleitet werden soll, wie in [Zell, 1994] und [Lämmel & Cleve, 2008] erläutert ist. Die allgemeine Gewichtsänderung kann durch die Ableitung der Fehlerfunktion bestimmt werden. Dafür wird der Nabla-Operator auf die Fehlerfunktion angewendet:

$$\Delta W = -\eta \bar{\nabla} E(\Omega). \quad (18)$$

Der Vorfaktor η ist der Lernfaktor, der es ermöglicht, eine Schrittweite einzustellen. Je größer der Wert ist, desto stärker ist die Änderung der Gewichte, was jedoch dazu führen kann, dass ein Minimum übersprungen wird. Bei einem zu kleinen Faktor würde es kaum Änderungen beim Lernen geben und das Lernen würde mehr Zeit benötigen, sowie ein gefundenes lokales Minimum nicht wieder verlassen werden kann. Der negative Gradient der Fehlerfunktion gibt daher proportional an, wie stark die Gewichtsänderung der Gewichtsmatrix ΔW ist. Für ein einzelnes Element gilt daher:

$$\Delta \omega_{ij} = -\eta \frac{\partial}{\partial \omega_{ij}} E(\Omega). \quad (19)$$

Der Fehler von Neuron j ist der quadratische Abstand zwischen der Lerneingabe t_{pj} und der wirklichen Ausgabe o_{pj} . Durch die Verwendung des quadratischen Abstands wird später das Differenzieren erleichtert und Ausreißer stärker berücksichtigt. Für den Fehler des Musters p ergibt sich

$$E_p = \frac{1}{2} \sum_j (t_{pj} - o_{pj})^2. \quad (20)$$

Sowohl der quadratische Abstand, als auch der Faktor $\frac{1}{2}$ dient einer späteren Vereinfachung. Der Algorithmus verwendet die Minimierung des Fehlers, um die Parameter einzustellen. Ob der Algorithmus nun den halben Fehler oder den vollen Fehler minimiert, spielt dabei keine Rolle. Durch das Quadrieren werden größere Fehler deutlich stärker gewichtet und erhalten damit stärkere Berücksichtigung im Lernprozess. Das Summieren aller Einzelfehler der jeweiligen Muster p erzeugt den Gesamtfehler

$$E = \sum_p E_p. \quad (21)$$

Für die weitere Herleitung des Backpropagation-Verfahrens soll hier zuerst die Deltaregel zum Lernen von Netzen ohne verdeckte Schicht (einstufige Netze) eingeführt werden. Das hier verwendete Netz besitzt dabei eine lineare Aktivierungsfunktion. Aus den oben gegebenen Formeln ergibt sich:

$$\Delta\omega_{ij} = -\eta \frac{\partial}{\partial\omega_{ij}} E(\Omega) = \sum_p -\eta \frac{\partial}{\partial\omega_{ij}} E_p. \quad (22)$$

Durch die Anwendung mit der Kettenregel folgt

$$\frac{\partial E_p}{\partial\omega_{ij}} = \frac{\partial E_p}{\partial o_{pj}} \cdot \frac{\partial o_{pj}}{\partial\omega_{ij}}. \quad (23)$$

Durch das Einsetzen des Fehlers eines Musters (Formel (20)) kann die äußere Ableitung gebildet zu

$$\frac{\partial E_p}{\partial\omega_{ij}} = -(t_{pj} - o_{pj}) = -\delta_{pj}. \quad (24)$$

werden und für die innere Ableitung folgt

$$\frac{\partial o_{pj}}{\partial\omega_{ij}} = \frac{\partial}{\partial\omega_{ij}} \sum_i o_{pi} \omega_{ij}. \quad (25)$$

Da lediglich nach ω_{ij} differenziert wird, werden alle anderen Terme in denen die Variable nicht vorkommt, in der Ableitung zu null. Dadurch ergibt sich die Vereinfachung:

$$\frac{\partial o_{pj}}{\partial\omega_{ij}} = \frac{\partial}{\partial\omega_{ij}} \sum_i o_{pi} \cdot \omega_{ij} = o_{pi}. \quad (26)$$

Die gesamte Ableitung folgt dann durch das Zusammensetzen der einzelnen Terme zu

$$\frac{\partial E_p}{\partial\omega_{ij}} = -o_{pi} \cdot \delta_{pj}. \quad (27)$$

Damit lässt sich aus Formel (19) die Modifikationsregel aufstellen:

$$\Delta\omega_{ij} = \eta \sum_p o_{pi} \cdot \delta_{pj}. \quad (28)$$

Bei diesem Verfahren handelt es sich um ein Offline-Training. Das bedeutet, dass hier ein kompletter Batch (also eine Anzahl von Lernmustern) erst komplett vom System verarbeitet wird, bevor die Gewichte angepasst werden können. Wie bereits erwähnt, ist die oben beschriebene Delta-Regel nur für Netze geeignet, die lediglich aus einer Eingabe- und einer Ausgabeschicht bestehen und keine verdeckten Schichten besitzen. Das liegt daran, dass von jedem Neuron die Eingabe und die Ausgabe bekannt sein müssen. Für mehrstufige Netzwerke bietet das Backpropagation-Verfahren eine Möglichkeit des Trainings.

Wird ein Muster p als Netzeingabe net_{pj} für ein Neuron j verwendet, kann dies mit

$$net_{pj} = \sum_i o_{pi} \cdot \omega_{ij} \quad (29)$$

beschrieben werden. Aus der Ausgangsfunktion folgt dann mit der Netzeingabe net_{pj} das Ausgangssignal:

$$o_j = f_{act}(net_{pj}). \quad (30)$$

Erneut kann eine Gewichtsänderung aus den Fehlern berechnet werden:

$$\Delta\omega_{ij} = \sum_p -\eta \cdot \frac{\partial E_p}{\partial \omega_{ij}}. \quad (31)$$

Über die Kettenregel kann wieder die Ableitung bestimmt werden:

$$\frac{\partial E_p}{\partial \omega_{ij}} = \frac{\partial E_p}{\partial net_{pj}} \cdot \frac{\partial net_{pj}}{\partial \omega_{ij}}. \quad (32)$$

Die innere Ableitung sieht dabei wie folgt aus:

$$\frac{\partial net_{pj}}{\partial \omega_{ij}} = \frac{\partial}{\partial \omega_{ij}} \sum_i o_{pi} \cdot \omega_{ij} = o_{pi}. \quad (33)$$

Aus der äußeren Ableitung der Kettenregel ergibt sich das Fehlersignal:

$$\frac{\partial E_p}{\partial net_{pj}} = -\delta_{pj}. \quad (34)$$

Wird das Fehlersignal jetzt in Formel (32) eingesetzt, folgt:

$$\frac{\partial E_p}{\partial \omega_{ij}} = -\delta_{pj} \cdot o_{pi}. \quad (35)$$

Daraus folgt aus (31) durch Einsetzen die Gewichtsänderungsformel:

$$\Delta\omega_{ij} = \eta \sum_p \delta_{pj} \cdot o_{pi}. \quad (36)$$

Der Ausdruck ist identisch mit dem aus der Deltaregel, jedoch ist die Definition des Terms für δ_{pj} komplizierter:

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} = -\frac{\partial E_p}{\partial o_{pj}} \cdot \frac{\partial o_{pj}}{\partial net_{pj}} \quad (37)$$

mit

$$\frac{\partial o_{pj}}{\partial net_{pj}} = \frac{\partial}{\partial net_{pj}} f_{act}(net_{pj}) = f'_{act}(net_{pj}). \quad (38)$$

Der Faktor, der den Ausgang nach der Eingabe differenziert, ergibt logischerweise die Ableitung des Aktivierungssignals:

$$\delta_{pj} = -\frac{\partial E_p}{\partial o_{pj}} \cdot f'_{act}(net_{pj}). \quad (39)$$

Es ergeben sich zwei Fälle für den ersten Faktor $\frac{\partial E_p}{\partial o_{pj}}$, die im Folgenden betrachtet werden sollen. Im ersten Fall, wenn der Index j ein Neuron der Ausgabeschicht wäre, liefert

$$E_p = \frac{1}{2} \sum_j (t_{pj} - o_{pj})^2 \quad (40)$$

den Fehler für ein eingegebenes Muster. Dann gilt durch erneutes Anwenden der Kettenregel und unter der Annahme, dass es sich nur um das eine Neuron j handelt:

$$\frac{\partial E_p}{\partial o_{pj}} = - \frac{\partial \left(\frac{1}{2} \sum_j (t_{pj} - o_{pj})^2 \right)}{\partial o_{pj}} = -(t_{pj} - o_{pj}) \cdot (-1) = t_{pj} - o_{pj}. \quad (41)$$

Für das Ausgabeneuron folgt damit für δ_{pj} die Funktion:

$$\delta_{pj} = (t_{pj} - o_{pj}) \cdot f'_{act}(net_{pj}). \quad (42)$$

Der zweite Fall ist die Berechnung für ein Neuron der verdeckten Schicht:

$$\delta_{pj} = - \frac{\partial E_p}{\partial o_{pj}} \cdot f'_{act}(net_{pj}). \quad (43)$$

Die Ableitung kann hier nur indirekt bestimmt werden. Unter Zuhilfenahme der Gleichungen (29) und (34) folgt aus:

$$\delta_{pj} = - \sum_k \left(\frac{\partial E_p}{\partial t_{pk}} \cdot \frac{\partial t_{pk}}{\partial o_{pj}} \right) \cdot f'_{act}(net_{pj}). \quad (44)$$

Mit der Gleichsetzung der Netz-Eingabe und der Lern-Eingabe

$$t_{pj} = net_{pj} = \sum_i o_{pi} \cdot \omega_{ij} \quad (45)$$

sowie der Verwendung von

$$\frac{\partial E_p}{\partial net_{pk}} = -\delta_{pk} \quad (46)$$

ergibt sich:

$$\begin{aligned} \delta_{pj} &= - \sum_k \left(\delta_{pk} \cdot \frac{\partial}{\partial o_{pj}} \cdot \sum_i o_{pi} \cdot \omega_{ij} \right) \cdot f'_{act}(net_{pj}) \\ &= f'_{act}(net_{pj}) \cdot \sum_k \delta_{pk} \cdot \omega_{jk}. \end{aligned} \quad (47)$$

Daraus ergibt sich die folgende Fallunterscheidung:

$$\delta_{pj} = \begin{cases} f'_{act}(net_{pj}) \cdot (t_{pj} - o_{pj}) & \text{wenn } j \text{ Ausgabeneuron ist} \\ f'_{act}(net_{pj}) \cdot \sum_k \delta_{pk} \cdot \omega_{jk} & \text{wenn } j \text{ verdecktes Neuron ist.} \end{cases} \quad (48)$$

Wenn zum Beispiel die logistische Funktion als Aktivierungsfunktion verwendet wird, folgt aus Formel (34)

$$\delta_{pj} = \begin{cases} o_{pj} \cdot (1 - o_{pj}) \cdot (t_{pj} - o_{pj}) & \text{wenn } j \text{ Ausgabeneuron ist} \\ o_{pj} \cdot (1 - o_{pj}) \cdot \sum_k \delta_{pk} \cdot \omega_{jk} & \text{wenn } j \text{ verdecktes Neuron ist.} \end{cases} \quad (49)$$

Aus diesem Zusammenhang kann folgender Algorithmus approximiert werden:

```

1: PROCEDURE BACKPROPAGATION
2:    $n\_Zyklen := 0$ 
3:   REPEAT
4:      $fehler := 0$ 
5:      $n\_Zyklen := n\_Zyklen + 1$ 
6:     FOR  $i := 1 \dots AnzahlMuster$  DO
7:       Muster  $i$  anlegen
8:       FOR  $j := 1 \dots AnzahlOutputNeuron$  DO
9:         Bestimme Ausgabewert:  $o_j$ 
10:        Bestimme Fehlerwert:  $f_j := t_j - o_j$ 
11:        Bestimme Fehlersignal:  $fsig_j := o_j \cdot (t_j - o_j) \cdot f_j$ 
12:         $fehler := fehler + f_j^2$ 
13:      END FOR
14:      FOR  $s := Ausgabeschicht - 1$  DOWNTO  $ErsteSchicht$  DO
15:        FOR  $k := 1 \dots AnzahlNeuronenSchicht_s$  DO
16:           $fsum := 0$ 
17:          FOR  $m := 1 \dots AnzahlNeuronenSchicht_{(s+1)}$  DO
18:             $fsum := fsum + \omega_{km} \cdot fsig_{(s+1)m}$ 
19:          END FOR
20:           $fsig_{sk} := o_{sk} \cdot (1 - o_{sk}) \cdot fsum$ 
21:          FOR  $m := 1 \dots AnzahlNeuronenSchicht_{(s+1)}$  DO
22:             $\omega_{km} := \omega_{km} + \eta \cdot o_{sk} \cdot fsig_{(s+1)m}$ 
23:          END FOR
24:        END FOR
25:      END FOR
26:    END FOR
27:  UNTIL  $fehler < tolerierbarerFehler$  OR  $Zyklenzahl$  erreicht
28: END BACKPROPAGATION

```

Code 1: Pseudocode für den Backpropagation Lernalgorithmus
inspiriert aus: [Lämmel & Cleve, 2008]

Für die Berechnung des Fehlersignals eines Neurons werden alle Fehlersignale der darauffolgenden Schichten und die Verbindungsgewichte verwendet [Lämmel & Cleve, 2008]. In dem Code ist zu sehen, dass die Zeilen 8 bis 13 zuerst die Ausgabeschicht berechnen. Anschließend werden in den folgenden Zeilen die verdeckten Schichten verarbeitet, bis in Zeile 22 die Gewichte angepasst werden.

Die beschriebenen Gleichungen verwenden lediglich einen Punkt der Fehlerfunktion, daher ist nicht bekannt, wie die Fehlerfläche aussieht. Das kann zu verschiedenen Problemen führen. Einige Fehler sind in der Abbildung 14 dargestellt.

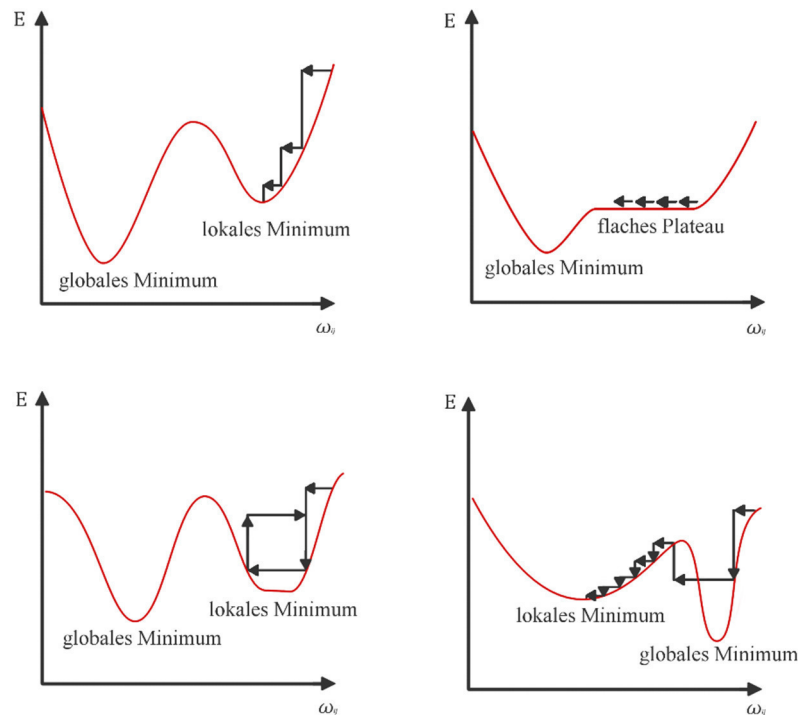
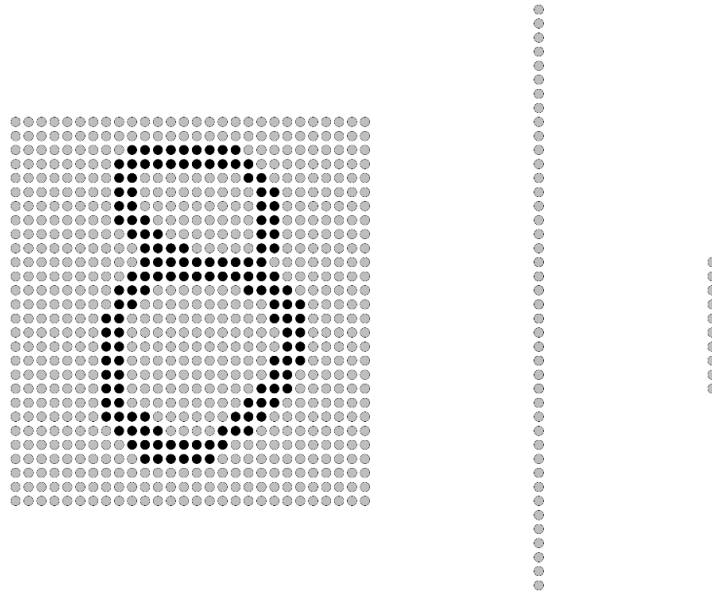


Abbildung 14: Fehlerfälle, die beim Gradientenabstiegsverfahren auftreten können
inspiriert aus: [Zell, 1994]

Ein Algorithmus, der auf diesen Gleichungen beruht, könnte statt im globalen Minimum in einem lokalen Minimum enden. Der Algorithmus würde durch die verringerte Steigung in einem Minimum die Schrittweite proportional dazu verringern und das Minimum daher nicht mehr verlassen können. Ein ähnliches Problem wäre, wenn der Algorithmus auf einem flachen Plateau der Fehlerfunktion festhängen würde. Auf dem Plateau wäre die Steigung gering und damit auch die Ableitung, beziehungsweise die Gewichtsänderung. Durch eine zu große Steigung um ein lokales Minimum könnte vom absteigenden Graphenteil zu dem aufsteigenden Graphenstück gesprungen werden. Hier dreht sich jedoch die Steigung, denn wenn diese ebenfalls zu stark ist, würde die Gewichtsänderung einen erneuten Sprung zurück zum gegenüberliegenden Steigungsstück verursachen. Dies würde zu einem Oszillieren führen. Es könnte auch das globale Minimum verlassen werden, wenn die Steigung einen zu großen Sprung verursachen würde und damit das Minimum übersprungen wird [Zell, 1994].

2.2.4 Bilder als Eingabeinformation im MLP

Die Schwellenwertelemente des vorangegangenen Kapitels verwenden als Eingabeinformation einzelne Parameter. Diese Parameter werden anschließend zur Klassifikation in dem Netz verwendet. Im nachfolgenden Fall soll jedoch ein Bild als direkte Eingabe dienen. Eine Beispielanwendung wäre die OCR (**O**ptical **C**haracter **R**ecognition), also Schriftzeichenerkennung. Ein Netzwerk welches diese Strukturen erkennen kann, hat folgenden Charakter:



*Abbildung 15: Aufbau eines MLP zur OCR Erkennung
inspiriert aus: [Lämmel & Cleve, 2008]*

In die Eingabeneuronen gelangen direkt die Grauwerte der einzelnen Pixel. In diesem Beispiel ist das Bild eines Zeichens 28x28 Pixel groß und es sollen Zahlen von 0 bis 9 erkannt werden. In der Abbildung 15 sind die Neuronen wieder als Kreise dargestellt. Die Eingabeschicht besteht damit aus 748 Neuronen (in der Abbildung ist eine Beispieleingabe der handschriftlichen Zahl Acht schwarz dargestellt). Diese Neuronen leiten die Daten in die Netzstruktur weiter. Abgesehen von einer möglichen Skalierung der 8 Bitwerte zwischen 0 und 1, erfolgt hier keine Verarbeitung. Die Verdeckte Schicht besteht hier aus 42 Neuronen. Jedes dieser Neuronen ist mit jedem Eingangsneuron über ein spezifisches Gewicht verbunden. Die Verbindungen und die Gewichte wurden im Bild aus Darstellungsgründen weggelassen. Jedes Neuron der Verarbeitungsschicht prüft wie stark das Eingabebild einer in dem Neuron abgespeicherten definierten Struktur ähnelt. Diese Struktur muss nicht zwangsweise direkt ein Schriftzeichen sein oder das gesamte Bild beinhalten. Die Information über die Ähnlichkeit wird an die Ausgabeschicht weitergegeben. Für jede Klasse, also jedes Schriftzeichen, existiert hier ein Neuron, das es repräsentiert. In diesem Beispiel sind es zehn Neuronen, eines für jede Ziffer. Jedes dieser Neuronen fasst die einzelnen Ergebnisse der Verarbeitungsschicht zusammen und ermittelt daraus die Wahrscheinlichkeit, inwiefern es sich bei dem jeweiligen Schriftzeichen um seine zu repräsentierende Ziffer handelt. Auch hier ist jedes Neuron der verdeckten Schicht mit jedem Neuron der Ausgabeschicht verbunden. Aus Darstellungsgründen wurden hier ebenfalls die Verbindungslinien weggelassen. Die Ausgabe jedes Neurons ist dann ein Wert zwischen 0 und 1, der die jeweilige Klassenwahrscheinlichkeit widerspiegelt.

Das Training dieses Netzes kann mit dem im vorherigen Kapitel dargestellten Backpropagation-Algorithmus durchgeführt werden. Für die Schriftzeichenerkennung bittet die MNIST-Datenbank über 70.000 Schriftzeichensätze zum Entwickeln und Testen von Anwendungen. Zum Training können dann zum Beispiel 50.000 Zeichen verwendet werden und 20.000 zum Testen. Diese Strukturen schaffen es auf eine Genauigkeit von über 97% [Lämmel & Cleve, 2008], 2010 erreichte ein neuronales Netz mit

6 Schichten und 5700 Neuronen eine Genauigkeit von 99,65% [Ciresan, et al., 2010], [Baldominos, et al., 2019]. Noch bessere Ergebnisse können mit Convolutional Neural Network (CNN) erreicht werden, wie sie nachfolgend beschrieben sind [Baldominos, et al., 2019].

2.2.5 Convolutional Neural Network

Das Convolutional Neural Network (CNN) ist eine von Yann LeCun vorgestellte Netzwerkarchitektur, die für eine effizientere Datenverarbeitung verschiedene Schichten (Layer) verwendet. Die CNN-Architektur eignet sich besonders gut dafür, Muster in Bildern und im Audibereich zu erkennen. Eines der Hauptbestandteile ist die Verwendung von verschiedenen Filtern, die in einigen Schichten sitzen. Diese Filter werden von dem Netz auf seine spezielle Aufgabe hintrainiert und verwendet, um spezielle Features (Merkmale) der zu klassifizierenden Klassen zu suchen und zu finden. Daher werden auch die Filter als Features bezeichnet. Vereinfacht kann dies als ein Filter für Kanten, Ecken oder Flächen aufgefasst werden. In Wirklichkeit ist es viel abstrakter, da das Netz mit modernen Lernverfahren die einzelnen Features in den zu klassifizierenden Klassen selbst sucht. Die ersten Netze wurden jedoch mit vordefinierten festen Features ausgestattet. Die Features sind Bestandteil der sogenannten Convolution-Schicht (Faltungsschicht). Es existieren dabei auch oft mehrere Convolution-Schichten, die aufeinander aufbauen. So sucht die erste Schicht zum Beispiel direkt Merkmale im Eingangsbild und eine nachgeschaltete Schicht vom Feature greift die gefundenen Merkmale der vorangegangenen Schicht auf und kombiniert diese zu komplexeren Modellen für die Klassifikation. Vereinfacht könnte die erste Schicht Kanten, Ecken und Flächen im Eingangsbild suchen und die zweite Schicht setzt diese Features zu Nase, Augen und Mund zusammen. Neben der Convolution-Schicht existieren unter anderem noch die Pooling-Schicht und die Dense-Schicht. Die wichtigsten Schichten sollen im Folgenden genauer erklärt werden.

2.2.5.1 Die Convolution Schicht

Die Convolution-Schicht ist das Arbeitspferd des CNN. Es verwendet, wie bereits erwähnt, ein Feature, in dem die gelernten Informationen der Klassenmerkmale stecken. Das Feature selbst hat meistens einen abstrakten Charakter. Damit die gelernten Merkmale, die im Feature hinterlegt sind im Eingangsbild gefunden werden können, wird das Feature mit dem Eingangsbild gefaltet und jeweils mit einem Schwellwert verarbeitet. Das Ergebnis dieser Berechnung wird anschließend in der sogenannten Featuremap gespeichert. Der Ablauf ist in Abbildung 16 dargestellt (hier mit der didaktischen Zuhilfenahme eines Zwischenergebnisses).

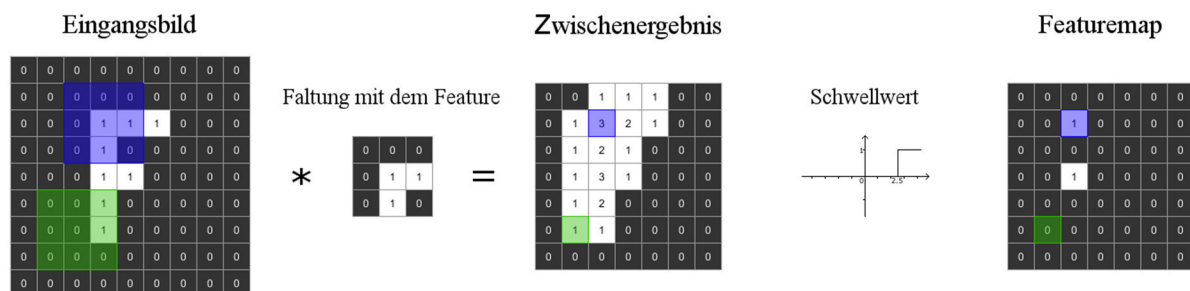


Abbildung 16: Darstellung der Arbeitsweise einer Convolution Schicht

Sofern ein Wert in einem Feld ungleich Null ist, ist das Feld weiß hinterlegt. Die Faltung funktioniert indem der mittlere Pixel des Features, der hier beispielhaft die Größe von 3x3 Pixel hat, schrittweise auf jeden Pixel des Eingangsbildes gelegt wird. Bei jedem dieser Schritte wird ein Produkt der jeweiligen übereinanderliegenden Pixel des Features und des Eingangsbildes berechnet. Anschließend wird die Summe über die in diesem Schritt berechneten Produkte gebildet (in der Abbildung zur Verdeutlichung als Zwischenergebnis bezeichnet). Der letzte Schritt besteht darin, die Summe mit einem Schwellwert

anzupassen. Das Ergebnis wird anschließend in der Featuremap gespeichert. Beispielhaft sind in der Abbildung zwei Fälle (Blau und Grün) hinterlegt. Würde das Feature auf die blaue Fläche gelegt werden, ergäbe sich folgender Zusammenhang:

$$(0 \cdot 0) + (0 \cdot 0) + (0 \cdot 0) + (0 \cdot 0) + (1 \cdot 1) + (1 \cdot 1) + (0 \cdot 0) + (1 \cdot 1) + (0 \cdot 0) = 3. \quad (50)$$

Dabei ist die linke Ziffer eines Produkts der jeweilige Wert des Eingangsbildes und die rechte Ziffer der Wert des Features. Das Ergebnis ist exemplarisch als Zwischenergebnis hinterlegt. Anschließend wird direkt über diese Summe der Schwellwert angewendet, dessen Ergebnis in die Featuremap geschrieben wird

$$\Phi(3) = 1 \text{ mit } \Phi(x) = \begin{cases} 1, & \text{wenn } x > 2,5 \\ 0, & \text{sonst} \end{cases}. \quad (51)$$

Analog dazu ergibt sich die Gleichung für den Fall, dass das Feature auf die grüne Fläche gelegt wird:

$$(0 \cdot 0) + (0 \cdot 0) + (0 \cdot 1) + (0 \cdot 0) + (0 \cdot 1) + (1 \cdot 1) + (0 \cdot 0) + (0 \cdot 1) + (0 \cdot 0) = 1 \quad (52)$$

und mit dem Schwellwert

$$\Phi(1) = 0 \text{ mit } \Phi(x) = \begin{cases} 1, & \text{wenn } x > 2,5 \\ 0, & \text{sonst} \end{cases}. \quad (53)$$

Jeder Pixel der Featuremap besteht in diesem Beispiel aus neun Produkten, die aufsummiert und mit einem Schwellwert verarbeitet sind. Anders als in diesem Beispiel bilden die Summierung und das Anwenden des Schwellwertes keine zwei getrennten Arbeitsschritte, sondern passieren direkt hintereinander. Wie in dem Beispiel zu sehen, ist die Featuremap kleiner als das Eingangsbild, was am Randbereich liegt. Wenn das Feature bei der Berechnung teilweise über die Grenzen des Eingangsbilds ragt, muss eine Methode existieren, wie mit den Pixeln umgegangen werden muss, die nicht mehr auf dem Eingangsbild liegen. Es gibt verschiedene Möglichkeiten, mit dem Randbereich umzugehen. Eine Möglichkeit besteht darin, das Eingangsbild künstlich mit Nullen zu vergrößern. Hier wurde die Faltung nur in dem Bereich verarbeitet, in welchem das Feature komplett im Eingangsbild liegt, wodurch die Featuremap minimiert wird. In diesem Beispiel wurde der Einfachheit halber ein einfacher Schwellwert als Funktion gewählt. In der Praxis wird jedoch häufig die Rectifier Linear Unit (ReLu) Funktion verwendet. Die Charakteristik der ReLu Funktion ist in der Abbildung 17 abgebildet.

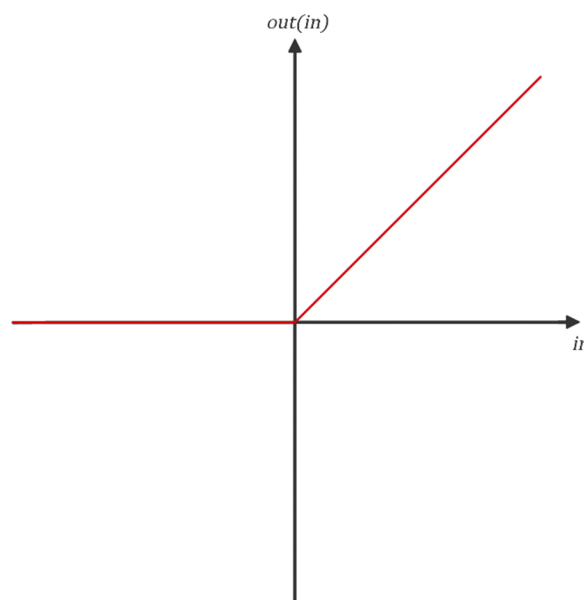


Abbildung 17: Charakteristik der Rectifier Linear Unit (ReLu) Funktion
inspiriert aus: [Wartala, 2018]

2.2.5.2 Die Pooling Schicht

Ein weiterer wichtiger Schritt im Convolutional Neural Network ist das Pooling. Dies ist ein nicht linear arbeitender Daten-Reduzierer und wird meistens direkt hinter einer Convolution-Schicht angewendet. Das Pooling verwendet eine Filtermaske, in der Regel von der Größe 2x2. Innerhalb von diesen vier Werten des Eingangsbildes wird nun überprüft, welcher Wert am größten ist. Der größte Wert wird behalten und die anderen Werte werden verworfen. Anschließend wird die Bearbeitungsmaske um die Pooling-Größe (hier 2) weiterverschoben, bis das ganze Datenbild abgearbeitet ist. Ein Beispiel des Max-Pooling-Verfahrens ist in Abbildung 18 dargestellt.

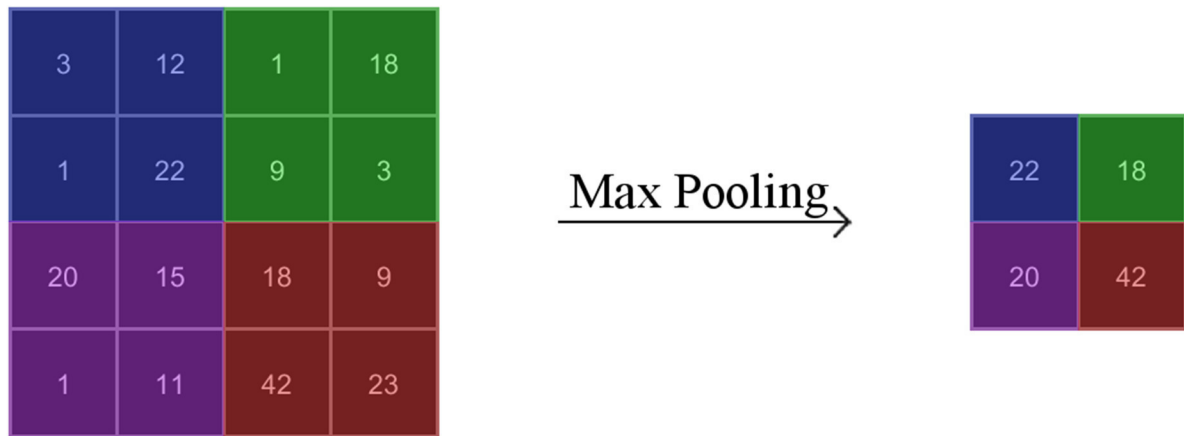


Abbildung 18: Darstellung der Arbeitsweise des Max Pooling

In der Abbildung besteht das Eingangsbild aus 4x4 Pixeln, die Pooling-Maske aus 2x2 Pixeln. Es wird oben links im Bild begonnen. Jede der vier Farben Blau, Grün, Lila und Rot stellen eine Pooling-Verarbeitung dar. Auf der rechten Seite der Abbildung ist das Ergebnisbild zu sehen. Bei einer Verwendung einer Pooling-Maske, von zum Beispiel 2x2 Pixeln, ist die Größe des Ausgangsbildes nur noch $\frac{1}{4}$ des Eingangsbildes. Das Pooling vollzieht durch diese Reduktion eine Mustererkennung. Es kommt auch häufiger vor, dass nebenstehende Ausgangswerte des Convolution-Layer ähnliche Werte aufweisen. Diese ähnlichen Werte erzeugen jedoch keinen direkten Informationsgewinn und können daher ignoriert werden. Das Pooling verringert dadurch die Komplexität des Systems und beugt einem Auswendiglernen (Overfitting) vor, womit tiefere Netze ermöglicht werden. Mit dieser Datenreduktion spart das Pooling ebenfalls Rechenzeit ein. Das Max-Pooling ist die am häufigsten verwendete Pooling-Art beim CNN. Es gibt zwar theoretisch auch Median-Pooling, Mittelwertbildung, Min-Pooling und weitere, jedoch zeigen diese nicht dieselbe Qualität des Max-Pooling [Wartala, 2018]. Pooling wird auch als Subsampling bezeichnet.

2.2.5.3 Die Dense-Schicht

Die Dense-Schicht ist im Prinzip wie ein MLP aufgebaut. Sie bildet meistens den Schluss eines CNN. Die Dense-Schicht ist eine vollverschaltete Schicht, das heißt, dass jede Information (zum Beispiel jeder Pixel) aus der Featuremap der vorangegangenen Schicht, mit jedem Neuron der Dense-Schicht verbunden ist. Die Dense-Schicht fasst am Ende des Netzwerkes die gesammelten Daten der Convolution-Schichten abschließend zusammen und teilt diese in Klassen ein. Die letzte Dense-Schicht im Netzwerk ist das Ausgangsnetz. Hier befindet sich zu jeder zu klassifizierender Klasse ein Neuron. Jedes dieser Neuronen gibt dann für die jeweilige Klasse die Wahrscheinlichkeit der Wiedererkennung an. Daher werden die realen berechneten Ausgangswerte mit einer Softmax-Funktion meistens normiert. Die Softmax-Funktion sorgt dafür, dass die einzelnen Ausgangswerte x_i aufsummiert den Wert 1 haben. Sie hat folgenden Charakter

$$\text{SoftMax}(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}}. \quad (54)$$

2.2.5.4 Die Drop-out-Schicht

Die Drop-out-Schicht wird nicht direkt im CNN für die Klassifizierung verwendet, jedoch steht sie beim Training des Netzwerkes hilfreich zur Seite. Das Problem besteht darin, dass Neuronale Netze unter Umständen zu einem Overfitting, also einem Auswendiglernen neigen. Das bedeutet, dass das Netzwerk keine eigenen Strukturen in den Lerndaten approximiert, sondern einfach die Testfälle auswendig lernt. Damit dies verhindert wird, sorgt die Drop-out-Schicht dafür, dass mit jedem Lernschritt die Ausgänge zufällig ausgewählter Neuronen nicht beachtet werden.

2.2.5.5 Der Aufbau eines Convolutional Neural Networks

Eines der ersten Convolutional Neural Networks ist das LeNet1 [LeCun, et al., 1995], [Wartala, 2018]. Dieses Netzwerk wurde für die Schriftzeichenerkennung von Zahlen verwendet. Es verwendet ein Eingangsbild von 32x32 Pixel und besteht aus zwei Paaren von Convolution- und Pooling-Schichten, gefolgt von zwei Dense-Schichten und der Ausgangsschicht. Eine Möglichkeit der Darstellung ist in Abbildung 19 veranschaulicht.

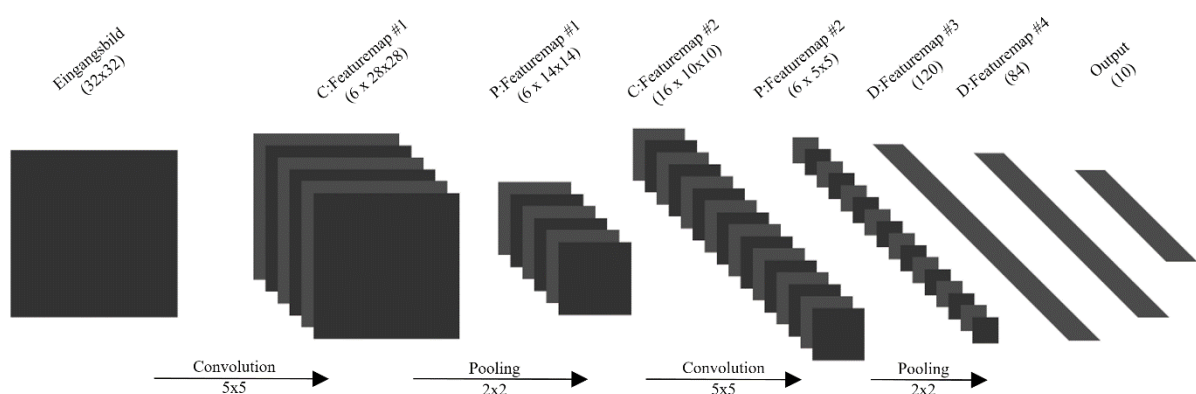


Abbildung 19: Aufbau des LeNet5 zur Klassifikation von Schriftzeichen [LeCun, et al., 1998]

In der Abbildung sind die Featuremaps als Kästen dargestellt. Über den Kästen ist der Name der Featuremap angegeben. Außerdem steht unter den Namen die Anzahl (Dimension) und die Pixelgröße der jeweiligen Schicht. Die Pfeile unten zwischen den Featuremaps geben die jeweilige Operation mit

der Operator-Größe an. Das hier abgebildete LeNet 5 setzt sich aus Schichten¹ zusammen. Das Eingangsbild besitzt eine Größe von 32x32 Pixeln und beinhaltet hier eine handgeschriebene Ziffer. Auf das Eingangsbild wirken 6 Features mit der Größe von 5x5 Pixeln. Jedes der 6 Feature-Ergebnisse wird in einer eigenen der 6 Featuremaps gespeichert. Anschließend folgt ein Pooling. Nach dem Pooling wird erneut eine Convolution mit der Größe von 5x5 durchgeführt und ein weiteres Pooling. Auf das Pooling-Ergebnis folgen zwei Dense-Schichten und die Ausgangsschicht. Am Ende des Netzes stehen 10 Neuronen, welche die jeweilige Klassenwahrscheinlichkeit angeben. Die Klassen wären in diesem Beispiel die Ziffern 0 bis 9. Das Netzwerk erreichte 1998 nach dem Training eine Genauigkeit von über 98% [LeCun, et al., 1998], [Wartala, 2018]. Neuere Veröffentlichungen mit CNNs erreichen Genauigkeiten von 99,76% auf dem Datensatz [Baldominos, et al., 2019]. Eine Klassifikation kann mit einem CNN als auch mit einem MLP umgesetzt werden. Ein CNN bietet in vielen Aufgaben jedoch bessere Ergebnisse, da es die räumliche Beziehung von Informationen (bei Bildern die benachbarten Pixel) für ihre Auswertung mitverwendet.

2.3 Statistische Methoden zur Untersuchung von Klassifikatoren

Für die Evaluation eines Klassifikators stehen verschiedene Maße zur Verfügung. Eine wichtige Grundlage für die Auswertung dieser Statistik ist dabei die Konfusions-Matrix [Japkowicz & Shah, 2012]. Diese Matrix besteht aus vier Werten und ist in Abbildung 20 dargestellt. Die Statistik betrachtet eine Klasse des Klassifikators, so dass der Entscheidungsprozess des Klassifikators als binär aufgefasst werden kann. Entweder sagt das Ergebnis der Klassifikation aus, dass es genau zu dieser Klasse gehört oder nicht. Der Wert True-Positiv gibt dabei die Anzahl der Klassifikationen an, die als diese Klasse richtig klassifiziert worden sind. True-Negativ sind die Anzahl der Klassifikationen, die tatsächlich nicht zu dieser Klasse gehören. Die Werte False-Positiv und False-Negativ geben an, ob die Klassifikation ein Element zur falschen Klasse zugeordnet hat.

Konfusions-Matrix		Ergebnis des Klassifikators	
		Positiv	Negativ
Gehört es wirklich zu der Klasse?	Positiv	True-Positiv (TP)	False-Negativ (FN)
	Negativ	False-Positiv (FP)	True-Negativ (TN)

Abbildung 20: Aufbau der Konfusions-Matrix inspiriert aus: [Japkowicz & Shah, 2012]

Anhand dieser vier Werte können anschließend die statistischen Größen, die in dieser Arbeit verwendet werden sollen, berechnet werden: Genauigkeit, Precision, Recall und der F-Score. Die mathematische Beschreibung ist in [Japkowicz & Shah, 2012] nachzulesen. Es sollen hier nur die wesentlichen Gleichungen dargestellt werden. Die Genauigkeit (Accuracy) gibt ein Verhältnis davon an, wie viele Klassifikationen insgesamt richtig waren zu allen vorgenommenen Klassifikationen. Der Wert kann mit der nachfolgenden Formel berechnet werden [Japkowicz & Shah, 2012]:

$$\text{Genauigkeit} := \frac{TP + TN}{TP + TN + FP + FN} \quad (55)$$

¹ Die Zählweise der Schichten ist in der Literatur nicht eindeutig. Einige Quellen zählen das Eingangsbild oder die Ausgangsschicht nicht mit. Hier sollen zur Vereinfachung alle Schichten gezählt werden.

Bei der Auswertung muss darauf geachtet werden, dass die Klassenverteilung gleich ist. Wenn zum Beispiel Klasse A 95% der Klassen stellt und Klasse B 5% und der Klassifikator für jede Klassifikation nur Klasse A angibt, würde der Klassifikator hier eine Genauigkeit von 95% erreichen.

Ein weiteres Maß ist die Precision [Japkowicz & Shah, 2012]

$$Precision := \frac{TP}{TP + FP}. \quad (56)$$

Die Precision ist ein Verhältnis dafür, wie viele Klassen in Bezug auf die Gesamtzahl der Ergebnisse in dieser Klasse wirklich richtig sind. Wohingegen der Recall-Wert [Japkowicz & Shah, 2012], Formel (57) ein Verhältnis dafür ist, wie viele Klassifikationsergebnisse wirklich richtig sind zu der Gesamtzahl des Vorkommens dieser Klasse in der verwendeten Menge. Beide Werte sollten im Idealfall 1 betragen. Der Recall wird auch als True-Positiv-Rate (TPR) bezeichnet.

$$Recall := \frac{TP}{TP + FN}. \quad (57)$$

Das gewichtete harmonische Mittel aus der Precision und dem Recall ergibt den F1-Score, wie in der nachfolgenden Formel [Japkowicz & Shah, 2012]:

$$F1 := \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \quad (58)$$

3. Sensorauswahl und Vorverarbeitung

3.1 Einleitung

In diesem Kapitel soll mit der Beschreibung des Grundkonzeptes begonnen werden, in welcher der Jäteroboter in verschiedene Komponenten aufgeteilt wird, um diese anschließend unabhängig voneinander veranschaulichen zu können. Der erste Schritt zur Pflanzenklassifikation ist es, ein passendes Trägersystem zu finden und dieses mit Sensoren auszustatten, die die benötigten Informationen aufnehmen, so dass anschließend ein Klassifikator eine Unterscheidung der Informationen berechnen kann. Eine Herausforderung in diesem Projekt liegt besonders in der rauen Umgebung, da sich der komplette Aufbau entweder den Umwelteinflüssen auf dem Feld anpassen muss oder von eben diesen abgeschirmt werden soll. In diesem Kapitel soll das Trägersystem kurz vorgestellt werden. Anschließend werden die verschiedenen untersuchten Sensoren und ihre Ergebnisse präsentiert. Anhand eines Fazits zum Kapitel Sensoren wird eine Sensorart für dieses Projekt ausgewählt und entsprechend diskutiert. Weiter soll in dem Kapitel die Vorverarbeitung hergeleitet werden, die für die Klassifikation benötigt wird. Es werden verschiedene Verfahren zur Segmentierung der Pflanzen vorgestellt und evaluiert. Auch sollen in einem speziellen Fazit die Schlussfolgerungen und die Auswahl eines Segmentierungsalgorithmus erläutert werden. Ein abschließendes, umfassendes Fazit steht schließlich am Ende des Kapitels.

3.2 Grundkonzept

Der Jäteroboter lässt sich in zwei Bestandteile aufteilen. Das Trägersystem, das autonom über das Feld fährt, und die daran montierte Jäteeinheit. Die Jäteeinheit ist die Einheit, die das eigentliche intelligente Jäten des Roboters übernimmt. Das Konzept dieser Einheit setzt sich aus drei wesentlichen Modulen zusammen. Das erste Modul ist die Sensoreneinheit. Diese muss die Informationen zu den Pflanzen aufnehmen und bereitstellen. Der anschließend arbeitende Klassifikator nimmt die gesammelten Informationen und teilt die darin gefundenen Pflanzen in die definierten Klassen Nutzpflanze (hier Karotte) und Beikraut (umgangssprachlich Unkraut) ein. Anschließend werden die Klassifikationsergebnisse an den dritten Bestandteil, eine Vernichtungseinheit, übergeben, um das unerwünschte Beikraut zu entfernen. Eine schematische Darstellung ist zur Verdeutlichung in Abbildung 21 dargestellt.

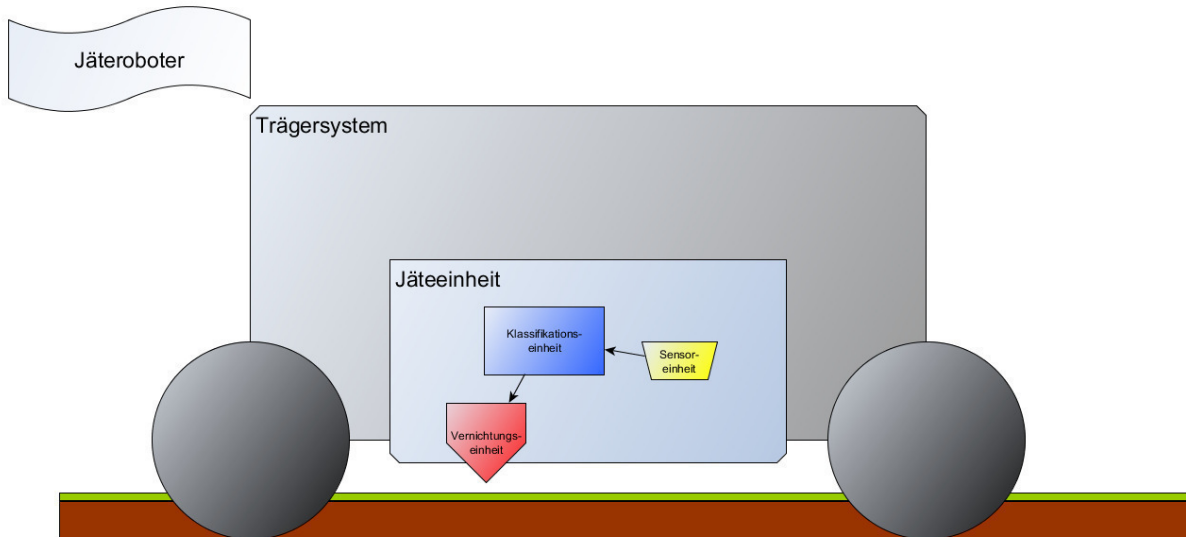


Abbildung 21: Skizze der Aufteilung der Bestandteile des Jäteroboters

In Abbildung 21 stellen die schwarzen Pfeile in der Jäteeinheit den Informationsfluss von der Sensoreinheit über die Klassifikatoreinheit zu der Vernichtungseinheit dar. Durch die endliche Informationsverarbeitungsgeschwindigkeit und das konstante Fahren des Jäteroboters sind die Daten, die die Sensoren aufnehmen und das, was die Vernichtungseinheit bearbeiten muss, nicht identisch. Damit die Vernichtungseinheit die Position der zu entfernenden Pflanze kennt, muss diese Translationsbewegung berechnet werden. Durch die Verwendung einer Kamera, die während der Translationsbewegung zwei überlappende Bilder aufnimmt, kann über den Kameratakt und den im Bild entstehenden Bewegungsvektor die Translation berechnet werden. Dazu müssen in jedem Bild mehrere verschiedene Punkte markiert werden, die in jedem der beiden überlappenden Bilder vorkommen. Mit Hilfe eines Korrespondenzalgorithmus kann ein Verschiebungsvektor berechnet werden, mit dem die Translation vorhergesagt werden kann, wie dies in [Knoll, et al., 2015 a], [Knoll, et al., 2015 b] und [Knoll, et al., 2016 a] beschrieben ist.

3.3 Die Vernichtungseinheit

Die Vernichtungseinheit stellt keinen wesentlichen Bestandteil dieser Arbeit dar, dennoch sollen vollständigkeithalber die verschiedenen Ansätze vorgestellt werden. Ein besonders wichtiger zu berücksichtigender Aspekt ist die kurze Zeit von weniger als einer Sekunde, die der Vernichtungseinheit pro Pflanze bleibt. Daher muss ein Aufbau eine entsprechende Geschwindigkeit besitzen und hat keine große Zeit, um auf die Pflanze einzuwirken. Weiter muss bedacht werden, dass der Aufbau auch in der rauen Umgebung auf dem Feld über eine längere Zeit eingesetzt werden muss.

Als Vernichtungseinheit könnte hier ein Stempelsystem arbeiten. Ein solches System drückt die unerwünschte Pflanze ein Stück in den Erdboden. Die Pflanze wird damit nicht direkt vernichtet, jedoch wird sie in Ihrer Entwicklung soweit behindert, dass die Nutzpflanze in der Nachbarschaft den Nährstoffwettkampf auf dem Feld gewinnt. Der Ansatz sieht vor, dass nicht ein einzelner Stempel verwendet wird, sondern ein Array von nebeneinander liegenden Stempeln. Durch die Bewegung des Roboters wäre damit immer ein Stempel in Position und ein aufwendiges Anfahren der Pflanze durch einen Stempel würde entfallen. Das Konzept wurde in [Michaels, et al., 2015] dargestellt. Eine weitere Alternative wäre das mechanische Abfräsen der Pflanze, womit diese wirklich entfernt würde. Auch ein Laser-System würde in Zukunft denkbar, sobald das Energieproblem gelöst wird [Kaierle, et al., 2013]. Damit eine thermische Vernichtung durchgeführt werden kann, muss die Pflanze auf mindestens 60 Grad erhitzt werden. Bei dieser Temperatur werden die Eiweißmoleküle in der Pflanze angegriffen, so

dass die Pflanze daraufhin abstirbt². Ein Laser, der genügend Energie in eine Pflanze einbringen kann, damit diese in wenigen Millisekunden beseitigt wird, konnte jedoch nicht gefunden werden. Ein Problem stellt hier auch die raue Umgebung auf dem Feld dar, welche mit Staub und Erschütterungen auf den Laser einwirken würde. Eine weitere Überlegung wäre ein MASER (*Microwave Amplification by Stimulated Emission of Radiation*) oder die Verwendung eines Magnetron mit einer Mikrowellenlinse zur Fokussierung der Pflanze, um mit Mikrowellen die Pflanzen zu beseitigen. Dies soll in Zukunft noch getestet werden. Der Vorteil bei der Verwendung eines Lasers oder von Mikrowellen wäre die Minimierung von mechanischen Bauteilen und die damit verbundene Verringerung des mechanischen Verschleißes. Zurzeit stellt eine mechanische Vernichtungseinheit jedoch die praktikablere Lösung dar. Auch eine Indizierung eines elektrischen Stroms in die Pflanze könnte einen weiteren Ansatz darstellen³. Als Trägersystem innerhalb des Roboters könnte ein Deltaroboter dienen, der durch seine Achsen die Unebenheiten des Feldes bei der Fahrt ausgleicht⁴. Der Westhof nutzt zur allgemeinen flächendeckenden Vernichtung das sogenannte Abflammen. Beim Abflammen wird mittels einer aus Gas erzeugten Flamme das gesamte Feld von Pflanzen befreit. Auch dies könnte sich durch ein gezieltes Einsetzen der Flamme weiterentwickeln lassen [Knoll & Czymbek, 2018].

Welches Konzept sich schließlich aus seiner theoretischen Überlegung heraus zu einer realistisch anwendbaren Lösung mit hoher Effizienz weiterentwickeln lässt, muss noch untersucht werden.

3.4 Das Trägersystem der Sensoren

3.4.1 BoniRob

Für die Reproduzierbarkeit der Ergebnisse ist es wichtig, ein System zu haben, welches eine gleichbleibende Arbeitsumgebung bereitstellt. In diesem Fall stellt das Trägersystem ein Roboter der Firma Deepfeld Robotics dar. Der vom Hersteller gegebene Name des Roboters ist BoniRob, welcher sich aus den Wörtern Bonitur und Roboter zusammensetzt. Das System wurde für die autonome Feldarbeit entwickelt, besitzt aber selbst keinerlei Bearbeitungs-Mechanismen und stellt einen experimentellen Prototyp dar. Es ist eine reine Trägerplattform für raue Umgebungen. BoniRob ist unter anderem mit einem Laserscanner ausgestattet, mit dessen Hilfe das vor ihm liegende Feld gescannt wird. In den aufgezeichneten Daten werden die typischen Karottenwälle gesucht und anhand der Wälle kann das System autonom über das Feld navigieren. Am Ende einer Reihe kann der Roboter selbständig die Reihe verlassen, sich drehen und die nächste Reihe abfahren. Die Geschwindigkeit kann dabei zwischen 0,1 m/s und 1,5 m/s eingestellt werden.

² nicht veröffentlichte Aussage von Biologen und Landwirten

³ der Autor hatte die Idee mittels Teslaspule eine Hochspannung in die Pflanze zu induzieren, die Idee wurde in der Abschlussarbeit von [Niang, 2015] umgesetzt und untersucht.

⁴ die Idee der Verwendung eines selbstgebauten Deltaroboters wurde ebenfalls in einer Abschlussarbeit [Meißner, 2018] untersucht und in [Hussmann, et al., 2019] veröffentlicht.



Abbildung 22: *BoniRob zwischen vier Karottenreihen*

Wie in Abbildung 22 zu erkennen ist, besitzt der Roboter vier an beweglichen Armen befestigte Räder. Die vier Arme können elektrisch innerhalb von 90° bewegt werden. Damit können verschiedene Spurbreiten den jeweiligen Bedürfnissen auf dem Feld angepasst werden. Für eine autarke Stromversorgung befindet sich auf der Rückseite ein Benzingenerator. Der Roboter besitzt die Form eines U, dessen Öffnung sich an der Seite befindet. In dieser Öffnung können die jeweiligen individuellen Aufbauten befestigt werden [Albert, 2015].

3.4.2 Sensorhalterung

Die entworfene Sensorhalterung wird in der Öffnung des BoniRobs befestigt. Der Aufbau soll es ermöglichen, mit verschiedenen Sensoren aus verschiedenen Winkeln das Feld aufzuzeichnen. Der Roboter soll bei seiner späteren Arbeit unabhängig von Umwelteinflüssen wie Tageszeit oder Wetter arbeiten können. Daher wird der Aufbau mittels Abschirmung von der Umgebung getrennt. Die Abschirmung ist der hölzerne Kasten, der in der Abbildung 22 unter dem BoniRob zu sehen ist. Weiter wird mit Hilfe eines Beleuchtungskranzes, wie in Abbildung 23 zu erkennen ist, ein künstliches Licht auf den Arbeitspunkt gebracht. Als Beleuchtung wurden im ersten Schritt 16 Halogenlampen mit einer Farbtemperatur von 8600 K so angeordnet, dass diese gemeinsam Brennpunkte auf dem Wall ausleuchten. Es wurde sich für die Halogenlampen entschieden, da hier auch eine Ausleuchtung im infraroten Spektrum möglich ist und dies für einige Sensoren wichtig ist. Mit dieser Anordnung kann die Schattierung, die durch die Beleuchtung entsteht, minimiert werden.

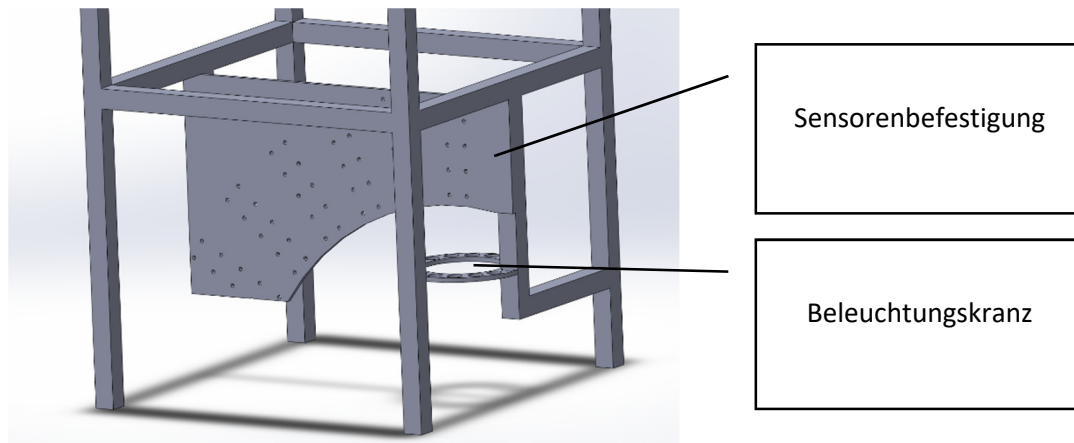


Abbildung 23: *Sensoraufhängung [Knoll, et al., 2016 b]*

Die gefertigte Sensorbefestigung weist verschiedene Bohrlöcher auf. Diese sind jeweils doppelt für eine einfache Arretierung. Mit den vordefinierten Löchern können die Sensoren zwischen 0° (Vogelperspektive auf die Pflanzen) und 90° komplett von der Seite auf die Pflanzen reproduzierbar angebracht werden. Außerdem können die Sensoren in Schritten von 5 cm zwischen 60 cm und 80 cm vom Fokus des Lichtkegels entfernt befestigt werden. Die Sensoren wurden extern getriggert, so dass mit einer Verfahrensgeschwindigkeit von 0,2 m/s eine Bildüberlappung von circa 20% zwischen zwei aufgenommenen Bildern entsteht. Mit diesem Aufbau wurden die im folgenden Kapitel beschriebenen Sensoren auf dem Feld getestet [Knoll, et al., 2016 b], [Holtorf, et al., 2016].

3.5 Die untersuchten Sensoren und Konfigurationen

Im Folgenden werden die Sensoren beschrieben, die in der ersten Phase des Projektes untersucht wurden. Die Ergebnisse wurden unter anderem in [Knoll, et al., 2016 b] veröffentlicht. Die Sensoren sind mittels des beschriebenen Trägersystems auf den Feldern des Kooperationspartners der Westhof Bio GmbH vom 22. Juli 2015 bis zum 14. August 2015 auf täglich über 1000 Meter Karottenwällen getestet worden. Die gesammelten Daten wurden anschließend verwendet, um eine Aussage über die folgenden Punkte zu treffen:

- Welcher Sensor bietet die besten Eigenschaften, um die Informationen bereitzustellen, damit der Klassifikator eine Klassifikation zwischen den Pflanzen vornehmen kann?
- Welche Konfiguration des jeweiligen Sensors ist am besten geeignet, um die Daten aufzunehmen?
- Welches Wachstumsstadium der Pflanzen bietet die nötigen Informationen, um die Klassen zu unterscheiden, ist aber nicht zu groß, um eine anschließende Bearbeitung zu behindern?

Einige Sensoren wurden in späteren Saisons weiter getestet.

3.5.1 Bispektral-Kamera JAI

Eine Bispektral-Kamera ist eine Kamera, die zwei verschiedene Sensorchips verwendet. Ein Sensor zeichnet dabei ein Bild im herkömmlichen RGB-Spektrum auf, während ein weiterer Sensor ein Infrarot-, beziehungsweise nahes Infrarot-Spektrum verwendet. Beide Sensoren werden dabei physikalisch so angeordnet, dass sie dasselbe Objektiv verwenden. Damit sind beide Bilder (also alle vier Kanäle) deckungsgleich. Das bedeutet, dass jeder Pixel aller vier Kanäle dasselbe Objekt aus demselben Winkel zeigt. Damit ist eine komplizierte Koordinatentransformation zwischen den Sensoren nicht nötig. Siehe dazu Abbildung 24, in dem das RGB-Bild und NIR-Bild abgebildet ist.

Ein bekannter Hersteller für eine Bispektral-Kamera ist die Firma Jai, die die gleichnamige Kamera JAI herstellen [JAI, 2015]. Die Kamera wurde bereits in anderen Agrarprojekten verwendet [Haug, et al., 2014]. Die Kamera besitzt eine Auflösung von 1296 x 966 Pixel.

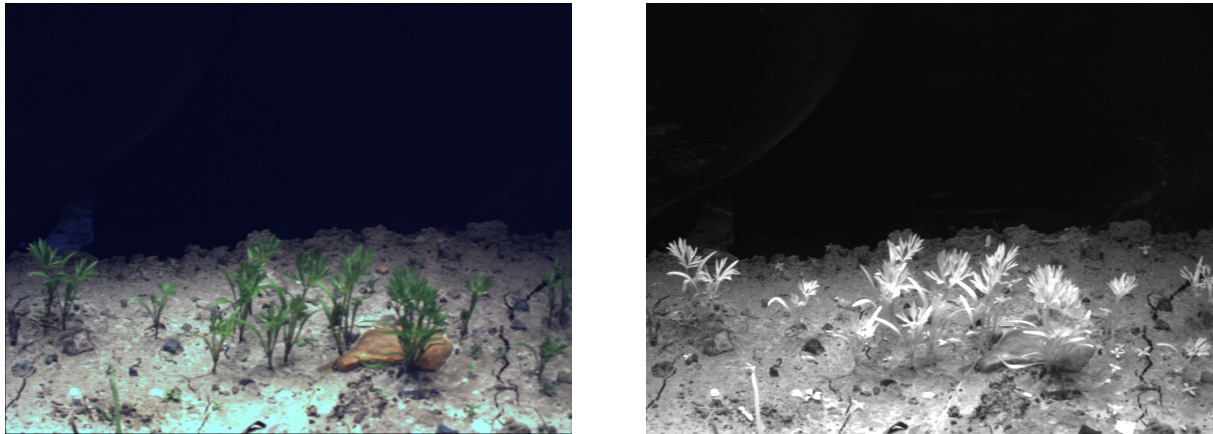


Abbildung 24: Links RGB Bild und Rechts IR-Bild der JAI Kamera

Wie in der Abbildung 24 zu erkennen ist, ist das Nahe IR-Bild als Graustufenbild dargestellt. Das organische Pflanzenmaterial weist durch seinen hohen Gehalt an Chlorophyll eine hohe Reflexionseigenschaft auf, da es im Bereich des infraroten Spektrums reflektiert. Dadurch heben sich die Pflanzen im Bild hervor.

3.5.2 RGB Kamera

Neben der JAI-Kamera, die bereits ein RGB-Bild bereitstellt, sollen zusätzlich herkömmliche RGB-Kameras getestet werden, die eine höhere Auflösung haben und für die Farben eine höhere Bittiefe bereitstellen. Im ersten Jahr wurde eine Kamera von Baumer, sowie die Nikon 5500D getestet. Die Nikon-Kamera, die eine Spiegelreflex-Kamera ist, wurde verwendet, da die Saison bereits fortgeschritten war und die Baumer-Kamera nicht zuverlässig lief. Die Nikon-Kamera bietet eine maximale Auflösung von 6000 x 4000 Pixel [Nikon, 2015]. Eine Beispielaufnahme ist in Abbildung 25 dargestellt. Im darauffolgenden Jahr wurde die Raytrix R42-Kamera von der Firma Raytrix verwendet. Diese Kamera bietet mit seinen 42 Megapixel eine Auflösung von 7716 x 5364 Pixel [Raytrix, 2016]. Im dritten Jahr wurde eine Kamera der Firma Basler (die daA1920-30uc) verwendet, mit einer Auflösung von 1920×1080 Pixel [Basler, 2017].



Abbildung 25: Aufnahme der Nikon 5500 D

3.5.3 Lichtfeld Kamera

Die Firma Raytrix bietet die R42-Kamera auch als Lichtfeld-Kamera an [Raytrix, 2016]. Eine Lichtfeld-Kamera verwendet Mikrolinsen, um zusätzlich ein 3D-Bild zu erzeugen. Die Software, die aus dem Mikrolinsenbild anschließend ein 3D-Bild erzeugt, war jedoch extrem rechenintensiv. Daher wurde diese Technik lediglich im Labor getestet, aber für den Feldeinsatz wegen ihrer Geschwindigkeit als nicht geeignet weiter verfolgt.

3.5.4 Cam Cube III

Eine weitere 3D Kamera ist die Cam Cube III⁵. Diese Kamera verwendet die Time of Flight (TOF) Technologie um mittels Phasenverschiebung ein 3D-Bild zu erzeugen. Dafür sendet die Kamera mit einer integrierten Beleuchtung 4 Belichtungsimpulse im infraroten Spektrum aus. Die vier Impulse sind dabei um je 90° phasenverschoben. Die 4 aufgenommenen Bilder, die die Kamera aus den 4 Belichtungsimpulsen aufnimmt, können zur Berechnung der Phasenverschiebung für jeden Pixel separat verwendet werden. Anhand der Wellenlänge kann damit eine Entfernung berechnet werden. Außerdem können unter anderem aus den 4 Bildern ein Grauwertbild und ein Amplitudenbild erzeugt werden. Der Vorteil der Kamera ist ihre Geschwindigkeit. Anders als bei einer Stereo-Vision-Kamera, können hier mittels relativ einfacher Berechnungen die Entfernungen für jeden Pixel berechnet werden [Husmann & Knoll, 2013], [Husmann, et al., 2014]

Die Cam Cube III hat eine Auflösung von 204 x 204 Pixel und ermöglicht bei einer Modulationsfrequenz von 20 MHz eine Genauigkeit von kleiner als 2 mm.

Eine Überlegung wäre, dass durch einen Umbau der Beleuchtung neben dem Entfernungsbild auch ein RGB-Bild und Infrarot-Bild erzeugt wird, sodass jedes der 4 Bilder in einem anderen Spektrum belichtet wird, also zum Beispiel im roten, grünen, blauen und infraroten Farbbereich. Der Vorteil wäre, dass hier das 3D-Bild, das RGB-Bild und das IR-Bild deckungsgleich wären, da alle Bilder über denselben Chip aufgenommen werden.

⁵ Es existiert leider kein öffentlich zugängliches Datenblatt der Cam Cube III, daher soll das Datenblatt der Cam Cube II [PMD, 2015], das ähnlich ist, hier als weiterführende Information angegeben werden.

Beim Test auf dem Feld zeigte sich jedoch, dass die Kamera nicht mit der Translation des Objektes zurechtkam. Dies liegt an der versetzten Zeit der Aufnahmen. So hat sich das Motiv zwischen den einzelnen 4 Bildern während der Aufnahme leicht bewegt. Daher lässt sich eine Phasenverschiebung für einen Pixel nicht eindeutig bestimmen und es entstehen nicht definierte Fragmente im 3D Bild. Das Problem der TOF-Kamera bei Translation ist in [Hussmann, et al., 2011] und in [Hussmann & Hermanski, 2013] genauer beschrieben.

3.5.5 Argos 3D P100

Eine weitere TOF-Kamera, die Argos 3D P100 [Bluetechnix, 2015], kommt von der Firma Bluetechnix. Diese Kamera bietet eine Auflösung von 160 x 120 Pixel mit 160 FPS. Der Öffnungswinkel beträgt dabei 90° und die Messreichweite reicht von circa 0,1 bis 3 m. Die integrierte Beleuchtung arbeitet bei einer Wellenlänge von 850 nm. Die Genauigkeitsangabe liegt dabei bei +/- 2 mm bei einer Entfernung von 0,7 m. Diese Kamera wies ähnliche Probleme wie die Cam Cube III auf und wurde daher nicht weiter untersucht.

3.5.6 Kinect II

Die Kinect II⁶ verwendet zwei Sensoren, unter anderem einen TOF-Sensor mit einer Auflösung von 512 x 424 Pixeln. Der TOF-Sensor erzeugt neben dem 3D-Bild zusätzlich ein infrarotes Bild. Der andere Sensor liefert ein RGB-Bild mit einer Auflösung von 1920 x 1080 Pixeln. Die mitgelieferte Software bietet die Möglichkeit, alle drei Bilder (3D, RGB, IR) aufeinander zu projizieren. Die Kinect II wurde von Microsoft entwickelt und überwiegend für die X-Box vertrieben. Eine 3D-Aufnahme eines Feldausschnitts ist in Abbildung 26 dargestellt. Die Kinect wurde dabei außerhalb der Lichtabschottung befestigt. So konnte eine Störung der anderen Sensoren durch die Kinect II Beleuchtung verhindert werden.

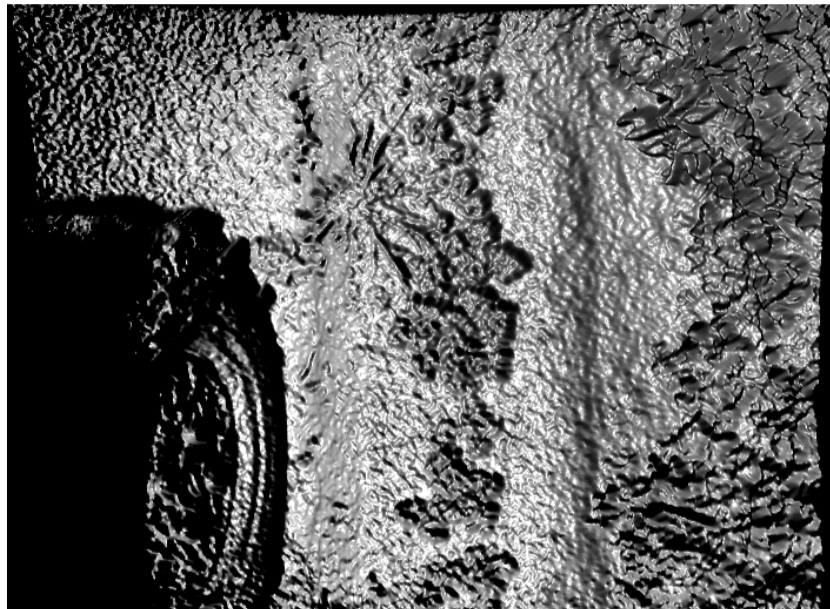


Abbildung 26: *TOF 3D Aufnahme der Kinect II*

In der Abbildung 26 ist links der Reifen des BoniRobs zu erkennen. In der Mitte des Bildes ist der Wall auf dem die Karotten gezüchtet werden erkennbar. Es lässt sich zwar auf dem Wall eine Struktur von Objekten erkennen, jedoch kann nicht eindeutig interpretiert werden, ob es sich bei der Struktur um eine

⁶ Nachdem Microsoft die Produktion der Kinect II eingestellt hat, verschwanden scheinbar auch alle offiziellen zugänglichen Datenblätter und Produktbeschreibungen. Vereinzelt technische Details können jedoch noch auf der Website von Microsoft und seinen Partnern an verschiedenen Stellen gefunden werden.

Pflanze oder eine Erdunebenheit handelt. Weiter lässt sich in dem Bild nicht feststellen, zu welcher Klasse eine gegebene Pflanze gehört. Auch das RGB-HD-Bild kann für diese Information nicht verwendet werden. Die Belichtungszeit der RGB-Kamera bekommt kein scharfes Bild bei der Translation des Roboters hin.

Das Problem der Kinect II liegt in ihrem Aufgabenbereich. Microsoft hat den Sensor für eine Spielekonsole entworfen. Der Sensor dient dabei als Interface und soll die Spielumgebung (in der Regel das Wohnzimmer) komplett aufnehmen. Daher besitzt die Kinect II ein festverbautes Objektiv, das für diesen Anwendungsbereich geeignet ist. Die Kamera arbeitet auf dem Feld jedoch im Nahbereich und ist mit 80 cm in der Nähe ihres minimalen Arbeitsbereiches. Wie auf der Abbildung 26 zu erkennen ist, wird mit dem verbaute Objekt über 2/3 der Auflösung verschwendet.

Daher wurde die Kinect mit einem neuen Objektiv ausgestattet [Knoll, et al., 2017], damit eine höhere Auflösung im Arbeitsbereich erreicht wird. Diese Prototypkonstruktion konnte zwar im Labor getestet werden, sie versagte jedoch in der rauen Umgebung auf dem Feld. Durch die fortgeschrittene Saison konnte aus zeitlichen Gründen kein neuer Umbau mehr erfolgen.

3.5.7 LMI Gocator 2350

Der LMI Gocator 2350 ist ein Laserscanner [LMI, 2015]. Dieser tastet die Objekte mit einem Laserstrich ab. Dadurch, dass bekannt ist, wie dieser Strich auf einer flachen Oberfläche in einem bestimmten Abstand aussieht, kann beim Eintreten eines Objektes durch die Verschiebung eine Entfernung berechnet werden. Damit kann eine Genauigkeit zwischen 0,019 bis 0,060 mm in der Entfernung erzielt werden. Die Abtastung erfolgt dann in 0,150 bis 0,300 mm Schritten in der Verfahrrichtung. Das Field of View ist dabei zwischen 158 und 365 mm.

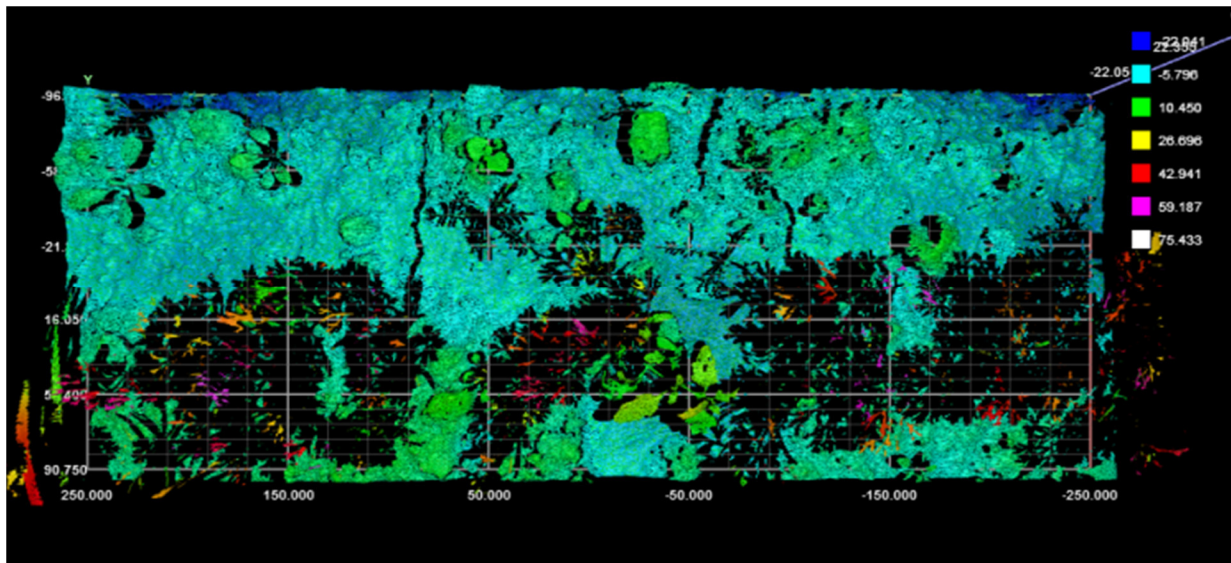


Abbildung 27: *Abgetastetes Bild des Laserscanners*

In der Abbildung 27 ist ein aufgenommenes Bild zu sehen. Die Entfernung zum Sensor ist farblich dargestellt. Flächen, die in blau dargestellt sind, weisen eine weite Entfernung zum Sensor auf. Flächen die grün bis rot dargestellt sind, liegen dicht am Sensor. Im Bild können in der oberen Hälfte einzelne Beikräuter erkannt werden. In der unteren Hälfte befinden sich neben Beikräutern auch Karotten. Durch die verzweigte Blattstruktur der Karotten, schatten diese den Sensor zu sehr ab. Auch besitzen einige Pflanzen für die Wellenlänge einen unzureichenden Reflexionswert, der durch die Winkel der Blätter verstärkt wird, sodass eine flächendeckende Angabe eines Entfernungswertes an diesen Stellen nicht möglich ist.

Der Sensor liefert zwar bessere Werte als eine TOF-Kamera, jedoch sind der Preis und die Problematik mit der lückenhaften Abtastung ein Problem. Auch müssen die einzelnen Scanlinien während der Fahrt erst zu einem Bild zusammengesetzt werden.

3.5.8 Fazit zu den Sensoren

Die dargestellten Sensoren unterscheiden sich hinsichtlich ihrer physikalischen Messmethoden. Der wesentliche Unterschied ist, dass einige Sensoren ein Bild in einem oder mehreren Spektralbereichen aufnehmen, andere hingegen Tiefenbilder erstellen. Die später verwendete Messmethode und der daraus resultierende Sensor müssen ausreichend Informationen bereitstellen, damit eine Klassifikation ermöglicht wird. Eine wichtige Betrachtung bei der Sensorauswahl ist es, nicht nur zu entscheiden, ob ein Klassifikator, also eine KI, mit den Informationen des Sensors eine Entscheidung treffen kann, sondern ob diese Informationen auch dem Menschen für die Erstellung der Lernmasken ausreicht. Eine KI könnte aus den gegebenen Informationen Zusammenhänge erkennen, die der Mensch zwar nicht erkennt, jedoch wäre das Training dieser KI ein anderer Ansatz als in dieser Arbeit. Dieser Aspekt hat in der Zukunft eventuell keine entscheidende Auswirkung mehr, was in dem Moment der Fall wäre, wenn eine KI soweit ist, dass sie sich selber beibringt, was sie zu klassifizieren hat. Im Kapitel 3.6.1 *Ground Truth Erstellung* wurde ein solcher Ansatz des Autolernens verfolgt. In der jetzigen Phase des Projektes ist es jedoch sinnvoller, dass die Lernmasken direkt aus den jeweiligen Sensordaten erstellt werden können. In dem Moment, in dem der Mensch eine Klassifikation vornehmen kann, ist außerdem sichergestellt, dass genügend Informationen für diese Arbeit bereitstehen, um eine Auswertung und Prüfung des Systems zu gewährleisten. Weiter muss die Umgebung, in der der Sensor arbeitet, bedacht werden. Der Sensor muss aus einem Abstand von circa 0,7m auch bei Translation des Walls zuverlässige Informationen liefern. Hinzu kommt die raue Umgebung auf dem Feld.

Die TOF-Sensoren haben teilweise Probleme mit der Translation und besitzen eine zu geringe Auflösung, damit ein Mensch die Daten für die Lernmaskenerstellung verwenden kann. Der Laser-Scanner LMI Gocator 2350 besitzt zwar eine höhere Auflösung, jedoch gibt es hier Probleme mit der lückenlosen Abbildung. Die Lichtfeldkamera berechnet zwar ein detailliertes 3D-Bild, jedoch ist die Rechenkapazität, die für die Echtzeitberechnung nötig ist, nicht praktikabel auf dem Feld umsetzbar.

Die Bisppektralkamera und die reine RGB-Kamera liefern Informationen, aus denen der Mensch ohne Probleme Lernmasken erstellen kann. Daher wurde sich für eine herkömmliche RGB-Kamera entschieden, da ein NIR-Bild zwar weitere Informationen bereitstellt, jedoch diese Informationen für eine Klassifikation überflüssig sind, wie im Kapitel 4 zu sehen sein wird. Außerdem besitzt eine herkömmliche RGB-Kamera eine höhere Auflösung und ist kostengünstiger.

Die Wachstumsphase der Pflanzen sollte für eine vernünftige Bearbeitung so klein sein, dass sie keine anderen Pflanzen überdecken. Zwar könnte der Blickwinkel so angeordnet werden, dass die Kamera unter die Pflanze blicken kann, jedoch werden dann mehrere Kameras benötigt, die trotzdem nicht alles sehen können. Daher soll der Blickwinkel der Kamera von oben auf den Wall gerichtet sein. Dieser Blickwinkel hat zudem den Vorteil, dass die Pflanzen nicht optisch hintereinander stehen.



Abbildung 28: RGB Aufnahme unter unterschiedlichen Wetterbedingungen und Wachstumsstadien (links junges Wachstumsstadium mit Regen, rechts spätes Wachstumsstadium mit Trockenheit)

Auch wenn die 3D-Sensoren nicht für die Klassifikation in diesem Projekt verwendet werden sollen, können diese für das Ausrichten der Vernichtungseinheit zum Karottenwall verwendet werden. Der Karottenwall besitzt über die gesamte Reihe keine homogene Höhe. Für die Vernichtungseinheit ist es jedoch wichtig, dass ein gewisser Abstand zum Wall nicht überschritten wird. Daher muss dieser nachgeregelt werden. Damit der Abstand zum Wall bestimmt werden kann, kann mittels der Distanzwerte eines 3D-Sensors und einer daraus berechneten Regressionsfläche die Höhe reguliert werden [Knoll, et al., 2015 a], [Knoll, et al., 2015 b] und [Knoll, et al., 2016 a].

3.6 Vorverarbeitung der Sensorwerte

Nachdem der Sensor die Daten aufgenommen hat, erfolgt eine Vorverarbeitung, bevor der Klassifikator die Daten zur Berechnung der Klassen verwendet. Die Vorverarbeitung hat die Aufgabe, überflüssige Informationen, schneller als der Klassifikator es kann, aus den Daten zu entfernen. Damit muss der Klassifikator weniger überflüssige Informationen verarbeiten und kann sich auf seine Aufgabe der Pflanzenklassifikation konzentrieren. Die Vorverarbeitung soll hier eine Segmentierung vornehmen. Die Segmentierung soll den Erdboden in den Bildern auf schwarz setzen und so den Hintergrund herausfiltern.

3.6.1 Ground Truth Erstellung für die Segmentierung

Zur Untersuchung der verschiedenen Segmentierungsalgorithmen wird ein Ground Truth benötigt. Diese Vergleichsmasken wurden von verschiedenen Personen anhand der Original-Bilder erstellt. Für die Untersuchung der Segmentierung wurden die RGB-Bilder der JAI-Kamera verwendet. Dies hat den Grund, dass lediglich von dieser Kamera auch NIR-Bilder vorhanden sind, mit denen eine Vergleichsmessung mit dem „State of the Art“ Vegetationsindexverfahren NDVI (siehe dazu Kapitel: 3.6.4 Vegetationsindexverfahren) möglich ist.

Es wurden insgesamt zufällig 293 Bilder von verschiedenen Wachstumsstadien und Wetterbedingungen zur Auswertung ausgewählt und manuell segmentiert. Die Bilder können in die Kategorie *Nass* (93 Bilder⁷), *Normal* (100 Bilder) und *Trocken* (100 Bilder) eingeteilt werden. In Abbildung 28 sind die verschiedenen Fälle abgebildet.

Jedes Bild besitzt eine Auflösung von 1296 x 966 Pixel (siehe dazu Kapitel: 3.5.1 JAI). Beim Speichern der Bilder ist jedoch ein schwarzer Rahmen um die Bilder entstanden, sodass der Informationsgehalt lediglich innerhalb einer Auflösung von 1292 x 962 Pixel enthalten ist. Das bedeutet, dass insgesamt, wenn jeder Pixel einzeln betrachtet wird, 364.170.872 Messpunkte zur Verfügung stehen. Diese Betrachtung ist möglich, wenn die zu untersuchenden Algorithmen lediglich die Informationen aus einem Pixel ziehen, sodass jede Messung voneinander unabhängig ist. Daher soll sich in den nachfolgenden Messungen nicht nur auf die einzelnen Bilder bezogen werden, sondern es wird jeder Pixel als separierter Messpunkt angesehen. Eine Betrachtung auf Basis der einzelnen Bilder würde die Frage aufwerfen, warum ein Bild 1296 x 966 Pixel besitzt. Es wäre dann möglich, die Bildmenge zum Beispiel zu verdoppeln, indem aus einem Bild zwei Bilder mit je 648 x 966 Pixel erzeugt werden. Die eigentliche Information (Messpunkte) wäre jedoch gleich geblieben.

Weiter beinhalten die Bilder unterschiedliche Verhältnisse von Hintergrundpixeln zu Pflanzenpixel. Es gehören in den Bildern (je nach Bild) durchschnittlich 93% (mit einer Standardabweichung von 0,2%) der Pixel zum Hintergrund. Daher wird für die Auswertung eine Skalierung verwendet, damit der Genauigkeitswert nicht verfälscht wird.

⁷ Ursprünglich existierten auch von der Messreihe „Nass“ 100 Bilder. Nach dem Erzeugen sämtlicher Segmentierungsergebnisse, stellte sich jedoch heraus, dass sieben NIR-Bilder fehlerhaft waren und somit für die Vergleichsmessung des NDVI Verfahrens nicht verwendet werden konnten. Zusätzlich stand keine Rechenkapazität mehr zur Verfügung um erneut sieben Bilder zeitnah auszuwerten.

Werden nur die Anzahl der Messpunkte der Pflanzen (Anzahl Pixel aller Bilder) betrachtet, ergibt sich für die drei Kategorien jeweils anhand der Stichprobentheorie eine statistische Sicherheit⁸ von über 99,6%.

Die Grenzen der Genauigkeit des Ground Truth liegen in seiner Erstellung, die wie bereits erwähnt, von verschiedenen Personen manuell durchgeführt wurde. Dabei wurde von jeder Person jedes Bild segmentiert, so dass von allen Bildern mehrere manuelle Segmentierungen vorliegen. Die Ground-Truth-Masken wurden anschließend mit einer „best of“-Auswertung erstellt. Durch die manuelle, individuelle Arbeit können Fehler nicht ausgeschlossen werden, die durch dieses Verfahren herausgemittelt werden sollen. Die Unterschiede der einzelnen manuellen Masken, die hier als Fehler bezeichnet werden können, liegen in der Interpretation der Bilder durch die Personen selbst. So können Fehler durch fehlende Absprachen entstehen, wie zum Beispiel der Umgang mit grünen Steinen, Blättern von Bäumen oder grünen Stöcken, die von einigen Personen als Pflanze gekennzeichnet werden und von anderen nicht. Auch individuelle Betrachtungen von zum Beispiel toten Pflanzen, die schon eine braune Farbe aufweisen, ergeben Unterschiede. Die hier am häufigsten vorkommenden Unterschiede liegen jedoch im Übergang von Pflanze zu Erdboden (Hintergrund). Dadurch dass die Bilder nicht unendlich scharf sind, existiert eine gewisse Unschärfe in dem Bereich, wie dies in der Abbildung 29 zu erkennen ist.



Abbildung 29: *Vergrößerte Darstellung des Randbereichs zwischen Pflanze und Erde*

Zur besseren Verdeutlichung wurde von allen manuell segmentierten Bildern aller Personen die vier Werte True-Positiv, True-Negativ, False-Positiv und False-Negativ unter Einfluss der individuellen Entfernung zur Kante aufsummiert. Anschließend konnte von jeder Entfernung eine Genauigkeit⁹ berechnet werden. Das Ergebnis ist in Abbildung 30 dargestellt. Die Genauigkeit bezieht sich dabei auf die erstellten Ground-Truth-Bilder.

⁸ Aus der Vorlesung: „Aquisitions- und Risikoanalyse“ von Prof. Dr. M. Defant [Defant, 2016].

⁹ Die manuell erstellten Masken wurden mit dem Ground Truth verglichen.

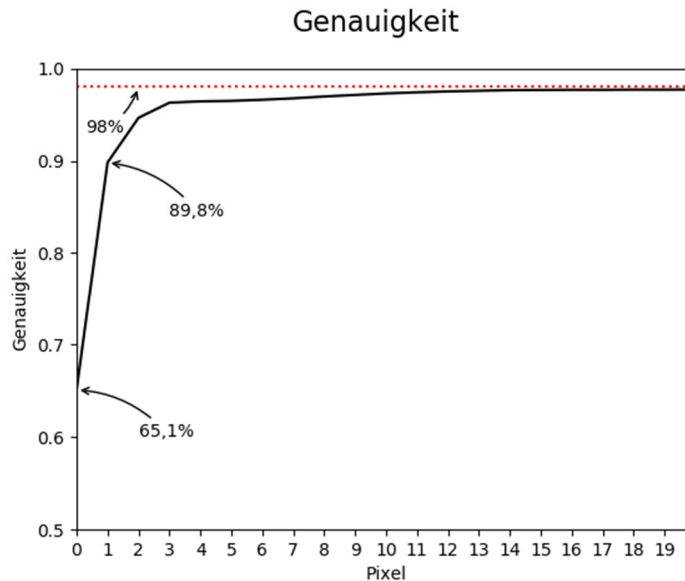


Abbildung 30: Genauigkeit der einzelnen Entfernungen der manuellen Segmentierungsmasken

Wie in der Grafik zu erkennen ist, nimmt die Genauigkeit mit zunehmendem Pixelabstand zu einer Kante (Abszisse) zu. Hier ist lediglich der Abstand bis zum 20. Pixel dargestellt, der Trend ist jedoch weiterführend. Auch ist die Anzahl der Ereignisse (Pixel) für die Genauigkeit je Abstand unterschiedlich. Obwohl wissenschaftliche Bedenken existieren, soll diese Grafik hier verwendet werden, um die Problematik zu erklären. Der Abstand eines Pflanzenpixels zu einem Erdpixel und eines Erdpixels zu einem Pflanzenpixel ist in der Auswertung gleichbedeutend. Interessant ist, dass die ersten beiden Pixel zu einer Kante eine Genauigkeit von 65,1% und von 89,8% aufweisen. Anschließend strebt die Kurve an 98% heran (dargestellt durch eine rote gestrichelte Gerade). Das bedeutet, dass in den ersten beiden Pixeln zu einer Kante große Unterschiede der Interpretation der manuellen Masken liegen. Daher soll eine spätere Auswertung der Algorithmen den Kantenabstand von den ersten beiden Pixeln ignorieren.

Eine weitere Idee war es, ein neuronales Netz mit einer MLP-Struktur zu verwenden, um die Segmentierung der Ground-Truth-Bilder zu erstellen. Dabei sollte das Netz nicht mit Lernmasken trainiert werden, sondern selber eine Unterscheidung vornehmen und sich auf dieses Problem optimieren. Die Vorgehensweise sah vor, dass für das Training auf dem Netz ein Bild gegeben wird, welches nun das Netz mit seinen zufallsinitialisierten Gewichten komplett klassifiziert und für jeden Pixel eine Wahrscheinlichkeit ausgibt. Dadurch, dass die Gewichte zufällig sind, liegt die Vermutung nahe, dass das Netz die Pixel in zwei Kategorien einteilt. Die Hoffnung war, dass in den Wahrscheinlichkeitswerten die Unterscheidung zwischen Pflanze und Erdboden enthalten sind. Nach der zufälligen Segmentierung sollten die Wahrscheinlichkeitswerte verwendet werden, um das Bild zu idealisieren. Das heißt, dass es sich ab einem bestimmten Wahrscheinlichkeitswert um eine Pflanze (gesetzt auf den Wert 1) handelt oder um Hintergrund (gesetzt auf den Wert 0). Dieses idealisierte Bild sollte anschließend zum Trainieren desselben neuronalen Netzes verwendet werden. Nach einigen Trainingsiterationen sollte erneut das Bild segmentiert, wieder idealisiert und zum Training benutzt werden, sodass das Bild immer besser durch das Netz segmentiert wird und auch so eine definierte Kante zwischen Pflanze und Hintergrund entsteht, die immer gleich segmentiert wird. Die Versuche in der Praxis mit diesem Autolernverfahren führten jedoch dazu, dass alles als Hintergrund oder Pflanze segmentiert wird. Um hier Abhilfe zu schaffen, wurde das zufällig initialisierte Netz vortrainiert. Im Training wurden jedoch absichtlich fehlerhafte Labels verwendet. Durch diese „Verwirrungsphase“ segmentierte das Netz mit seinen zufällig initialisierten Gewichten circa 50% der Pixel als Pflanze und 50% der Pixel als Hintergrund. Das Ergebnis war natürlich falsch, jedoch konnte dadurch eine gleiche Verteilung der Ergebnisse in dem Initialnetz erzeugt werden. Aber auch hier ergab eine weitere

Lernphase wie oben beschrieben, dass nach wenigen Iterationen das gesamte Bild als Hintergrund oder Pflanze segmentiert wird. Daher wurden vorerst Versuche mit dem Autolernen aus Zeitgründen eingestellt.

Ein Problem mit den Datensätzen ist, dass viele gleiche Fälle, auf die ein Segmentierer gut oder schlecht reagiert, die Ergebnisse zum Guten oder Schlechten verschieben könnten. Wenn zum Beispiel ein Algorithmus gut darin ist, den Erdboden zu erkennen, welcher weit weg von einer Pflanze ist, ist dieser Algorithmus im Vorteil, da dieser Fall häufiger in der Datenmenge vorkommt, als andere Fälle. Eine Möglichkeit wäre, jede gleiche Konstellation aus den Testfällen herauszufiltern.

Anders betrachtet gibt diese Zusammensetzung auch die Realität wieder, in der die Algorithmen arbeiten müssen. Daher wurden die Datensätze so belassen, dass sie möglichst gut die Realität widerspiegeln. Lediglich eine Skalierung wurde bei einigen Messungen verwendet.

3.6.2 Die verwendeten Auswertungen für die Segmentierung

Im folgenden Kapitel sollen für jeden Segmentierer verschiedene statistische Auswertungen erfolgen. Jede dieser Auswertungen betrachtet den Segmentierer auf eine andere Art. Dabei wird jede Messung auf den vier Datensätzen (Datensatz der alle Bilder beinhaltet, Datensatz der die nassen Bilder beinhaltet, Datensatz der normalen Bilder und der Datensatz der trockenen Bilder) durchgeführt. Die Untersuchung richtet dabei das Hauptaugenmerk auf die Entdeckung der Pflanzen und nicht auf die des Hintergrundes. Es ist wichtiger, alle Pflanzen zu erkennen, als Pflanzen zu übersehen. Einzelne Pflanzen weisen je nach Auflösung nur wenige Pixel auf, so dass ein Fehlen weniger Pixel eine Pflanze komplett aus dem Bild beseitigen könnte.

Die erste Auswertung betrachtet dabei keine einzelnen Bilder. Es werden die statistischen Größen (True-Positiv, True-Negativ, False-Positiv und False-Negativ) über alle Bilder summiert. Anschließend wird aus diesen Summen eine Skalierung vorgenommen, damit eine gleichmäßige Verteilung zwischen Pflanzenpixel und Hintergrundpixel vorliegt. Aus den skalierten Daten können dann die Genauigkeit, die Precision, der Recall und der F1-Score berechnet werden. Dabei sollen alle möglichen Parameter eines Algorithmus untersucht werden. Diese Messung soll hier als Analyse ohne Einzelbild-Betrachtung bezeichnet werden oder auch als globale Genauigkeitsanalyse, wie sie in Formel (59) dargestellt ist (analog: globale Precision, globaler Recall und globaler F1-Score):

$$\begin{aligned} \text{globale Genauigkeit} &= \frac{TP + TN \cdot s}{TP + TN \cdot s + FP \cdot s + FN} \\ s &= \frac{TP + FN}{TN + FP} \\ TP &= \sum_b^{\text{Bilder}} \text{TruePositiv}_b \\ TN &= \sum_b^{\text{Bilder}} \text{TrueNegativ}_b \\ FP &= \sum_b^{\text{Bilder}} \text{FalsePositiv}_b \\ FN &= \sum_b^{\text{Bilder}} \text{FalseNegativ}_b. \end{aligned} \tag{59}$$

Die zweite Auswertung wird benötigt, da vermutet wird, dass die berechneten Genauigkeiten, die über alle Bilder bestimmt worden sind, nicht auf den Einzelbildern realistisch erreicht werden. Daher soll mit Hilfe von Hypothesentests geprüft werden, ob die globalen Genauigkeiten P auch auf den Einzelbildern erreicht werden, bzw. ob es signifikante Abweichungen gibt. Dazu werden die Messgrößen True-Positiv, True-Negativ, False-Positiv und False-Negativ der einzelnen Bilder bestimmt und verwendet, um die zu erwartenden Schwankungen des Erwartungswerts zu vergleichen. Analysiert wird mit Hilfe eines linksseitigen Hypothesentests [Japkowicz & Shah, 2012] mit der folgenden Hypothese: Die Hypothese H_0 sagt aus, dass bei dem Einzelbild die Genauigkeit p größer ist als P .

$$H_0: p \geq P. \quad (60)$$

Die Anzahl des Stichprobenumfangs soll mit N gekennzeichnet werden und beinhaltet alle verwendeten Pixel des jeweiligen Bildes. Dabei wurde für die Parameter eine Skalierung zwischen Pflanzen,- und Hintergrundpixel verwendet. Die Skalierung wird über die Summe aller Pixel aller Bilder berechnet und auf die einzelnen Bilder angewendet.

$$N = TruePositiv_{Bild} + TrueNegativ_{Bild} + FalsePositiv_{Bild} + FalseNegativ_{Bild}. \quad (61)$$

Der Erwartungswert μ kann berechnet werden über

$$\mu = N \cdot P \quad (62)$$

und die untere Grenze a ergibt aus

$$a = \mu - r \cdot \sigma \quad (63)$$

$$\text{mit } \sigma = -\sqrt{N \cdot P \cdot (1 - P)}.$$

Der verwendete Radius r wird einmal für den Test auf Signifikanz (95%; $r = 1,64$) und auf Hochsignifikanz (99%; $r = 2,33$) eingesetzt. In dem Test wird auch nur die untere Grenze betrachtet, da die Genauigkeiten, die besser sind als der Erwartungswert und außerhalb dieser Grenze liegen, das Ergebnis nicht negativ beeinflussen.

Mit dem Hypothesentest soll einmal berechnet werden, wie viele Bilder nicht innerhalb der Grenzen für eine bestimmte Genauigkeit sind. Weiter soll berechnet werden, auf welchen Wert die Genauigkeit herabgesenkt werden muss, damit nur noch eine statistisch zu erwartende Anzahl von Bildern signifikant bzw. hoch signifikant abweicht.

Die dritte Messung berechnet die statistischen Größen (True-Positiv, True-Negativ, False-Positiv und False-Negativ) über die einzelnen Bilder. Auch die Skalierung wird separat für das jeweilige Bild berechnet und angewendet. Dabei werden jedoch die Parameter für den Segmentierer über den gesamten Datensatz unverändert gelassen. Anschließend wird für jedes Bild die Genauigkeit, die Precision, der Recall und der F1-Score berechnet. Die einzelnen Werte werden dann über alle Bilder gemittelt. Dabei soll erneut eine Auswertung der vier Datensätze erfolgen. Diese Messung soll Analyse mit Einzelbildbetrachtung bezeichnet werden oder auch lokale Genauigkeitsanalyse, wie in Formel (664) dargestellt (analog dazu lokale Precision, lokaler Recall und lokaler F1-Score).

$$\begin{aligned}
 \text{lokale Genauigkeit} &= \frac{1}{\text{Anzahl Bilder}} \cdot \sum_b^{\text{Bilder}} \frac{TP_b + TN_b \cdot s_b}{TP_b + TN_b \cdot s_b + FP_b \cdot s_b + FN_b} \\
 s_b &= \frac{TP_b + FN_b}{TN_b + FP_b} \\
 TP_b &= \text{TruePositiv vom Bild } b \\
 TN_b &= \text{TrueNegativ vom Bild } b \\
 FP_b &= \text{FalsePositiv vom Bild } b \\
 FN_b &= \text{FalseNegativ vom Bild } b
 \end{aligned} \tag{64}$$

Die vierte Messung wiederholt die Messungen eins (Analyse ohne Einzelbildbetrachtung) und die Messung drei (Analyse mit Einzelbildbetrachtung), jedoch ohne Skalierung. Damit soll kurz auf die realen Bedingungen eingegangen werden und die Ergebnisse der Realität gegenübergestellt werden.

Die fünfte und letzte Messung verwendet für jedes Bild einen optimalen Datensatz und eine separate Skalierung. Es soll hier simuliert werden, wie ein fiktiver Algorithmus vor der Verwendung des Segmentierers die optimalen Parameter bestimmt und anwendet. Die Genauigkeit, Precision, Recall und F1-Score werden dabei über alle Bilder des Datensatzes gemittelt.

Außerdem wird vollständigkeithalber (sofern dies möglich ist) die ROC-Kurve angeben, da diese in der Literatur häufig verwendet wird. Aus didaktischen Gründen ist an vielen Stellen jedoch eine andere Grafik sinnvoller, daher soll auf die ROC-Kurve im Einzelnen nicht weiter eingegangen werden. Zudem werden alle notwendigen Charakteristika der Algorithmen mit den anderen Graphen erläutert.

3.6.3 Einfacher globaler Schwellwert auf dem grünen Kanal

Zum Vergleich soll vorerst ein einfacher globaler Schwellwert auf dem grünen Kanal des RGB-Bildes gesetzt werden. Hier gibt es jeweils zwei Möglichkeiten. Entweder wird der normale Schwellwert verwendet, das heißt, wenn der Farbwert des grünen Kanals des jeweiligen Pixels oberhalb des Schwellwertes liegt, wird er in der Maske als Pflanze markiert, oder es handelt sich um den invertierten Schwellwert. Dabei wird der Pixel in der Maske als Pflanze gesetzt, wenn der Farbwert des grünen Kanals des jeweiligen Pixels unterhalb eines Schwellwertes liegt. Die in dem vorangegangenen Kapitel dargestellten Ground-Truth-Bilder wurden jeweils alle mit jedem der Schwellwerte [0...255] durchgerechnet. In der Abbildung 31 ist links der Verlauf der Genauigkeitsberechnung der einzelnen Schwellwerte abgebildet. Die Graphen verlaufen alle auf einer Glockenkurve. Dies liegt daran, dass niedrige Schwellwerte dafür sorgen, dass die Wahrscheinlichkeit steigt, sodass der jeweilige Wert über dem Schwellwert liegt und hohe Schwellwerte dafür sorgen, dass die Wahrscheinlichkeit erhöht wird, unterhalb des Schwellwertes zu liegen. Eine größtmögliche Unterscheidung liegt daher in der Mitte der beiden Extrema. Der Vollständigkeit halber ist rechts die ROC-Kurve angegeben.

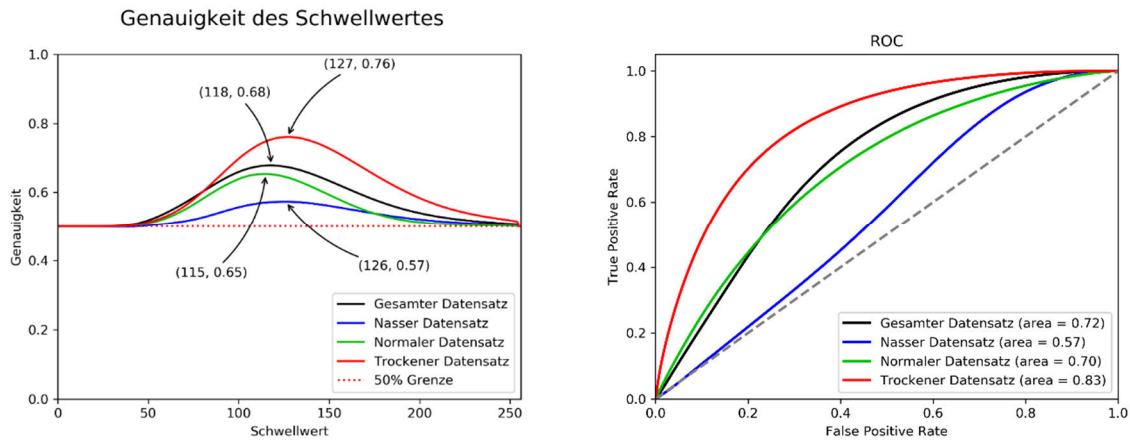


Abbildung 31: Genauigkeitsverlauf der einfachen Schwellwerte, ohne Betrachtung der einzelnen Bildelemente und rechts die ROC-Kurve

Bei der Berechnung wurden alle Bilder mit demselben invertierten Schwellwert untersucht. Dadurch, dass für jeden Schwellwert alle Bilder eines Datensatzes den gleichen Bedingungen unterlegen waren (einschließlich einer Skalierung Pflanzenpixel zu Hintergrundpixel), kann die Betrachtung der Auswertung über die Gesamtzahl aller Messpunkte (Pixel) über alle Bilder erfolgen. Daher wurden hier die Summen der statistischen Größen True-Positiv, True-Negativ, False-Positiv und False-Negativ über alle Bilder berechnet und anschließend die Genauigkeit, Precision, Recall und der F1-Score aus diesen Werten abgeleitet (globale Genauigkeitsanalyse). Die statistischen Größen sind nachfolgend immer auf die Klasse Pflanze bezogen, da für den späteren Klassifikator (Karotte/Beikraut) nur die Pflanzenpixel relevant sind. Es ist dabei zu erkennen, dass die höchsterreichbare Genauigkeit in dem Datensatz der trockenen Bilder erfolgt ist. Dies liegt unter anderem an den sehr hellweißen Bereichen im Hintergrund (siehe dazu Abbildung 28). Weiter ist der Datensatz der nassen Bilder durch den sehr geringen Farbwertunterschied der einzelnen Pixel im grünen Farbkanal zwischen Pflanzen und Hintergrund sehr schlecht zu trennen. In den Klammern sind die jeweiligen besten Schwellwerte und ihre jeweiligen Genauigkeiten angegeben. Die Zahlenwerte der Precision, des Recall und des F1-Scores, an denen die Genauigkeit den besten Wert hat, sind in der nachfolgenden Tabelle aufgeführt.

Tabelle 1: Statistische Auswertung des einfachen Schwellwertes an der Position des besten Genauigkeitswertes der Analyse ohne einzelne Bildelemente

Datensatz	Bester Schwellwert	Genauigkeit	Precision	Recall	F1-Score
Alle	118	0,68	0,65	0,79	0,71
Nass	126	0,57	0,55	0,88	0,67
Normal	115	0,65	0,64	0,72	0,67
Trocken	127	0,76	0,74	0,80	0,77

Wie an den Werten zu sehen ist, werden 21% der Pflanzenpixel in dem Datensatz der alle Bilder einschließt nicht erkannt und es sind lediglich 65% der erkannten Pflanzenpixel wirklich Pflanzenpixel. Analog ist die Aufsplittung des Datensatzes in die einzelnen Rubriken. Der Datensatz der trockenen Bilder bringt hier das beste Ergebnis.

Durch erneutes Anwenden des besten Schwellwertes auf die Einzelbilder, wobei diesmal die Genauigkeit über jedes Einzelbild ermittelt wird, soll eine statistische Signifikanz der Ergebnisse bestimmt werden. Es handelt sich bei der Methode um einen linksseitigen Test, da der Ablehnungsbereich links des Erwartungswertes liegt. Die verwendete Skalierung zwischen Pflanzenpixel und Hintergrundpixel verwendet einen Skalierungsfaktor, der alle Pixel aller Bilder einbezogen und anschließend auf jedes Bild angewendet worden ist. Durch die unterschiedliche Anzahl

der Pixel, die durch das Filtern der Kanten zwischen Pflanzen und Hintergrund entsteht, wird für jedes Bild der Grenzwert neu bestimmt. Es soll auf Signifikanz (95%; $r = 1,64$) und Hochsignifikanz (99%; $r = 2,33$) getestet werden. Da jedoch Genauigkeiten, die besser sind als der Grenzwert, das Ergebnis nicht negativ beeinflussen sollen, wird lediglich die untere Grenze für die Messung verwendet. Die Ergebnisse des einfachen Schwellwertes ergaben, dass auf dem Datensatz, der alle Bilder beinhaltet, 50% der Bilder nicht innerhalb des Bereiches von 95% liegen. Selbst bei einem Herabsetzen der Genauigkeit und damit des Erwartungswertes zeigt dies bei einer Genauigkeit von 50% immer noch circa 22% der Bilder außerhalb des Bereiches. Bei einer Genauigkeit von 33% liegen 8% der Bilder außerhalb des Bereiches. Erst ab 8% liegt eine Signifikanz und hohe Signifikanz vor. Damit ist ein Genauigkeitswert von 68%, wie es oben beschrieben ist, fragwürdig¹⁰.

Der einfache Schwellwert soll lediglich als Vergleich gelten, um zu zeigen, dass mit diesem relativ einfachen Verfahren nicht möglich ist, zu segmentieren, daher sollen trotz der Signifikanzanalyse dieselben Auswertungen betrachtet werden, wie sie für die anderen Verfahren ebenfalls berechnet wurden.

Wird eine Analyse über jedes einzelne Bild vorgenommen, so dass die Genauigkeit, die Precision, der Recall-Wert und der F1-Score erst von jedem Bild mit jedem Schwellwert berechnet wird und anschließend über die Bilder mit einem Schwellwert die Mittelwerte gebildet werden, ergeben sich die Graphenverläufe in Abbildung 32. Die rechte Abbildung zeigt der Vollständigkeit halber die ROC-Kurve, die Standardabweichung wurde jedoch zur besseren Übersicht entfernt.

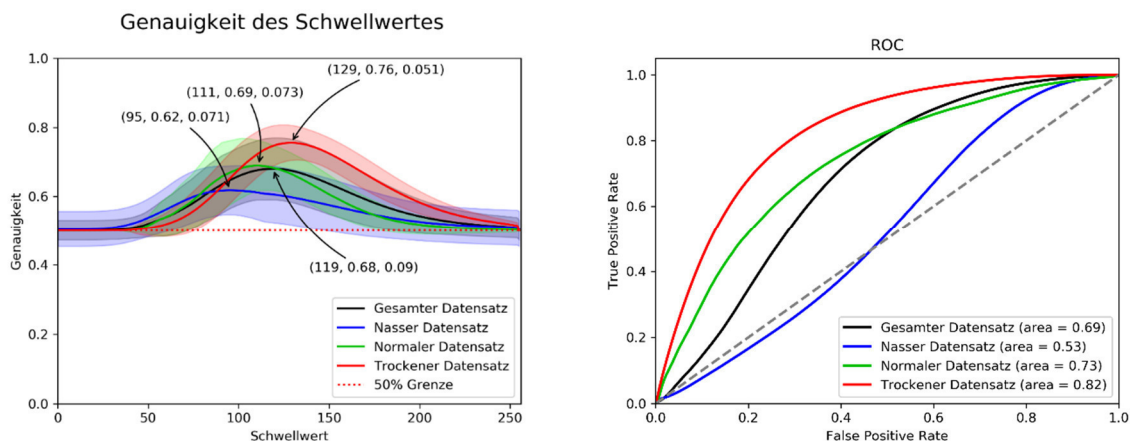


Abbildung 32: Genauigkeitsverlauf des einfachen Schwellwertes, Analyse über einzelne Bildelemente, sowie rechts die ROC-Kurve

Dadurch, dass hier jedes Bild für sich separat betrachtet wird, wurde auch für jedes Bild und jeden Schwellwert einzeln entschieden, ob der normale Schwellwert oder der invertierte Schwellwert das bessere Ergebnis in Blick auf die Genauigkeit bringt. Auch in dieser Abbildung sind durch die Klammern die Maxima der Graphen angegeben. Der dritte Wert zeigt die Standardabweichung an den jeweiligen Punkten. Die Standardabweichung wird durch das Band um den Graphen im Verlauf ebenfalls visualisiert. Ebenfalls sind die Werte der Precision, des Recall und des F1-Score in der Tabelle hinterlegt.

¹⁰ Der Hypothesentest zeigt, dass keine belastbaren Ergebnisse ermittelt werden konnten, und auch die Vermutung nahelegt, dass hier keine Normalverteilung vorliegt wegen der Schwankung trotz der 3σ Regel. Dies könnte an der Skalierung liegen. Da es sich hier jedoch nur um einen Vergleich der einfachsten Herangehensweise handelt, soll hier nicht weiter darauf eingegangen werden.

Tabelle 2: Statistische Auswertung des einfachen Schwellwertes an der Position des besten Genauigkeitswertes der Analyse über einzelne Bildelemente

Datensatz	Bester Schwellwert	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	119	0,68 (0,090)	0,69 (0,110)	0,70 (0,235)	0,66 (0,168)
nass	95	0,62 (0,071)	0,66 (0,123)	0,51 (0,176)	0,55 (0,127)
normal	111	0,69 (0,073)	0,67 (0,066)	0,75 (0,151)	0,76 (0,084)
trocken	129	0,76 (0,051)	0,73 (0,047)	0,81 (0,103)	0,76 (0,058)

Der Datensatz der nassen Bilder hat die größte Unterscheidung zur vorherigen Messung. Gravierend ist hier der Abfall des Recall-Wertes auf 0,51. Das bedeutet, dass nur noch circa die Hälfte aller Pflanzenpixel als Pflanzenpixel erkannt wird. Die anderen Werte sind in einem ähnlichen Wertebereich geblieben. Die Standardabweichung zeigt auch die Instabilität über die Bilder. Als letzte Untersuchung soll davon ausgegangen werden, dass ein fiktiver Algorithmus für jedes separate Bild den besten Schwellwert (an dem die Genauigkeit am höchsten ist) berechnet und anwendet. Die Standardabweichung zeigt, dass die Schwankung der Genauigkeit über die einzelnen Bilder mit einem dieser festen Schwellwerte unter einem Prozent liegt.

Tabelle 3: Statistische Auswertung des einfachen Schwellwertes unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild

Datensatz	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	0,72 (0,082)	0,70 (0,088)	0,77 (0,135)	0,72 (0,098)
nass	0,66 (0,079)	0,64 (0,100)	0,72 (0,157)	0,66 (0,107)
normal	0,72 (0,076)	0,71 (0,076)	0,77 (0,148)	0,73 (0,093)
trocken	0,77 (0,049)	0,74 (0,050)	0,82 (0,058)	0,78 (0,047)

Mit Hilfe der Auswahl des besten Schwellwertes lassen sich die Werte nur um wenige Prozent steigern. Die Genauigkeit erreicht für den Datensatz, der alle Bilder beinhaltet 72% mit einer Standardabweichung von circa 8 Prozentpunkten. Weiter ist die Precision auf 70% gestiegen und der Recallwert auf 77%. Die Untersuchung zeigt, dass mit einem einfachen globalen Schwellwertverfahren jedoch nur eine Genauigkeit von 72% erreicht werden kann.

Für den Klassifikator würde dies bedeuten, dass er Bearbeitungszeit investieren muss, um falsch maskierte Hintergrundpixel zu klassifizieren, und dass ihm Pixel von Pflanzen fehlen, die ausschlaggebend für eine Klassifikation wären, zum Beispiel bei sehr kleinen Pflanzen, die nur mit wenigen Pixeln abgebildet werden. Das bedeutet nicht, dass mit diesem Segmentierer nicht klassifiziert werden könnte, jedoch muss seine Unzulänglichkeit vom Klassifikator wieder ausgeglichen werden.

3.6.4 Vegetationsindexverfahren

Für die Bisppektralkamera gibt es ein einfaches Verfahren, das in verschiedenen Anwendungen verwendet wird, um Pflanzen vom Hintergrund zu trennen. Es gibt hier verschiedene ähnliche Ansätze. Das hier verwendete ist das NDVI (Normalized Difference Vegetation Index) [Rouse, et al., 1974]. Es kann über die Formel (65) berechnet werden [Matsushita, et al., 2007], [Oebel, 2006]:

$$NDVI(x, y) = \frac{\Psi_{NIR}(x, y) - \Psi_{Rot}(x, y)}{\Psi_{NIR}(x, y) + \Psi_{Rot}(x, y)} \quad (65)$$

Wobei $\Psi(x, y)$ das vier Kanal Bild (RGB + NIR) darstellt und der Index den jeweiligen Kanal. Der resultierende Wert $NDVI$ hat durch seine Normierung durch seinen Zähler den Wertebereich von -1 bis

+1. Das Verfahren kann auch verwendet werden, um bestimmte Eigenschaften der Pflanzen zu berechnen. Das Verfahren ist auf einem Bildpaar der JAI-Kamera in Abbildung 33 dargestellt. An der linken Seite ist das RGB und NIR Bild und rechts das resultierende NDVI Bild in der dargestellten Farbskala.

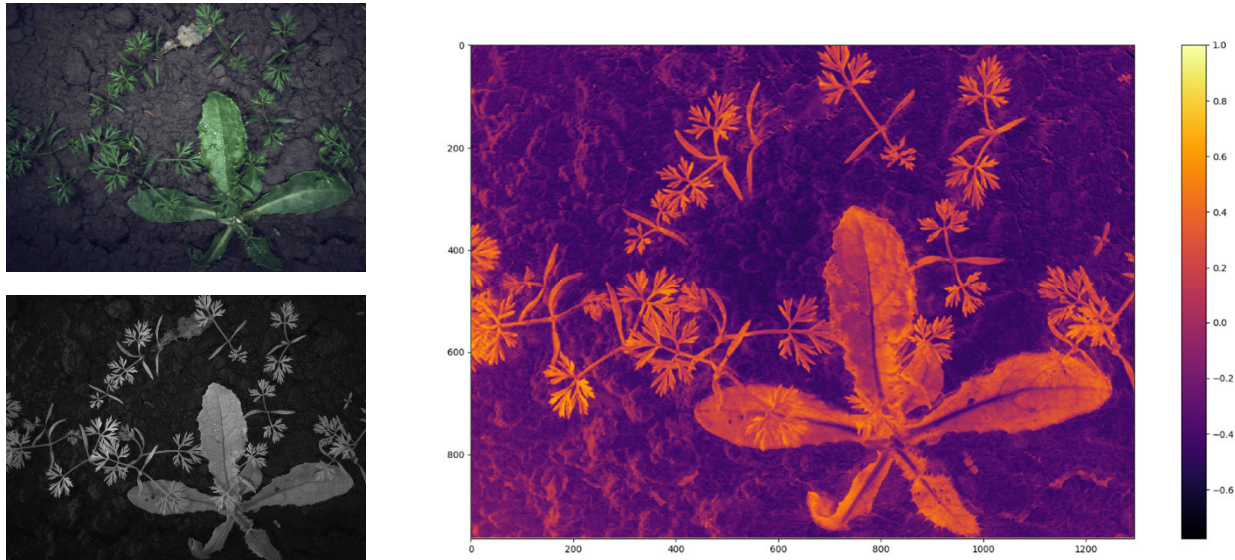


Abbildung 33: *NDVI Verfahren*

Ein NDVI Wert unter Null könnte unter anderem zum Beispiel Wasser sein. Sollte der Wert in der Nähe von Null liegen, handelt es sich um Erdboden. Alles über circa 0,01 stellt Vegetation da. Diese Skalierung stammt aus der Fernerkundung, indem mittels Satellit oder Flugzeug Waldbestände untersucht werden können [Schüttler, et al., 2017], [Ganie & Nusrath, 2016]. Die Werte lassen sich jedoch auch auf den Nahbereich anwenden. Für eine Segmentierung kann auf die NDVI Werte ein Schwellwert angewendet werden. Hier wurde 0,01 verwendet. Die entstehende Maske wurde mittels logischem „Und“ mit dem RGB Bild verknüpft. In Abbildung 34 ist der Hintergrund durch den Schwellwert mit dem Wert Null, also schwarz belegt. Wie in der Abbildung zu erkennen ist, ist auch dieses Verfahren nicht zu 100% fehlerfrei. Pflanzenteile, die einen hohen Wasseranteil haben, wie es in der großen Blattstruktur zu sehen ist, werden hier durch ihren Wert mit dem Schwellwert auf null gesetzt. In der praktischen Anwendung wird auf die Normierung des NDVI verzichtet und dafür der Schwellwert angepasst, dadurch entfällt eine zeitaufwändige Division.



Abbildung 34: *Maskiertes RGB Bild einer JAI-Kamera*

Wird der Algorithmus auf die Evaluierungsdaten angewendet, wobei die Schwellwerte von -1 bis +1 in der Schrittweite von 0,1 durchlaufen werden, ergibt sich folgender Graph für die Genauigkeit über alle Bilder (globale Genauigkeitsanalyse). Die Werte in den Klammern zeigen erneut den Schwellwert, bei dem das angegebene Maximum des Graphen erreicht worden ist. Hier wurde eine Skalierung zwischen Pflanzenpixel und Hintergrundpixel verwendet. Der Skalierungsfaktor wurde dabei aus der Summe aller Pixel aller Bilder berechnet. Rechts in der Abbildung ist die ROC-Kurve abgebildet.

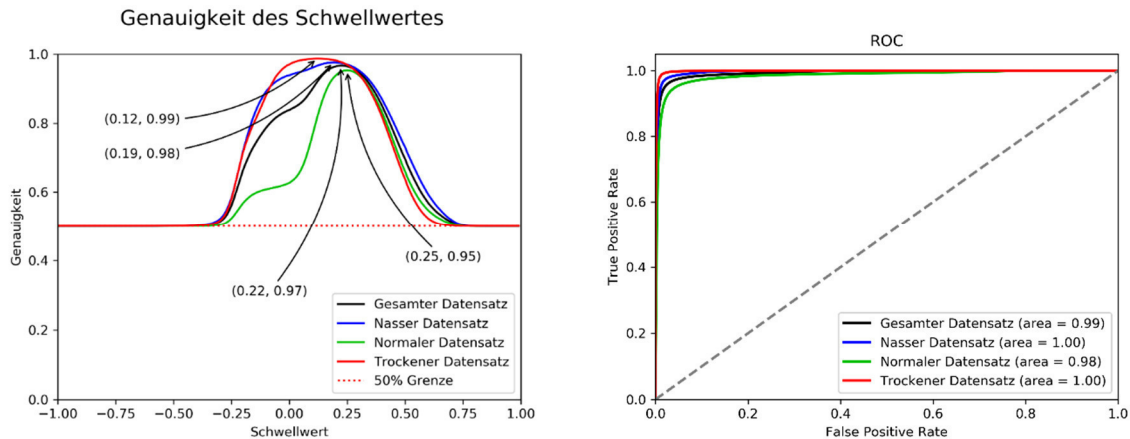


Abbildung 35: Genauigkeit über alle Bilder von dem NDVI-Verfahren sowie rechts die ROC-Kurve

Auch hier sind die Werte für die Precision, des Recall und des F1-Score für die Maxima der Genauigkeit in der Tabelle angegeben.

Tabelle 4: Statistische Auswertung des NDVI-Verfahrens an der Position des besten Genauigkeitswertes der Analyse ohne einzelne Bildelemente

Datensatz	Bester Schwellwert	Genauigkeit	Precision	Recall	F1-Score
alle	0,22	0,97	0,97	0,96	0,97
nass	0,19	0,98	0,98	0,98	0,98
normal	0,25	0,95	0,96	0,95	0,95
trocken	0,12	0,99	0,99	0,99	0,99

Auf den ersten Blick weisen diese Werte sowohl in der Genauigkeit, als auch in der Precision und dem Recall hohe Werte auf. Die Ergebnisse zeigen, dass es minimale Unterschiede zwischen den verschiedenen Datensätzen gibt. Die verschiedenen Umwelteinflüsse der Datensätze (nass, normal und trocken) zeigen, dass das NDVI-Verfahren demgegenüber robust ist. Auch die Graphen in der ROC-Kurve spiegeln das Ergebnis deutlich wieder.

Wie bei dem einfachen Schwellwert soll auch hier eine Signifikanzanalyse erfolgen. Auch hier soll der Schwellwert verwendet werden, der über den gesamten Datensatz (ohne Betrachtung der einzelnen Bilder) die besten Ergebnisse erzielte. Die Genauigkeit, die der Schwellwert über alle Pixel und Bilder lieferte, soll nun mit den Genauigkeiten verglichen werden, die über die einzelnen Bilder bestimmt werden. Dabei wurde der verwendete Skalierungsfaktor (zwischen Pflanzenpixel und Hintergrundpixel) wieder über die Summe aller Bilder bestimmt und auf die Einzelbilder angewendet. Wie vorher auch, soll auf Signifikanz (95%; $r = 1,64$) und Hochsignifikanz (99%; $r = 2,33$) geprüft werden, wobei auch hier nur die untere Grenze für die Betrachtung verwendet werden soll, da bessere Genauigkeiten das Ergebnis nicht beeinflussen. Die Analyse des NDVI-Verfahrens mit der Genauigkeit von circa 97% und dem Schwellwert von 0,22 erwies, dass 44% der Bilder nicht im signifikanten Bereich liegen. Wenn eine Genauigkeit von 87,3% verwendet wird, sind nur zwei Bilder außerhalb des Bereiches. Auch bei der Hochsignifikanz-Analyse sind ab einem Genauigkeitswert von 87,0% nur zwei Bilder außerhalb des

Bereiches. Die letzten beiden Bilder können erst bei circa 40% in die beiden Messungen integriert werden.

Für die Auswertung über die einzelnen Bilder (lokale Genauigkeitsanalyse) ergeben sich ähnliche Werte wie bei der Messung ohne Einzelbildanalyse. Anders als bei dem Signifikanztest, wird die Skalierung hier jedoch von jedem Bild separat berechnet. Die Genauigkeit ist in Abbildung 36 dargestellt. Die Werte in den Klammern zeigen den Schwellwert, bei dem das angegebene Maximum erreicht worden ist. Der letzte Wert in der Klammer steht für die Standardabweichung. Diese ist ebenfalls durch das Band um die jeweiligen Graphen angegeben. Rechts ist die dazugehörige ROC-Kurve angegeben. Die Standardabweichung wurde der Übersichtlichkeit halber nicht angegeben.

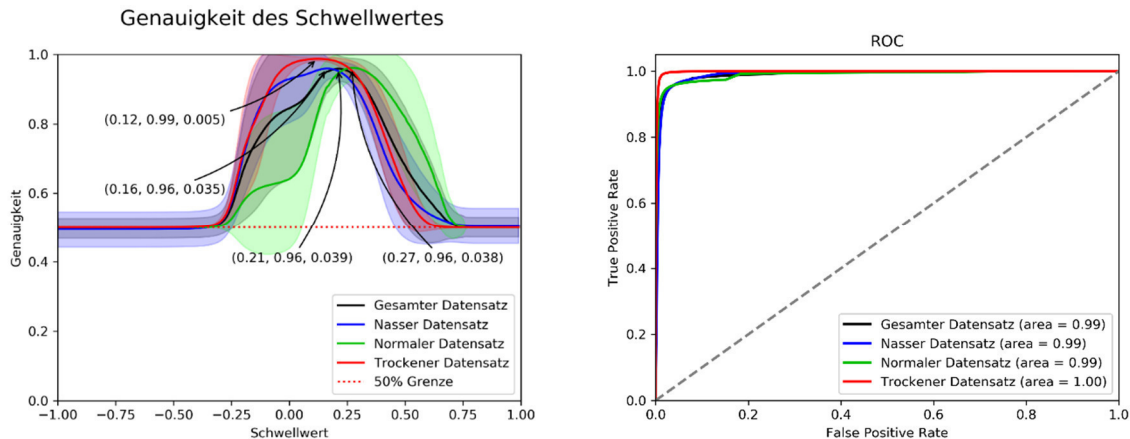


Abbildung 36: Genauigkeitsverlauf des NDVI-Verfahrens, Analyse über einzelne Bildelemente sowie rechts die ROC-Kurve

Auch hier sollen die Maximalwerte in der nachfolgenden Tabelle zur Übersicht aufgelistet werden. Es ist zu erkennen, dass auch diese hohen Werte sowohl für die Genauigkeit, Precision und Recall für denselben Schwellwert auf den Testdaten erreicht werden.

Tabelle 5: Statistische Auswertung des NDVI-Verfahrens an der Position des besten Genauigkeitswertes der Analyse über einzelne Bildelemente

Datensatz	Bester Schwellwert	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	0,21	0,96 (0,039)	0,96 (0,074)	0,95 (0,056)	0,96 (0,068)
nass	0,16	0,96 (0,035)	0,96 (0,114)	0,96 (0,038)	0,95 (0,104)
normal	0,27	0,96 (0,038)	0,97 (0,029)	0,95 (0,076)	0,96 (0,044)
trocken	0,12	0,99 (0,005)	0,98 (0,010)	0,99 (0,009)	0,99 (0,005)

Es muss hier jedoch bedacht werden, dass die Daten der Auswertung skaliert worden sind. Wie vorher bei dem einfachen Schwellwert wurden die Anzahl der Pflanzenpixel und der Hintergrundpixel in Verhältnis gesetzt, damit der Hintergrund nicht mit seiner Häufigkeit die Statistik beeinflusst. Auf den Bildern befindet sich jedoch eine größere Anzahl der Hintergrundpixel als der Pflanzenpixel. Wird der Precisionverlauf betrachtet ohne Skalierung (wie in Abbildung 37) ändert sich der Verlauf. Mit Hilfe des Precisionwertes wird angegeben, wie viel Fremdkörper sich innerhalb der Menge der Positiv (Pflanzenpixel) klassifizierten Ergebnisse befinden. Er gibt an, wie viele Hintergrundpixel sich innerhalb der Gesamtzahl der klassifizierten Pflanzenpixel befinden. Wird die Skalierung für die Messung nicht verwendet, existieren mehr Hintergrundpixel für die statistische Auswertung. Damit erhöht sich trivialerweise auch die Anzahl der als False-Positiv klassifizierten Pixel und somit nimmt der Precisionwert ab. Für die Messung über alle Bilder ergibt sich für die beste Genauigkeit ein Precisionwert von 85% beim gesamten Datensatz. Zu bedenken ist hier, dass sich der Schwellwert und

auch der Genauigkeitswert für diese Messung zu der vorherigen Messung geändert haben, da der Datensatz nicht skaliert ist. Die Genauigkeiten¹¹ für die besten Werte des Datensatzes über alle Bilder sind 99% (Analyse ohne einzelne Bilder) und ebenfalls 99% (Analyse über einzelne Bilder). Für die Betrachtung der einzelnen Bilder in der Messung ergibt sich ein Wert von 68% mit einer Standardabweichung von 36 Prozentpunkten. Das bedeutet für eine reale Messung, dass im Schnitt nur 68% der erkannten Pflanzenpixel wirklich Pflanzenpixel sind. Die Standardabweichung zeigt dazu, dass dieser Mittelwert über die einzelnen Bilder stark schwankt. Die Unterschiede zwischen den einzelnen Datensätzen basieren auf der unterschiedlichen Verteilung der Pflanzenpixel zu Hintergrundpixel. Im Datensatz der normalen Wetterbedingung befinden sich Pflanzen, die kleiner sind als in den anderen Datensätzen. Dadurch ist der Anteil der Hintergrundpixel größer. Ohne Skalierung besitzt dieser Datensatz daher eine größere Wahrscheinlichkeit für einen höheren False-Positiv-Wert.

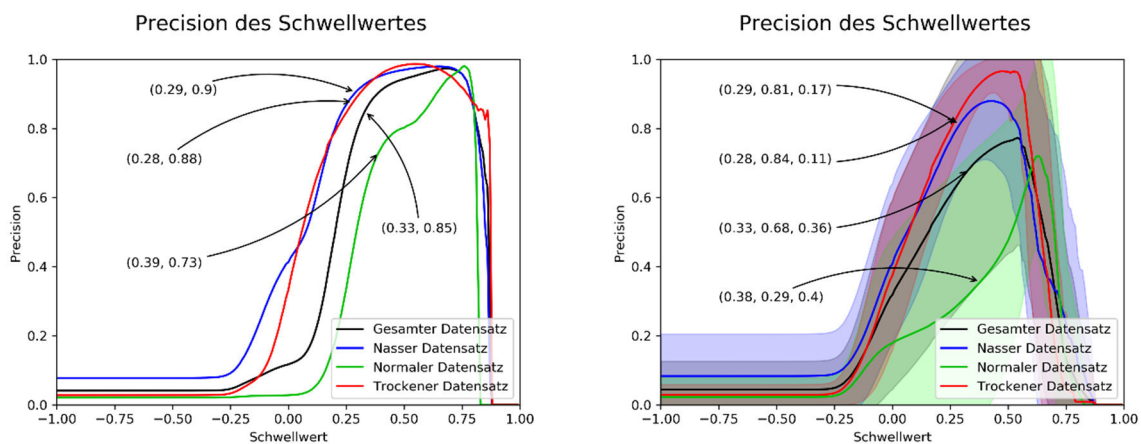


Abbildung 37: Precisionswerte ohne Skalierung für das NDVI-Verfahren. Links Analyse über alle Bilder; rechts Analyse über Einzelbildmessung

Der Unterschied der Precision zwischen der Betrachtung über alle Bilder (linker Graph in Abbildung 37) und der Betrachtung über einzelne Bilder (rechter Graph) liegt an der Anzahl der zu mittelnden Ereignisse. Wenn für jedes Bild einzeln die Precision berechnet wird und die einzelnen Precisionen anschließend gemittelt werden (rechter Graph), wirkt sich der False-Positiv-Wert nicht so stark aus, als wenn über alle Bilder der False-Positiv-Wert summiert wird und anschließend damit die Precision berechnet wird (linker Graph). Wenn zum Beispiel fiktiv ein gesamtes Bild nur aus False-Positiv-Werten bestehen würde, würde dieser entweder mit seiner kompletten Anzahl an Pixeln in die Precision eingerechnet werden (linker Graph), wohingegen bei der Mittelung über einzelne Bilder dieses Extrem lediglich ein Bild aus der gesamten Bildmenge darstellt.

Abschließend soll vollständigshalber auch hier ein fiktiver Algorithmus für jedes Bild den optimalen Schwellwert bestimmen. Für die Auswertung wurde wieder die Skalierung verwendet. Mit Hilfe des fiktiven Algorithmus lässt sich nach rechnerischen Werten die Genauigkeit für den Datensatz, der alle Bilder beinhaltet, um circa 2% verbessern, wie dies in der untenstehenden Tabelle dargestellt wird. Das NDVI-Verfahren erreicht schon mit einem globalen festen Schwellwert über alle Bilder Werte im oberen neunziger Prozentbereich. Daher ist die Steigerung und Auswirkung des fiktiven Algorithmus nicht gravierend.

¹¹ Zur Erinnerung, würde der Segmentierer bei dieser Messung alle Pixel als Hintergrundpixel klassifizieren, würde die Genauigkeit bereits bei circa 93% liegen, da 93% aller Pixel im Schnitt Hintergrundpixel sind.

Tabelle 6: Statistische Auswertung des einfachen Schwellwertes unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild

Datensatz	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	0,98 (0,017)	0,98 (0,061)	0,99 (0,021)	0,98 (0,060)
nass	0,98 (0,012)	0,96 (0,101)	0,98 (0,012)	0,97 (0,102)
normal	0,98 (0,025)	0,98 (0,032)	0,98 (0,033)	0,98 (0,025)
trocken	0,99 (0,004)	0,99 (0,005)	0,99 (0,004)	0,99 (0,004)

Die Segmentierung mittels NDVI zeigt deutlich bessere Werte als beim einfachen Schwellwert. Nachteil ist jedoch das komplexere Setup der Hardware. Es wird eine Bispektralkamera benötigt und eine Lichtquelle, die sowohl sichtbares als auch NIR-Licht erzeugt. Ein Vorteil wäre eventuell, dass das NIR-Bild der Kamera zusätzliche Informationen zur Klassifikation zur Verfügung stellt.

3.6.5 RGB Segmentierungsalgorithmus

Das NDVI-Verfahren benötigt zu dem roten Kanal im RGB-Bild zwangsweise einen NIR-Kanal in der Kamera. Das bedeutet, dass für eine reine RGB-Kamera ein anderes Verfahren angewendet werden muss. Das in dieser Arbeit entwickelte Verfahren ist als Flussdiagramm in Abbildung 38 dargestellt und wurde in [Knoll, et al., 2016 c] veröffentlicht.

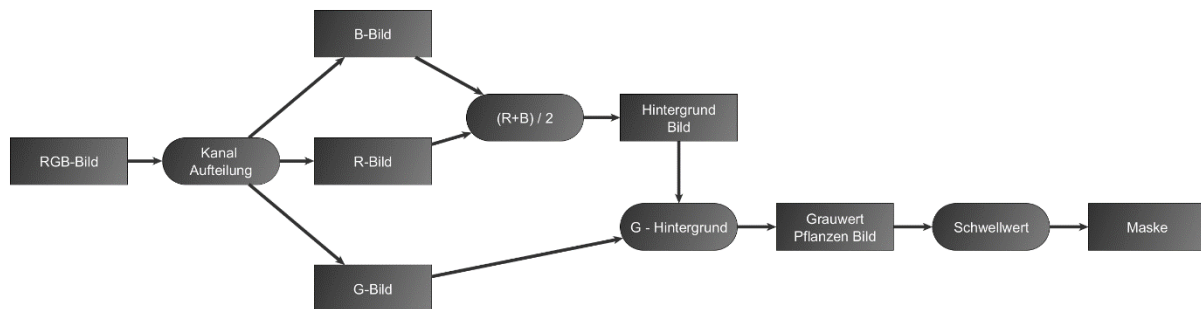


Abbildung 38: Flussdiagramm des Segmentierungsalgorithmus für den RGB Farbraum

Wie im Flussdiagramm in Abbildung 38 dargestellt, wird das RGB-Bild in seine drei Komponenten aufgeteilt. Der erste Schritt und seine Eingangsinfomation ist in Abbildung 39 veranschaulicht. Es ist nicht überraschend, dass bei genauer Betrachtung die Pflanzen einen hohen grünen Farbanteil besitzen (mittleres Bild unten). Ein Schwellwert auf dem Grünkanal, um die Pflanzen zuverlässig vom Hintergrund zu segmentieren, ist jedoch wie in Kapitel 3.6.3 gezeigt, nicht möglich, da einige Teile der Umgebung nicht herausgefiltert werden können. Durch die Betrachtung des roten (unten links) und des blauen Farbkanals (unten rechts), ist zu erkennen, dass hier die Pflanzen einen niedrigen Farbwert aufweisen. Außerdem ist ersichtlich, dass die Erdpixel in allen drei Kanälen einem ähnlichen Wertebereich unterliegen. Das Eingangsbild (oben) stammt hier beispielhaft von der Nikon-Kamera.

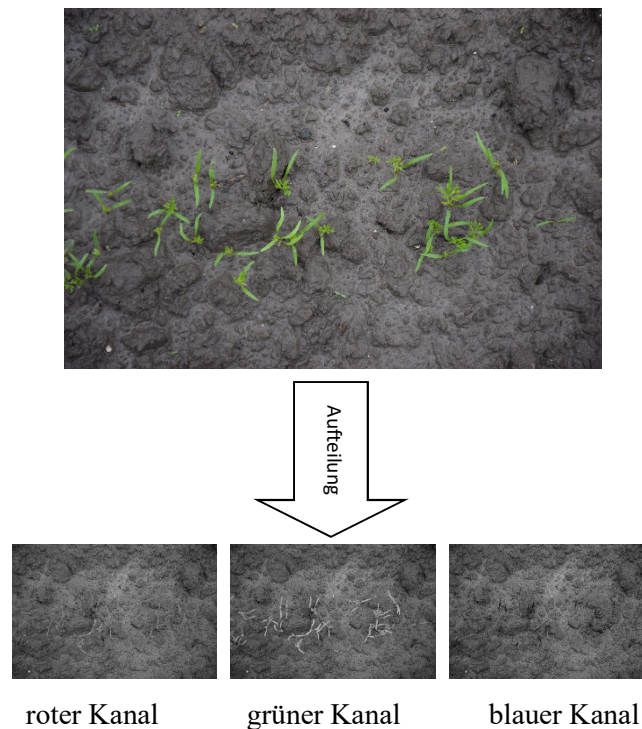


Abbildung 39: Das originale RGB Bild (oben) und unten von links, der rote Kanal, der grüne Kanal und der blaue Kanal als Grauwertbild

Werden die drei Farbkanaäle mit einem Histogramm dargestellt, wird der Zusammenhang ersichtlich. In dem Histogramm ist die jeweilige Häufigkeit der Frequenz mit ihrer jeweiligen Farbe (Rot, Grün und Blau) dargestellt. Alle drei Graphen haben einen identischen Verlauf. In Abbildung 39 ist dies dadurch zu erkennen, dass der Erdboden in allen drei Farbkälen einen ähnlichen Helligkeitswert aufweist. Lediglich der grüne Kanal weist eine leichte Erhebung im rechten Bereich auf (Markierung durch gelben Kreis im Histogramm). Diese Häufigkeitsansammlung kommt durch die Pflanzen zustande. Durch eine Differenzierung des grünen Kanals und eines anderen Kanals (Rot oder Blau) können die Pflanzen vom Hintergrund getrennt werden.

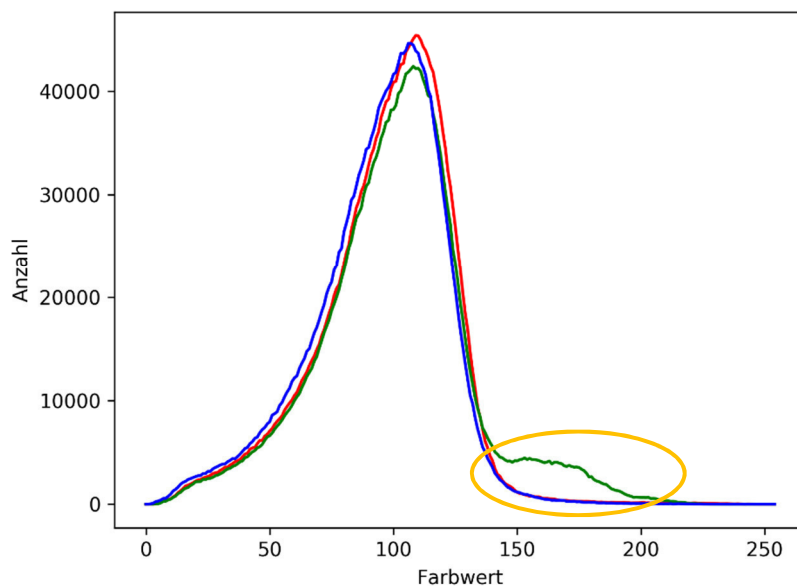


Abbildung 40: Histogramm der drei Farbkanaäle

Die Verwendung eines Mittelwertes des roten und blauen Farbkanals erwies sich als stabilere Lösung, als lediglich einen Kanal für die Subtraktion zu verwenden. Wird die Mittelwertberechnung für jeden Pixel ausgeführt, ergibt sich ein neues Bild, das den Namen *BackImg* hat. Die Berechnung, die für jeden Pixel angewendet werden muss, ist mit der Formel (66) dargestellt. Hier ist $\Psi(x, y)$ der jeweilige Pixelwert des Eingangsbildes. Der Index gibt an, um welchen Farbkanal es sich handelt.

$$BackImg(x, y) := \frac{\psi_{Rot}(x, y) + \psi_{Blau}(x, y)}{2} \quad (66)$$

Das *BackImg* wird anschließend von dem grünen Kanal des Eingangsbildes abgezogen und als Absolutwert¹² in *PlantImg* gespeichert (wie in Formel (67)):

$$PlantImg(x, y) := |\psi_{Grün}(x, y) - BackImg(x, y)|. \quad (67)$$

Das resultierende *PlantImg*, ist ein Grauwertbild, in dem sich die Pflanzen signifikant mit ihren Werten vom Hintergrund abheben. Das Ergebnis ist in Abbildung 41 dargestellt.

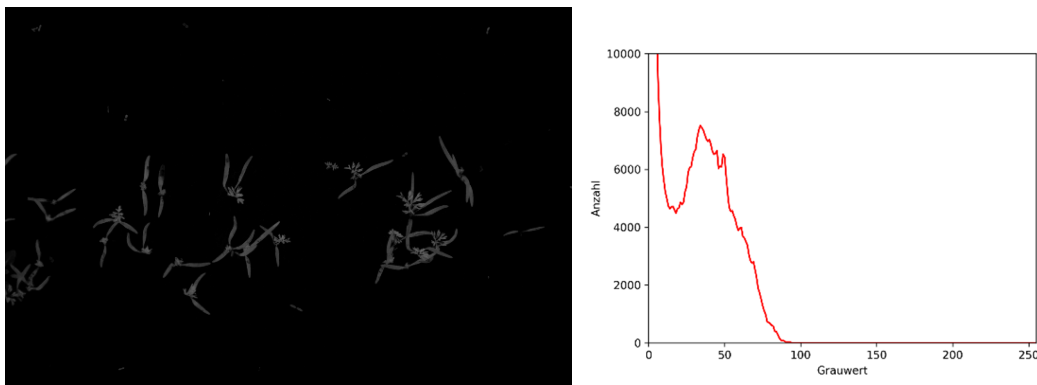


Abbildung 41: Darstellung des Grauwertbildes mit dem Namen *PlantImg* und sein Histogramm

Wird das *PlantImg* in einem Histogramm abgebildet (wie in Abbildung 41 rechts dargestellt) ergibt der entstehende Graph einen bimodalen Verlauf. Das sich auf der linken Seite befindende Maximum ist der dunkle Hintergrund, dessen Helligkeitswerte sich durch die Subtraktion der Formel zu dieser Stelle verschoben haben. Das rechte Maximum sind die Pflanzen. Das Minimum zwischen den beiden Maxima ergibt den Schwellwert, um eine Maske zu erstellen. Das Ergebnis der Maske ist in Abbildung 42 dargestellt. Außerdem sind in Abbildung 42 rechts die maskierten Pflanzen des Eingangsbildes dargestellt.

¹² Der Betrag in der Formel ermöglicht es, das Ergebnisbild als Grauwertbild darzustellen, ohne eine Skalierung vorzunehmen. Ohne die Skalierung wären auch negative Schwellwerte denkbar, die jedoch keinen weiteren Vorteil für die Pflanzensegmentation bringen würden. Es existieren hier zwei Fälle: Einmal handelt es sich um einen Pflanzenpixel, hier ist $\psi_{Grün}$ größer als der Mittelwert von Rot und Blau, oder es handelt sich um Hintergrund, hier kann der Mittelwert von Rot und Blau zwar größer sein als der grüne Wert, jedoch ist $\psi_{Grün}(x, y) - BackImg(x, y)$ kleiner als das Äquivalent bei einem Pflanzenpixel. Daher fällt der Betrag nicht ins Gewicht.

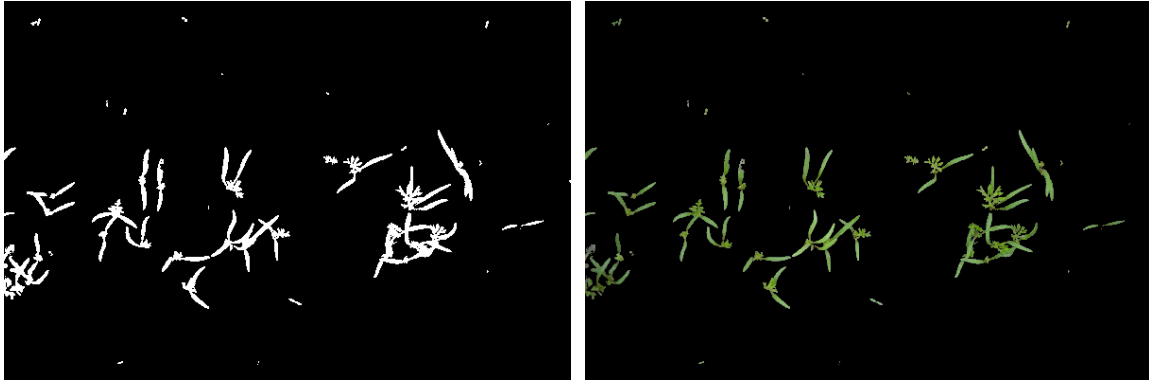


Abbildung 42: *Maske des RGB Algorithmus und das maskierte Eingangsbild als Ergebnis*

In der oberen Bildreihe in Abbildung 43 ist die Analyse über alle Bilder ohne die Betrachtung einzelner Bildelemente ersichtlich (globale Genauigkeitsanalyse) sowie die ROC-Kurve. Unten wird die Analyse über einzelne Bilder dargestellt (lokale Genauigkeitsanalyse und ROC-Kurve). Beide Berechnungen verwendeten die Skalierung für die gleiche Verteilung der Anzahl zwischen Pflanzenpixel und Hintergrundpixel. Wie die Diagramme zeigen, ist die Genauigkeit mit dem NDVI-Verfahren vergleichbar. Auch hier existieren nur leichte Unterschiede zwischen den verschiedenen Datensätzen, die zum Testen verwendet wurden. Die True-Positiv Werte bilden bei einem Schwellwert von 0 das Maximum. Dies bedeutet jedoch nicht, dass alle Pixel als Pflanzen segmentiert werden. Mit steigendem Schwellwert kann der True-Positiv-Wert daher nur gleichbleiben oder abnehmen. Eine Zunahme ist nicht möglich, da durch den Schwellwert eine feste Grenze besteht, ab dem sich die Klasse für einen Pixel ändert. Die Precision steigt jedoch, da die False-Positiv-Werte abnehmen. Auch dieser Wert kann mit steigendem Schwellwert lediglich sinken und nicht zunehmen. Das bedeutet, dass der Algorithmus bei einem Schwellwert von 0 die größte Anzahl von Pflanzenpixel hervorruft und bei einem Schwellwert von 255 die meisten Pixel als Hintergrund segmentiert. Diese Beschränkung führt dazu, dass True-Negativ nicht Null werden kann und sorgt dafür, dass die ROC-Kurve im Verlauf endet. Durch den Mittelwert der Formel (66) muss der Schwellwert nicht so quantisiert sein, wie es hier angegeben ist. Wenn der Betrag in Formel (67) entfernt werden würde, wären auch negative Schwellwerte denkbar.

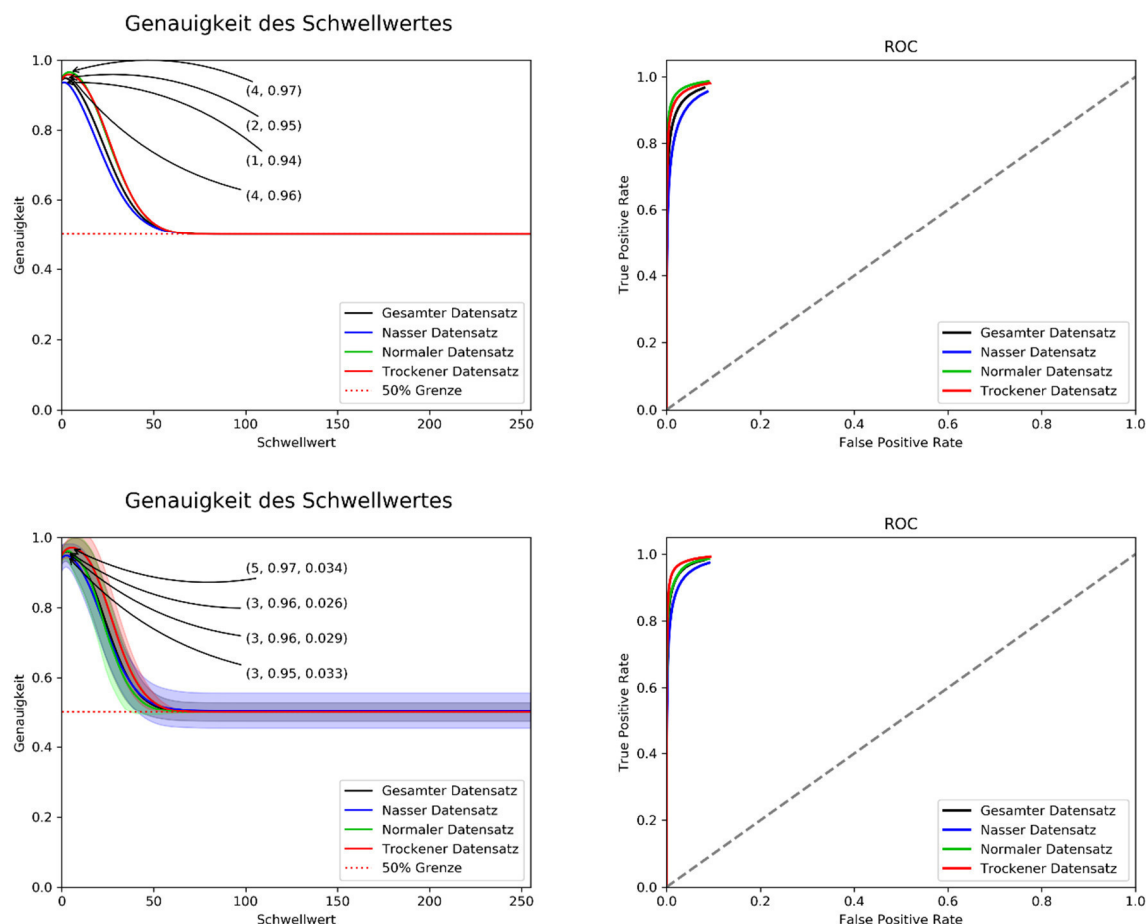


Abbildung 43: Genauigkeitsverlauf des RGB-Segmentierungsverfahrens; Links: Analyse ohne Betrachtung der einzelnen Bilder; Rechts: Analyse über einzelne Bildelemente.

Die Werte für die Precision, den Recall und den F1-Score für die Maximalwerte der Analyse ohne einzelne Bildelemente, stehen in der folgenden Tabelle:

Tabelle 7: Statistische Auswertung des RGB-Segmentierungsverfahrens an der Position des besten Genauigkeitswertes der Analyse ohne einzelne Bildelemente

Datensatz	Bester Schwellwert	Genauigkeit	Precision	Recall	F1-Score
alle	2	0,95	0,95	0,95	0,95
nass	1	0,94	0,93	0,94	0,94
normal	4	0,97	0,97	0,96	0,97
trocken	4	0,96	0,96	0,95	0,96

Alle Werte liegen im mittleren Neunziger-Prozentbereich und weisen wie erwähnt nur leichte Unterschiede auf. Ähnlich sind die Werte für die Analyse, die über einzelne Bildelemente laufen und die in der Tabelle 8 stehen.

Tabelle 8: Statistische Auswertung des RGB-Segmentierungsverfahrens an der Position des besten Genauigkeitswertes der Analyse über einzelne Bildelemente

Datensatz	Bester Schwellwert	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	3	0,96 (0,029)	0,96 (0,058)	0,96 (0,054)	0,96 (0,062)
nass	3	0,95 (0,033)	0,94 (0,104)	0,94 (0,053)	0,94 (0,104)
normal	3	0,96 (0,026)	0,96 (0,010)	0,97 (0,053)	0,96 (0,029)
trocken	5	0,97 (0,034)	0,97 (0,009)	0,97 (0,066)	0,97 (0,039)

Die Ergebnisse sind mit denen des NDVI-Verfahrens vergleichbar, der Vorteil hier ist jedoch, dass eine reine RGB-Kamera ausreicht und kein zusätzlicher IR-Sensor benötigt wird.

Wie bei den anderen Verfahren auch, soll hier auf Signifikanz (95%; $r = 1,64$) und Hochsignifikanz (99%; $r = 2,33$) geprüft werden. Dabei soll wieder der Schwellwert, der über den gesamten Datensatz das beste Ergebnis geliefert hat, auf die Einzelbilder des Datensatzes angewendet werden, um diese zu vergleichen. Auch hier wird die Skalierung verwendet, die über alle Bilder bestimmt wird und auf jedes einzelne Bild anschließend angewendet wird. Wenn der Schwellwert von 2 und die Genauigkeit von 95% als Erwartungswert verwendet werden, liegen 17,7% sowohl bei der Signifikanzmessung, als auch bei der hohen Signifikanzmessung außerhalb des Grenzbereiches. Ab einem Genauigkeitswert von circa 91,5% (signifikant) und 91,3% (hoch signifikant) liegen lediglich 8 Bilder außerhalb. Wenn der Genauigkeitswert auf 85,8% abgesenkt wird, sind für beide Messungen 4 Bilder außerhalb der Grenzen. Damit alle Bilder in den Grenzen liegen, muss die Genauigkeit für beide Messungen mindestens auf 73% herabgesetzt werden. Das bedeutet, das Verfahren erreicht mit hochsignifikanter Sicherheit eine Genauigkeit von mindestens 91%.

Wird die Skalierung auf die Datensätze nicht angewendet und somit eine Verarbeitung unter realen Bedingungen simuliert, verringert sich die Precision wie in Abbildung 44 angeben. Die Genauigkeit in diesen Fall liegt für die Messung über alle Bilder und für die Messung über die Einzelbilder wie beim NDVI-Verfahren bei 99%.

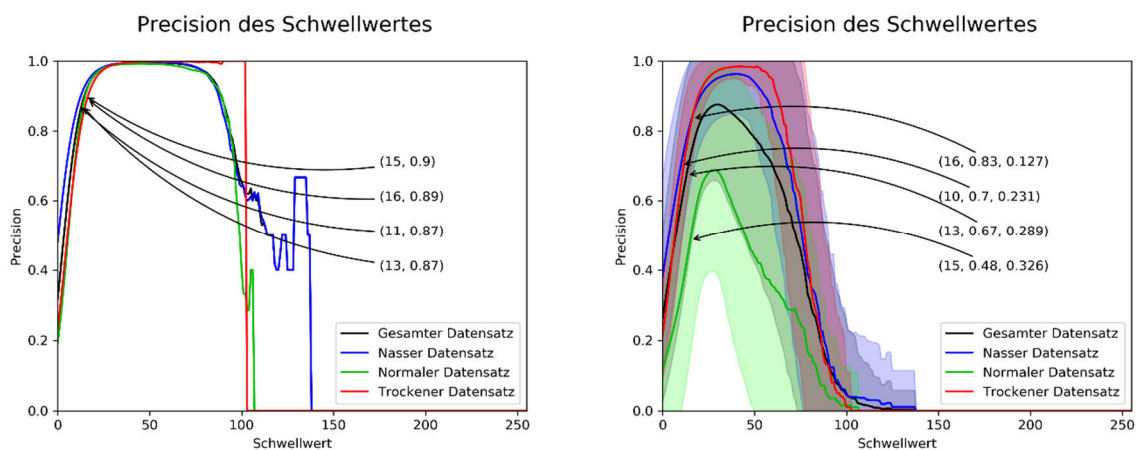


Abbildung 44: Genauigkeitsverlauf des RGB-Segmentierungsverfahrens; Links: Analyse ohne Betrachtung der einzelnen Bilder; Rechts: Analyse über einzelne Bildelemente

Die Precision liegt bei der Messung über alle Bilder bei 87% und bei der Messung über die einzelnen Bilder bei 67%. Damit sind auch diese Werte ähnlich wie beim NDVI-Verfahren. Auch die Standardabweichung zeigt eine ähnliche Schwankung der Werte. Die Sprünge bei der Analyse ohne die Betrachtung der einzelnen Bilder des Datensatzes, der die nassen Bilder beinhaltet, basieren auf der mathematischen Berechnung der Precision. Die Precision setzt sich aus den Parametern der True-Positiven- und der False-Positiven-Werte zusammen. Grob betrachtet bilden diese beiden Größen ein

Verhältnis. Wenn nun die False-Positiv-Werte sinken und die True-Positiv-Werte gleich bleiben, kommt ein Sprung zustande.

Der fiktive Algorithmus, der für jedes zu segmentierende Bild den besten Schwellwert bestimmt, ergibt die Werte wie in der Tabelle 9. Auch diese Werte bringen nur eine leichte Verbesserung.

Tabelle 9: Statistische Auswertung des RGB-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild

Datensatz	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	0,97 (0,027)	0,97 (0,059)	0,97 (0,033)	0,97 (0,060)
nass	0,96 (0,031)	0,95 (0,103)	0,96 (0,035)	0,95 (0,103)
normal	0,97 (0,023)	0,97 (0,021)	0,97 (0,033)	0,97 (0,024)
trocken	0,98 (0,022)	0,98 (0,022)	0,98 (0,023)	0,98 (0,022)

Die Ergebnisse zeigen, dass der Segmentierer auf demselben Niveau wie das NDVI-Verfahren arbeitet und durch die Verwendung einer herkömmlichen RGB-Kamera ein einfacheres Hardware-Setup benötigt.

3.6.6 HSV-Verfahren

Eine weitere Möglichkeit, die Pflanzen von dem Hintergrund anhand eines RGB-Bildes zu extrahieren ist es, den HSV-Farbraum zu verwenden. Der RGB-Farbraum interpretiert die Farbe und die Helligkeit als gemeinsame Größe, dies ist im HSV-Farbraum getrennt. Damit ist eine separate Betrachtung des Farbwertes und der Helligkeit möglich [Demant, et al., 2011]. Der in dieser Arbeit entwickelte Algorithmus ist in Abbildung 45 dargestellt und wurde in [Knoll, et al., 2016 c] und in [Knoll, et al., 2018] veröffentlicht. Im ersten Schritt wird das RGB-Eingangsbild in den HSV-Farbraum transformiert. Der HSV-Farbraum setzt sich aus den drei Kanälen H (Hue) für den Farbwert, S (Saturation) für die Sättigung und V (Value) für die Helligkeit zusammen.

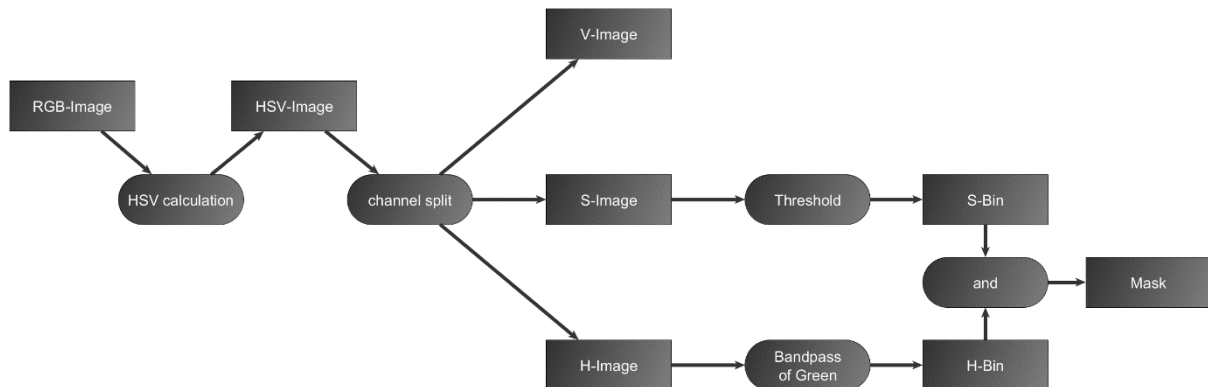


Abbildung 45: Flussdiagramm des Segmentierungsalgorithmus im HSV-Raum

Die Transformation vom RGB- in den HSV-Raum lässt sich über die folgenden Gleichungen berechnen [Ohser, 2018]. Die Berechnung muss dabei für jeden Pixel separat erfolgen. Damit ergibt sich für den Value-Wert die Formel (68). Aus dem RGB-Bild $\psi(x, y)$ wird das Maximum der Helligkeit der Farbwerte (Rot, Grün oder Blau) vom jeweiligen Pixel genommen. Anschließend wird der Wert auf 1 normiert:

$$V(x, y) = MAX(x, y) := \max\left(\frac{\psi_{\text{Rot}}(x, y)}{255}, \frac{\psi_{\text{Grün}}(x, y)}{255}, \frac{\psi_{\text{Blau}}(x, y)}{255}\right). \quad (68)$$

Es soll äquivalent zum Maximum das Minimum (Formel ((69)) definiert werden. Auch hier wird für den jeweiligen Pixel der Wert von Rot, Grün oder Blau verwendet, diesmal jedoch der minimale Wert.

$$MIN(x, y) := \min\left(\frac{\psi_{Rot}(x, y)}{255}, \frac{\psi_{Grün}(x, y)}{255}, \frac{\psi_{Blau}(x, y)}{255}\right) \quad (69)$$

Daraus lässt sich die Formel ((70)) für die normierte Sättigung $S(x, y)'$ aufstellen:

$$S(x, y)' := \begin{cases} \frac{MAX(x, y) - MIN(x, y)}{MAX(x, y)} & \text{wenn } MAX(x, y) \neq 0; \\ 0 & \text{sonst} \end{cases} \quad (70)$$

und die Formeln (71) und (72) für den normierten Farbwert $H(x, y)'$:

$$H(x, y)' := \begin{cases} \frac{60 \cdot \left(\frac{\psi_{Grün}(x, y) - \psi_{Blau}(x, y)}{255}\right)}{MAX(x, y) - MIN(x, y)} & \text{wenn } MAX(x, y) = \frac{\psi_{Rot}(x, y)}{255} \\ \frac{120 + 60 \cdot \left(\frac{\psi_{Blau}(x, y) - \psi_{Rot}(x, y)}{255}\right)}{MAX(x, y) - MIN(x, y)} & \text{wenn } MAX(x, y) = \frac{\psi_{Grün}(x, y)}{255} \\ \frac{240 + 60 \cdot \left(\frac{\psi_{Rot}(x, y) - \psi_{Grün}(x, y)}{255}\right)}{MAX(x, y) - MIN(x, y)} & \text{wenn } MAX(x, y) = \frac{\psi_{Blau}(x, y)}{255} \end{cases} \quad (71)$$

$$H(x, y)' = H(x, y)' + 360 \text{ wenn } H(x, y)' < 0. \quad (72)$$

Für die 8 Bit Darstellung der Bilder werden die Werte skaliert:

$$V(x, y) = V(x, y)' \cdot 255 \quad (73)$$

$$S(x, y) = S(x, y)' \cdot 255 \quad (74)$$

$$H(x, y) = H(x, y)' / 2. \quad (75)$$

In Abbildung 46 ist die Aufteilung eines RGB-Bildes (oben) in den HSV-Raum (unten) verdeutlicht. Der HSV-Raum ist hier durch die drei Grauwertbilder mit den Informationen Farbwert (H), Sättigung (S) und Helligkeit (V) dargestellt.

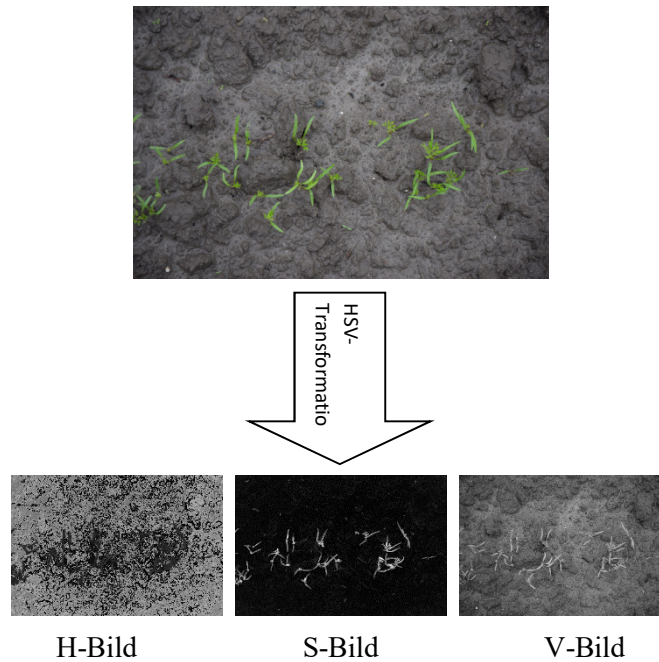


Abbildung 46: Aufteilung und Darstellung eines RGB Bildes in die drei Kanäle des HSV-Raumes

Werden das H-Bild und das S-Bild in Abbildung 46 betrachtet, ist zu erkennen, dass die Pflanzen in den Bildern einen starken Kontrast zum Hintergrund aufweisen. Ein globaler Schwellwert auf dem S-Bild zum Beispiel ist jedoch keine Alternative für eine Segmentierung, da hier zu viele Fragmente vom Hintergrund im Bild erhalten bleiben. Daher muss zuerst mit Hilfe eines Bandpassfilters, das grüne Farbspektrum aus dem H-Bild extrahiert werden. Das Histogramm des H-Bildes ist in Abbildung 47 dargestellt. Der Wertebereich, in dem das grüne Farbspektrum liegt, ist mit einem grünen Kasten umrandet. Im Histogramm kann erkannt werden, dass der Grünanteil im Bild niedriger ist als die anderen Farbanteile. Der Bandpassfilter kann wie folgt beschrieben werden:

$$H(x,y) = \begin{cases} H(x,y) & \text{wenn } 30 \leq H(x,y) \leq 90 \\ 0 & \text{sonst} \end{cases} \quad (76)$$

Die Grenzen des Filters bilden den grünen Farbbereich ab. Der Farbraum ist nicht auf 8-Bit ausgelegt. Der maximale Wert in dem Raum ist 180. Die beiden Grenzen THHmin und THHmax können mit folgender linearer Beziehung berechnet werden.

$$TH_{Hmin} = 60^\circ \triangleq 30 \cdot \frac{360^\circ}{180} \quad (77)$$

$$TH_{Hmax} = 180^\circ \triangleq 90 \cdot \frac{360^\circ}{180} \quad (78)$$

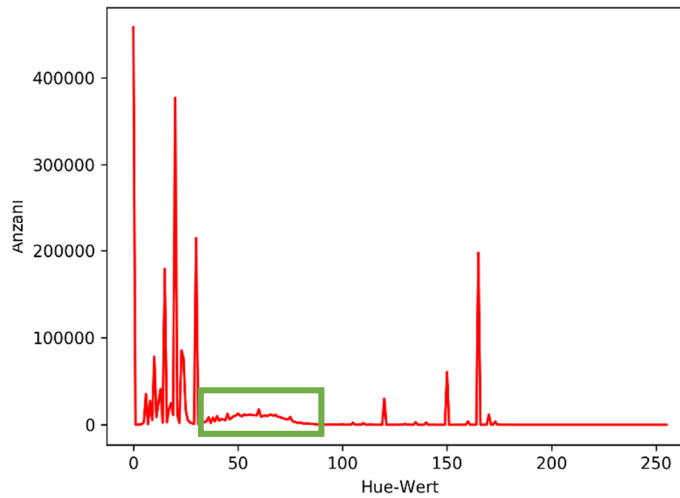


Abbildung 47: Darstellung des Histogramms des Farbwert-Kanals

Wird dieser Bandpassfilter in die Formel (71) eingesetzt, lässt sich die Berechnung vereinfachen zu:

$$H(x,y)' = \begin{cases} \frac{120 + 60 \cdot \left(\frac{\psi_{\text{Blau}}(x,y) - \psi_{\text{Rot}}(x,y)}{255} \right)}{\text{MAX}(x,y) - \text{MIN}(x,y)} & \text{wenn } \text{MAX}(x,y) = \frac{\psi_{\text{Grün}}(x,y)}{255} \\ 0 & \text{sonst} \end{cases} \quad (79)$$

Zudem kann die Formel mit einer binären Ausgabe vom Schwellwert erweitert werden

$$H(x,y)' = \begin{cases} 1 & \text{wenn } \text{MAX}(x,y) = \frac{\psi_{\text{Grün}}(x,y)}{255} \\ 0 & \text{sonst} \end{cases} \quad (80)$$

Das Ergebnis der Verwendung des Bandpasses und einer Binärisierung des Farbtonbildes ist in Abbildung 48 dargestellt. In diesem Bild sind die Pflanzen zu erkennen. Außerdem ist ersichtlich, dass noch einige kleine Fragmente im Bild existieren. Eine Verwendung der Dilatation- und Erosion-Operation ist hier nicht möglich, da die Pflanzengröße in Bezug auf die Fragmentgröße nicht ausreichend hoch ist und durch die Operationen eine zu große Veränderung aufweisen würde.

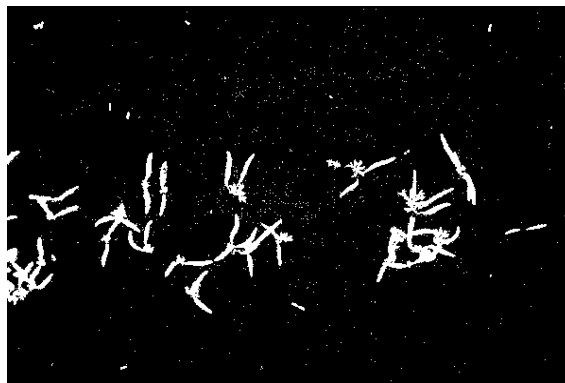


Abbildung 48: Das binärisierte H-Bin-Bild mit dem Bandpassfilter

Der nächste Schritt ist es, das S-Bild zu binärisieren. Dafür wird ein Schwellwert benötigt. Der Schwellwert kann mit Hilfe eines Histogramms bestimmt werden. Das Histogramm für das S-Bild ist in Abbildung 49 dargestellt. Ein Schwellwert muss hinter dem Maximum des Graphen liegen, da diese Häufung den gesamten Hintergrund abbildet. Ein möglicher Schwellwert ist mit einer grünen Linie im Histogramm eingezeichnet. Auf der rechten Seite des Schwellenwertes befinden sich die zu

segmentierenden Pflanzen und einige Teile des Hintergrundes. Als guter Schwellwert THs erwies sich in der Praxis der Wert, wo der Graph auf 16% des Maximums abfiel¹³. Der Schwellwert ist ein Resultat, das verschiedene Einflüsse beinhaltet. Einflüsse können unter anderem Feuchtigkeit und Belichtungszeit sein. Der Schwellwert soll später in der Arbeit noch genauer untersucht werden.

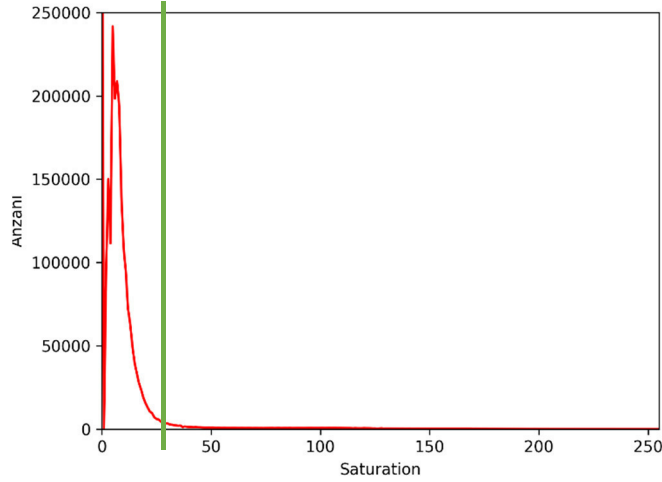


Abbildung 49: Histogramm von dem S-Bild

Der Schwellwert kann als Variable THs auf die Sättigung angewendet werden:

$$S(x,y)' = \begin{cases} S(x,y) & \text{wenn } S(x,y) \geq THs \\ 0 & \text{sonst} \end{cases} \quad (81)$$

Dieser Schwellenwert kann direkt auf die Formel (70) angewendet werden, damit ergibt sich:

$$S(x,y)' = \begin{cases} \frac{MAX(x,y) - MIN(x,y)}{MAX(x,y)} & \text{wenn } \frac{MAX(x,y) - MIN(x,y)}{MAX(x,y)} \geq \frac{THs}{255} \\ 0 & \text{sonst} \end{cases} \quad (82)$$

Daraus folgt durch Umformen:

$$S(x,y)' = \begin{cases} \frac{MAX(x,y) - MIN(x,y)}{MAX(x,y)} & \text{wenn } \frac{MIN(x,y)}{MAX(x,y)} \leq \frac{255 - THs}{255} \\ 0 & \text{sonst} \end{cases} \quad (83)$$

Als binärisierte Schwellwert-Operation folgt daraus:

$$S(x,y)' = \begin{cases} 1 & \text{wenn } \frac{MIN(x,y)}{MAX(x,y)} \leq \frac{255 - THs}{255} \\ 0 & \text{sonst} \end{cases} \quad (84)$$

Das Ergebnis der Formel (86) ergibt durch Verwendung des Schwellenwertes THs das S-BIN Bild aus Abbildung 50. In dem Bild sind einige Fragmente, die keine Pflanzen sind. Durch eine logische „Und“-Verknüpfung mit dem H-BIN Bild aus Abbildung 48 können die Fragmente herausgefiltert werden, da in beiden Bildern die Fragmente als „zufälliges“ Rauschen auftreten und lediglich die Pflanzen identisch sind. Der hier nicht benötigte Value-Kanal kann verwendet werden, um Belichtungsfehler wie zum

¹³ Dieser Wert ist ein in der Praxis auf dem Feld experimentell ermittelter Wert ohne wissenschaftlichen Bezug. Er soll dem Leser lediglich eine Größenordnung zur Veranschaulichung zeigen.

Beispiel Überbelichtung zu korrigieren. Dies ist notwendig wenn stark getrockneter Boden betrachtet wird.

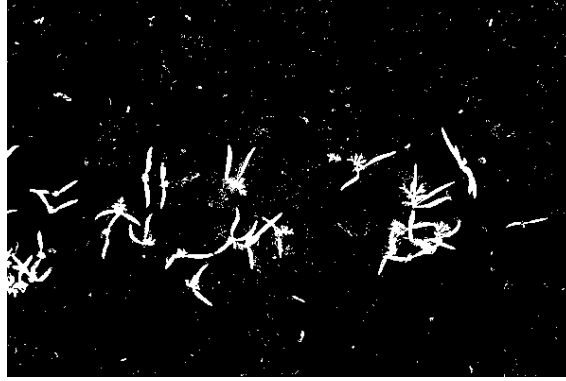


Abbildung 50: S-BIN Bild nach der Verwendung des Schwellwertes

Der Schwellwert der Formel (84) kann nur 1 sein, wenn $V(x, y)' = \psi_{Grün}(x, y)$, da eine logische “Und“-Verknüpfung mit Formel (80) erfolgen soll. Damit ergibt sich:

$$S(x, y)' = \begin{cases} 1 & \text{wenn } \frac{MIN(x, y)}{\frac{\psi_{Grün}(x, y)}{255}} \leq \frac{255 - THs}{255} \\ 0 & \text{sonst} \end{cases} \quad (85)$$

Mit $MIN(x, y) := \min\left(\frac{\psi_{Red}(x, y)}{255}, \frac{\psi_{Green}(x, y)}{255}, \frac{\psi_{Blue}(x, y)}{255}\right)$ folgt:

$$S(x, y)' = \begin{cases} 1 & \text{wenn } \frac{MIN(\psi_{Rot}(x, y), \psi_{Grün}(x, y), \psi_{Blau}(x, y))}{\psi_{Grün}(x, y)} \leq \frac{255 - THs}{255} \\ 0 & \text{sonst} \end{cases} \quad (86)$$

Wird die berechnete Maske durch Formel (86) zum Ausschneiden der Pflanzen verwendet, entsteht die Abbildung 51. Durch die nicht benötigte direkte Transformation in den HSV-Raum, wie sie in dem Flussdiagramm aus Abbildung 45 dargestellt ist, kann Rechenzeit eingespart werden.

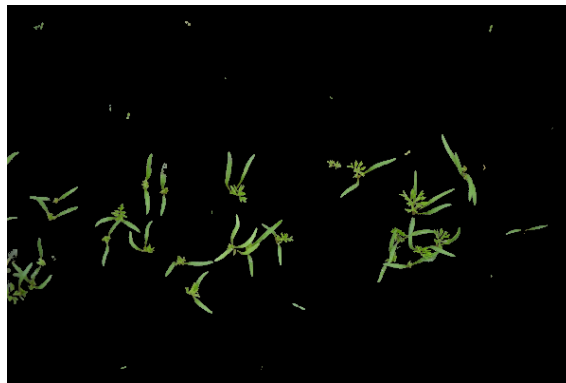


Abbildung 51: Vom Hintergrund segmentierte Pflanzen aus dem RGB Bild

Auch dieser Algorithmus soll mit Hilfe der zur Verfügung stehenden Datensätze untersucht werden. Die Ergebnisse der globalen Genauigkeitsanalyse sind oben und die der lokalen Genauigkeitsanalyse unten in der Abbildung 52 dargestellt. Anders als das RGB-Segmentierungsverfahren, verwendet das HSV-Segmentierungsverfahren einen Schwellwert, der am Ende der Skala des 8-Bit-Farbwertes liegt. Weiter zeigen die Graphen, dass Genauigkeitswerte in der Nähe¹⁴ von 100% erreicht werden. Diese Werte sind

¹⁴ Die Genauigkeit von exakt 1 entsteht durch Rundung des Ergebnisses.

skaliert, so dass die Menge von Pflanzenpixeln und Hintergrundpixeln identisch sind. Die Ergebnisse um 1 ergeben sich auch wie bereits beschrieben durch die Verteilung der Testbilder im Bild.

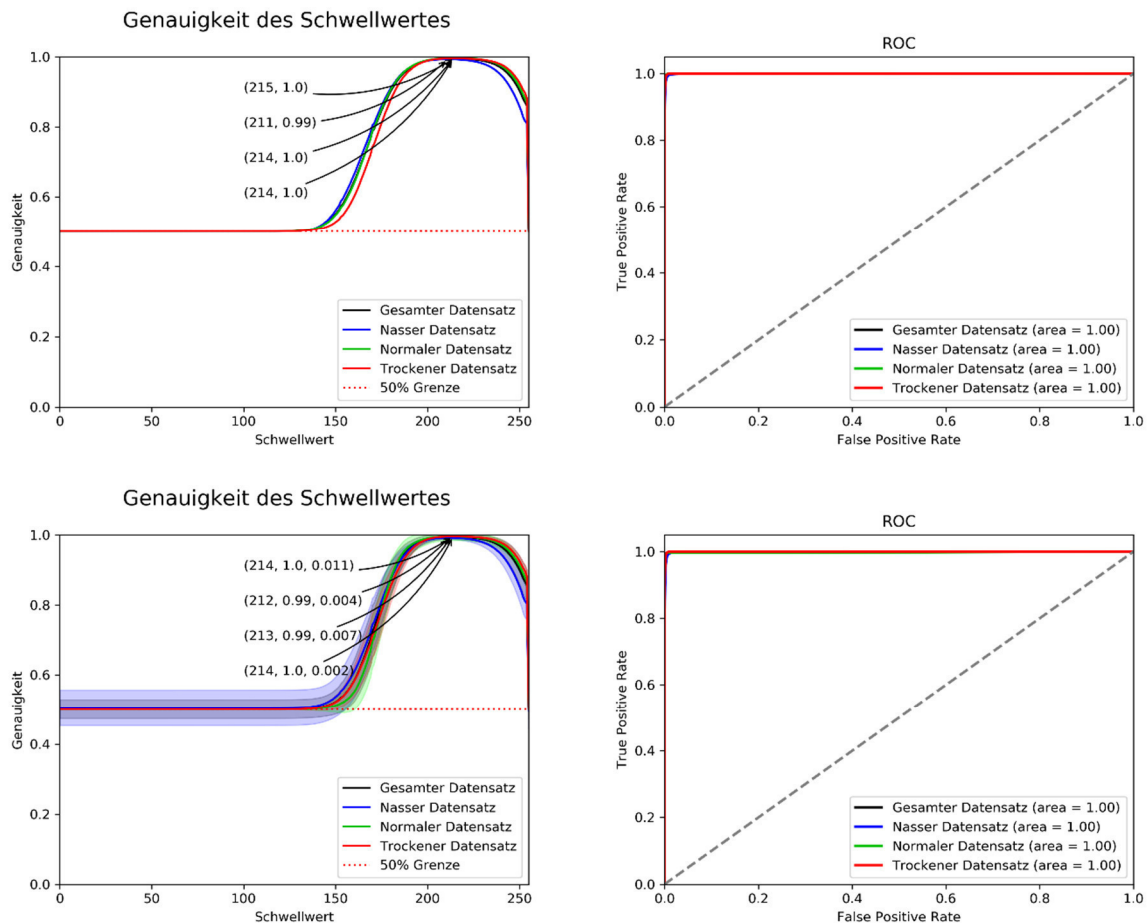


Abbildung 52: Genauigkeitsverlauf des HSV-Segmentierungsverfahrens; links: Analyse ohne Betrachtung der einzelnen Bilder; rechts: Analyse über einzelne Bildelemente

Dies ist in Zahlen ausgedrückt in Tabelle 10 dargestellt, die die Werte der Konfusionsmatrix zeigt bei einem Schwellwert von 213, der für den Datensatz, der alle Bilder beinhaltet, also der Messung ohne Einzelbildbetrachtung, die beste Genauigkeit erzielt. Es ist zu sehen, dass die Summe der richtig klassifizierten Pixel mit circa 385 Millionen der Summe der falschklassifizierten Pixel mit circa 2 Millionen weit überlegen ist. Das Verhältnis von 2 zu 385 ergibt 5 Promille. Für die Werte der Einzelbildbetrachtung ergibt sich ein ähnliches Bild. Auch wenn hier die nicht skalierten Roh-Messgrößen verwendet worden sind, bleibt die Größenordnung auch beim Skalieren erhalten.

Tabelle 10: Konfusions-Matrix des HSV-Segmentierungsverfahrens über den Datensatz, der alle Bilder beinhaltet und ohne Einzelbildbetrachtung (ohne Skalierung)

Konfusions-Matrix für den Schwellwert 213		Ergebnis des Klassifikators	
		Positiv	Negativ
Gehört es wirklich zu der Klasse?	Positiv	14.327.676 (True-Positiv)	60.792 (False-Negativ)
	Negativ	1.950.868 (False-Positiv)	370.237.826 (True-Negativ)

Die Fehler folgen selber keinem bestimmten Muster, sodass nicht angegeben werden kann, wo der Segmentierer bei der fehlerhaften Klassifikation eine besondere Häufung aufweist. Die Precision-, Recall- und F1-Score-Werte liegen, wie die Werte der Konfusionsmatrix zeigen, in der Nähe von 1, wie die Tabelle 11 für die Analyse ohne Einzelbildbetrachtung zeigt und

Tabelle 12 für die Analyse mit Einzelbildbetrachtung.

Tabelle 11: Statistische Auswertung des HSV-Segmentierungsverfahrens an der Position des besten Genauigkeitswertes der Analyse ohne einzelne Bildelemente

Datensatz	bester Schwellwert	Genauigkeit	Precision	Recall	F1-Score
Alle	214	1,00	0,99	1,00	1,00
Nass	211	0,99	0,99	0,99	0,99
Normal	214	1,00	0,99	1,00	1,00
Trocken	215	1,00	1,00	1,00	1,00

Tabelle 12: Statistische Auswertung des HSV-Segmentierungsverfahrens an der Position des besten Genauigkeitswertes der Analyse über einzelne Bildelemente

Datensatz	bester Schwellwert	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	213	0,99 (0,007)	0,99 (0,055)	1,00 (0,013)	0,99 (0,055)
nass	212	0,99 (0,004)	0,98 (0,102)	0,99 (0,006)	0,98 (0,102)
normal	214	1,00 (0,011)	0,99 (0,004)	1,00 (0,023)	1,00 (0,013)
trocken	214	1,00 (0,002)	0,99 (0,003)	1,00 (0,001)	1,00 (0,002)

Die Messung erfolgte, wie bei den anderen Verfahren auch, durch Verwendung des Schwellwerts, in welchem der Genauigkeitswert sein Maximum über alle Bilder des Datensatzes einnimmt. Die Skalierung wurde auch hier über alle Bilder berechnet und anschließend separat auf jedes Bild des Datensatzes einzeln angewendet. Zwar sind bei einer Genauigkeit von 99,5% bei der signifikanten Messung 26% der Bilder nicht innerhalb der Grenzen und 24% der Bilder für die hochsignifikanten Messungen außerhalb der Grenzen. Doch bereits wenn der Genauigkeitswert auf 98% herabgesenkt wird, fehlen lediglich nur noch 4 Bilder bei der signifikanten Messung und 2 Bilder bei der hochsignifikanten Messung. Ab einem Genauigkeitswert von circa 96,5% sind alle Bilder für beide Messungen statistisch verwertbar. Wird die Genauigkeit ohne Skalierung untersucht, ergibt sich folgender Verlauf des Precisionwertes.

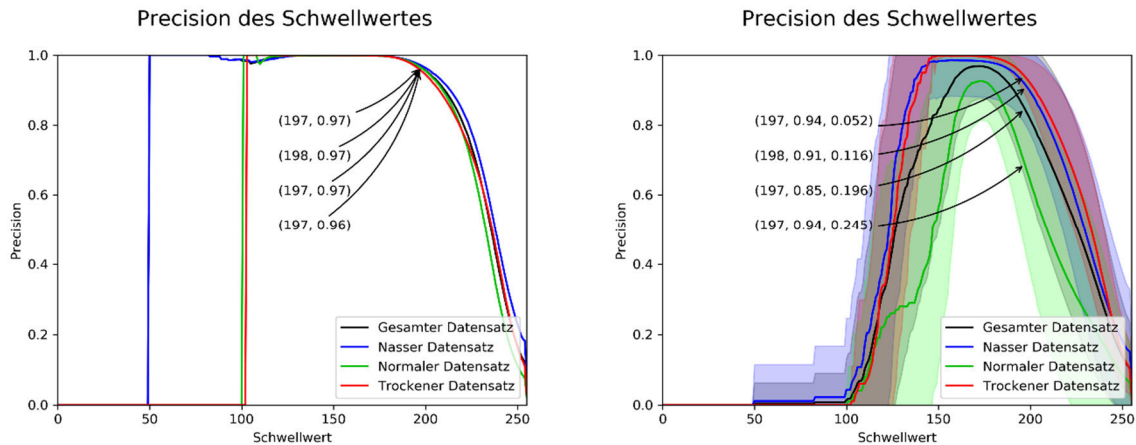


Abbildung 53: Verlauf des Precision-Wertes des HSV-Segmentierungsverfahrens; links: Analyse ohne Betrachtung der einzelnen Bilder; rechts: Analyse über einzelne Bildelemente.

Der plötzliche Sprung des Precision-Wertes bei der Messung, die keine Analyse der einzelnen Bildelemente vornimmt (linkes Bild in Abbildung 53), entsteht durch die Berechnung des Precision-Wertes. Die Formel bildet ein Verhältnis aus den Größen der True-Positiv- und der False-Positiv-Werte. Wenn jetzt der True-Positiv-Wert auf 1 steigt und der False-Positiv-Wert bei 0 bleibt, ergibt sich die Precision von 1. Dadurch, dass dieser Graph jedoch über alle Bilder zusammen berechnet wird, muss dieser Fall nur in einem Bild bei dem Schwellwert vorkommen, damit der Sprung zustande kommt. Im rechten Bild, wo der Precision-Wert über die einzelnen Bilder gemittelt wird, hat dieser Fall nicht die große Auswirkung wie im linken Graphen. Die Graphen der Precision ohne Skalierung zeigen, dass das HSV-Segmentierungsverfahren auch unter realen Bedingungen bessere Werte erreicht als das NDVI-Verfahren und das RGB-Segmentierungsverfahren. Den schlechtesten Wert von 85% mit einer Standardabweichung von 19,6 Prozentpunkten zeigt hier die Messung über einzelne Bilder in dem Datensatz, der alle Bilder beinhaltet.

Wird für jedes Bild des Datensatzes, der alle Bilder beinhaltet, der optimale Schwellwert für die maximale Genauigkeit verwendet und die Verteilung der Schwellwerte in einem Histogramm veranschaulicht, ergibt sich die Darstellung in Abbildung 54. Wie die Abbildung zeigt, besitzt die Verteilung der Schwellwerte keine große Streuung. Dies zeigt, dass das HSV-Segmentierungsverfahren mit einem Schwellwert robust auf verschiedene Gegebenheiten der Bilder reagieren kann.

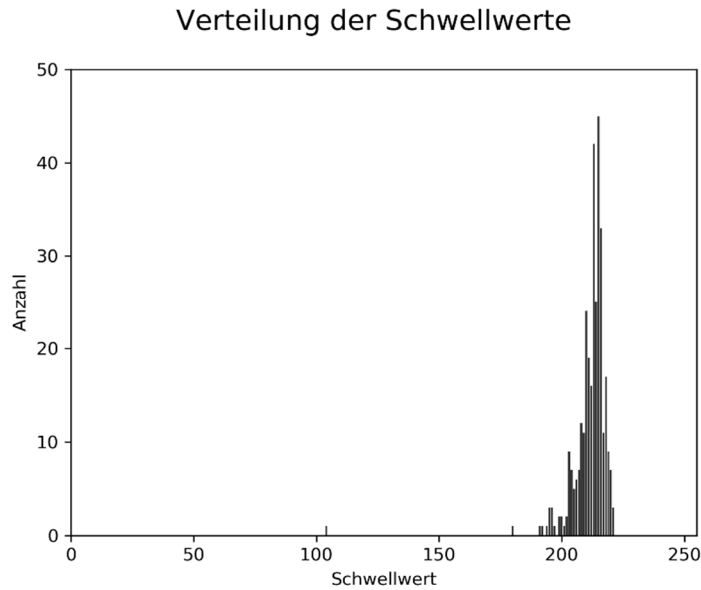


Abbildung 54: Histogramm der Schwellwertverteilung für den optimalen Schwellwert für jedes Bild im HSV-Segmentierungsverfahren

Wenn für jedes Bild der Schwellwert verwendet wird, der die beste Genauigkeit ermöglicht, ergeben sich die Werte aus Tabelle 13. Es ist nicht überraschend, dass diese Werte dasselbe Niveau wie die vorherigen Ergebnisse des HSV-Segmentierungsverfahrens besitzen. Die Werte sollen hier jedoch nicht wie in den Abbildungen mit zwei Nachkommastellen angegeben werden, sondern mit einer weiteren.

Tabelle 13: Statistische Auswertung des HSV-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild

Datensatz	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	0,995 (0,0069)	0,992 (0,0551)	0,996 (0,0128)	0,992 (0,0555)
nass	0,993 (0,0039)	0,981 (0,1023)	0,993 (0,0051)	0,982 (0,1024)
normal	0,996 (0,0113)	0,997 (0,0025)	0,995 (0,0224)	0,996 (0,0125)
trocken	0,996 (0,0017)	0,995 (0,0023)	0,998 (0,0012)	0,996 (0,0016)

Ein noch besseres Ergebnis als die vorhergegangenen Segmentierungsalgorithmen zeigt der Segmentierer, der in dem HSV-Farbraum arbeitet. Zudem benötigt er wie das RGB-Verfahren kein kompliziertes Hardware-Setup, sondern verwendet eine herkömmliche RGB-Kamera. Durch die niedrigen Zahlen der False-Positiv- und False-Negative-Werte, muss ein nachfolgender Klassifikator weniger Hintergrundpixel verarbeiten und es besteht nicht die Gefahr, dass zu viele Pflanzenpixel durch die Maske des Segmentierers entfernt wurden.

3.6.7 Multilayer Perceptron pixelweise

Wird zum Beispiel das linke Bild aus Abbildung 55 in einem dreidimensionalen Raum dargestellt, indem der Raum mit den Farben (Rot, Grün und Blau) als Achsen aufgespannt wird, ergibt sich die dargestellte Punktwolke. In der Punktwolke sind die grünen Punkte die Farbwerte der Pflanzen und die roten Punkte bilden den Hintergrund ab. Der Hintergrund kann bei genauer Betrachtung als Zylinder interpretiert werden, welcher alle drei Farbparameter wachsend durchläuft. Er bildet daher verschiedene Facetten von grau ab. Ziel sollte es daher sein, die Pflanzen, die weiter verstreut abseits des Zylinders liegen, zu separieren.

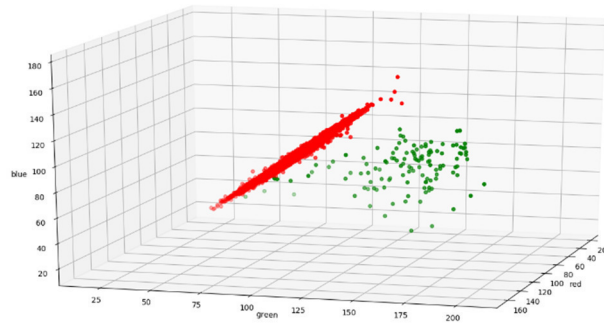


Abbildung 55: Darstellung eines RGB Bildes (links) in einer 3 Dimensionale Punktwolke (rechts)

Die Trennlinien zwischen den Pflanzen und dem Hintergrund können in einem zweidimensionalen Raum bestimmt werden. Dazu dient eine Transformation und eine Projektion der Punktwolke. Der erste Schritt besteht darin, den Zylinder, in dem die Hintergrundpunkte liegen, so zu drehen, dass sie parallel zu einer der drei Achsen liegen. Für die parallele Ausrichtung müssen zwei Drehwinkel bestimmt werden. Die Winkel können über Regressionsgeraden bestimmt werden.

Der erste Drehwinkel kann durch eine Projektion der Punktwolke innerhalb der rot-grünen Ebene (siehe Abbildung 56) und der in ihr bestimmten Steigung der Regressionsgeraden bestimmt werden. In der Abbildung ist die Regressionsgerade blau eingezeichnet.

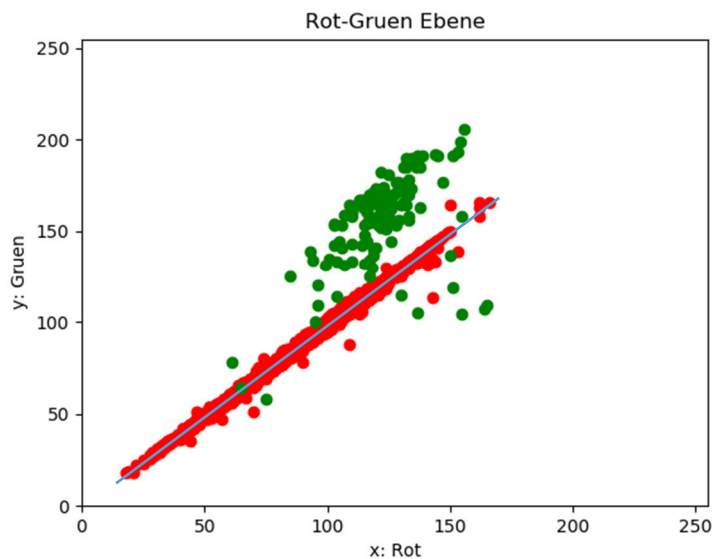


Abbildung 56: Projektion der Punktwolke auf die rot-grüne Ebene

Zur Bestimmung der Steigung kann die Gleichung

$$m = \frac{\sum_{i=1}^N (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^N (x_i - \bar{x})^2} \quad (87)$$

verwendet werden [Bartsch, 2004]. Der Parameter x ist in diesem Fall der Rot-Wert und der Parameter y ist der Grün-Wert und die Mittelwerte ergeben sich aus

$$\bar{x} := \frac{1}{N} \cdot \sum_{i=1}^N x_i \text{ und analog } \bar{y} := \frac{1}{N} \cdot \sum_{i=1}^N y_i. \quad (88)$$

Aus der Steigung der Formel (87) kann über den trigonometrischen Zusammenhang des Tangenssatzes [Bartsch, 2004]

$$\tan(\alpha) = \frac{\text{Gegenkathete}}{\text{Ankathete}} \Rightarrow \alpha = \arctan\left(\frac{\text{Gegenkathete}}{\text{Ankathete}}\right) \quad (89)$$

mit $m = \frac{\text{Gegenkathete}}{\text{Ankathete}}$ geschrieben werden:

$$\alpha = \arctan\left(\frac{\sum_{i=1}^N (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^N (x_i - \bar{x})^2}\right). \quad (90)$$

Die Drehung der Punktwolke erfolgt anschließend über die Z-Achse (blaue-Achse) in der 3-dimensionalen Punktwolke mit der Drehmatrix [Bartsch, 2004]:

$$R_z(\alpha_1) = \begin{pmatrix} \cos(\alpha_1) & -\sin(\alpha_1) & 0 \\ \sin(\alpha_1) & \cos(\alpha_1) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (91)$$

Nach der Drehung der Punktwolke erfolgt eine Projektion auf die blau-grüne-Ebene (ZY-Ebene), wie sie in Abbildung 57 dargestellt ist. Es wird wieder die Steigung der blau eingezeichneten Regressionsgeraden mit der Formel (90) bestimmt.

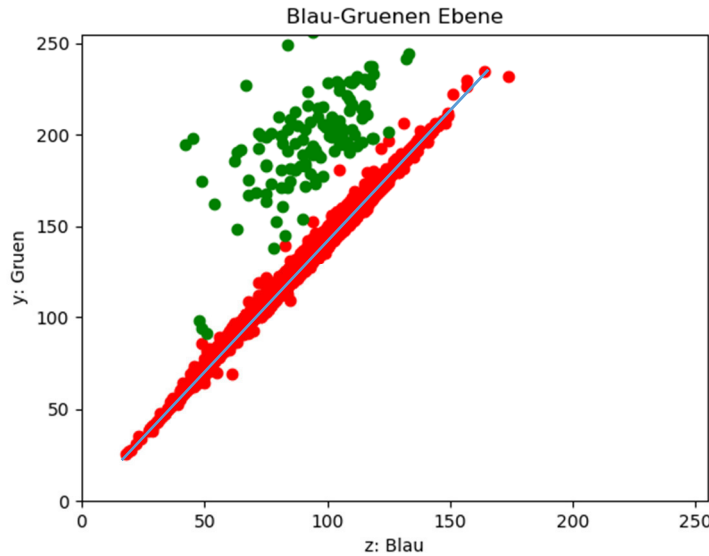


Abbildung 57: Projektion der Punktwolke auf die blau-grüne Ebene

Diesmal erfolgt mit dem berechneten Winkel eine Drehung über die X-Achse (rote Achse) im dreidimensionalen Raum mit der Drehmatrix [Bartsch, 2004]:

$$R_x(\alpha_2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_2) & -\sin(\alpha_2) \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) \end{pmatrix}. \quad (92)$$

Mit dieser Drehung liegt die Regressionsgerade parallel zur blauen Achse (Z-Achse). Das neue Koordinatensystem der Projektion wird dann über die Achsen X und Z aufgespannt. Diese beiden Achsen werden nun u und v genannt, da diese sich von dem ursprünglichen Parameter durch die Drehung unterscheiden. Es wird entlang der Z-Achse in die Ebene geblickt. Zur Vereinfachung der

Rechenschritte sollen diese zusammengefasst werden. Wird die erste Drehmatrix mit dem RGB-Farbvektor $\vec{\chi} = (R \ G \ B)^T$ multipliziert, ergibt sich der Transformationsvektor $\vec{t}_1$ mit

$$\vec{t}_1 = R_z(\alpha_1) \cdot \vec{\chi} = \begin{pmatrix} \cos(\alpha_1) & -\sin(\alpha_1) & 0 \\ \sin(\alpha_1) & \cos(\alpha_1) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} \cos(\alpha_1) \cdot R - \sin(\alpha_1) \cdot G \\ \sin(\alpha_1) \cdot R + \cos(\alpha_1) \cdot G \\ B \end{pmatrix}. \quad (93)$$

Wird nun der Transformationsvektor $\vec{t}_1$ auf die zweite Drehmatrix wie folgt angewendet

$$\vec{t}_2 = R_x(\alpha_2) \cdot \vec{t}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_2) & -\sin(\alpha_2) \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) \end{pmatrix} \cdot \begin{pmatrix} \cos(\alpha_1) \cdot R - \sin(\alpha_1) \cdot G \\ \sin(\alpha_1) \cdot R + \cos(\alpha_1) \cdot G \\ B \end{pmatrix} \quad (94)$$

entsteht der Transformationsvektor

$$\vec{t}_2 = \begin{pmatrix} \cos(\alpha_1) \cdot R - \sin(\alpha_1) \cdot G \\ \cos(\alpha_2) \cdot [\sin(\alpha_1) \cdot R + \cos(\alpha_1) \cdot G] - \sin(\alpha_2) \cdot B \\ \sin(\alpha_2) \cdot [\sin(\alpha_1) \cdot R + \cos(\alpha_1) \cdot G] + \cos(\alpha_2) \cdot B \end{pmatrix}. \quad (95)$$

Das neue Koordinatensystem setzt sich jedoch nur aus den ersten beiden Parametern zusammen, weshalb der dritte Parameter ignoriert werden kann. Es folgt der Transformationsvektor

$$\vec{t} = \begin{pmatrix} \cos(\alpha_1) \cdot R - \sin(\alpha_1) \cdot G \\ \cos(\alpha_2) \cdot [\sin(\alpha_1) \cdot R + \cos(\alpha_1) \cdot G] - \sin(\alpha_2) \cdot B \end{pmatrix}. \quad (96)$$

Durch das Ausmultiplizieren der Klammern folgt:

$$\vec{t} = \begin{pmatrix} \cos(\alpha_1) \cdot R - \sin(\alpha_1) \cdot G \\ \cos(\alpha_2) \cdot \sin(\alpha_1) \cdot R + \cos(\alpha_2) \cdot \cos(\alpha_1) \cdot G - \sin(\alpha_2) \cdot B \end{pmatrix}. \quad (97)$$

Dies bedeutet, dass der erste Parameter den RGB-Wert in den u-Wert und der zweite Parameter den RGB-Wert in den v-Wert transformiert. Dies kann wie ein neuronales Netz in Abbildung 58 beschrieben werden. Jeder Parameter wird durch ein Neuron dargestellt. Die Gewichte können einfach aus dem Transformationsvektor $\vec{t}$ abgelesen werden.

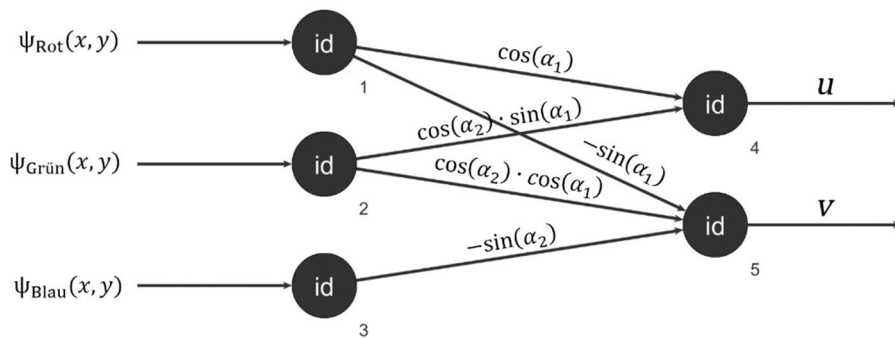


Abbildung 58: neuronales Netz zum Erstellen der Transformation in den u-v-Raum

Das Netzwerk besteht aus zwei Schichten: der Eingabeschicht, die die RGB-Werte in das Netzwerk leitet und der Verarbeitungssicht, bestehend aus zwei Neuronen. Die Verarbeitungssicht dient ebenfalls als Ausgabeschicht. Alle Neuronen verwenden als Aktivierungsfunktion die Identität. Die Abbildung 59 zeigt die projizierte Punktwolke. Auch wenn es sich idealisiert um einen Zylinder handelt, ist ein radialer Ansatz hier nicht möglich, wie die Abbildung zeigt.

Die Pflanzenpixel befinden sich im oberen linken Segment, das durch die Trennlinien beschränkt wird. Die meisten Hintergrundpixel befinden sich in den Segmenten unter der orangenen Trennlinie. In dem Segment oben rechts liegen in dem hier untersuchten Bild überwiegend Stöcke, Steine und Pflanzenstiele.

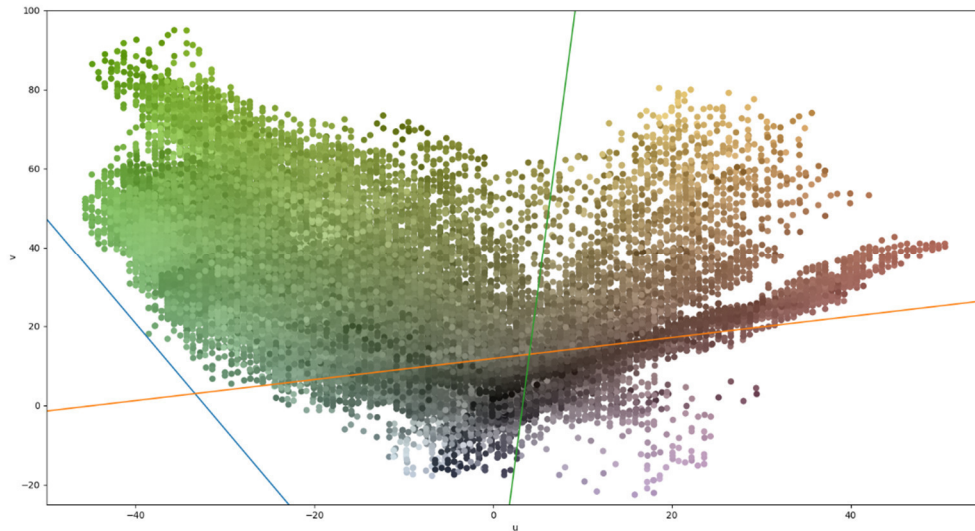


Abbildung 59: Darstellung der Projektion der kompletten Punktwolke auf den neuen zweidimensionalen Raum der u - v -Ebene

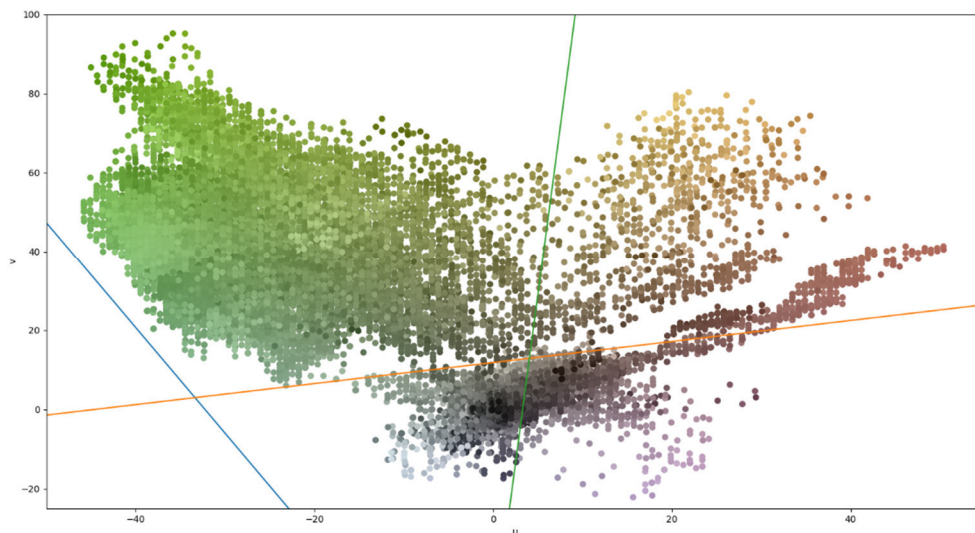


Abbildung 60: Darstellung der Projektion der reduzierten Punktwolke auf den neuen zweidimensionalen Raum der u - v -Ebene

Werden alle Pixel in der u - v -Ebene abgebildet und mit ihren jeweiligen RGB-Werten dargestellt, entsteht die Abbildung 59. In der Abbildung befinden sich die Pflanzen-Pixel oben links (2. Quadrant) im Koordinatensystem. Im ersten Quadranten befinden sich überwiegend tote Pflanzenreste und Steine. Der Erdboden befindet sich überwiegend in der Nähe des Koordinatenursprungs. Wie an der Farbdarstellung zu erkennen ist, gibt es Ähnlichkeiten zum HSL-Farbraum, jedoch sind die Werte unterschiedlich. Zur Bestimmung geeigneter Trennlinien bietet es sich an, eine reduzierte Darstellung

wie in Abbildung 60 zu wählen. Die reduzierte Darstellung wird erzeugt, indem die Kanten (± 2 Pixel) nicht dargestellt werden. Die Kanten und ihre direkte Nachbarschaft stellen eine unsichere Umgebung dar. Dies liegt an der Unschärfe der Bilder, so dass der Übergang zwischen dem Hintergrund und der Pflanze nicht exakt zugewiesen werden kann. Durch das Entfernen dieser unsicheren Pixel bildet sich ein Abstand zwischen den Clustern, in dem die Trennlinien gesetzt werden können. Für die Segmentierung werden hier drei dieser Trennlinien verwendet. Durch die blaue Trennlinie wird der Bereich definiert, in dem zu blaue Pixel nicht berücksichtigt werden. Es gibt in den dargestellten Abbildungen zwar keine Pixel, die rausgefiltert werden müssen, jedoch kommen auf den Feldern auch Muscheln mit diesen Farbwerten vor. Da die Pflanzen hier eine starke Kante im u-v-Raum bilden, soll diese Trennlinie unvorhersehbare Hintergrundanteile herausfiltern. Mit der orangefarbenen Trennlinie werden die Pflanzen vom Erdboden getrennt und mit der grünen Trennlinie werden die abgestorbenen Pflanzen abgetrennt. Zur Umsetzung der Trennlinien werden die Geradengleichungen benötigt. Die erste (blaue) Trennlinie läuft durch die Punkte $P_1(-45|34)$ und $P_2(-39|18)$. Damit ergibt sich die Geradengleichung in Parameterdarstellung zu

$$-\frac{1}{86} \cdot x_1 - \frac{4}{129} \cdot x_0 = 1. \quad (98)$$

Aus der Parameterdarstellung ergeben sich die Koeffizienten des Schwellwertelementes über die Gleichung

$$\sum_{i=0}^N \omega_i \cdot x_i(t) = \theta \quad (99)$$

zu den Gewichten und dem Schwellwert des Neurons:

$$\begin{aligned} \omega_{46} &= -\frac{4}{129} \\ \omega_{56} &= -\frac{1}{86} \\ \theta_6 &= 1. \end{aligned} \quad (100)$$

Analog kann die Parameterdarstellung der zweiten Trennlinie (orangene Trennlinie) bestimmt werden. Die Trennlinie läuft durch die Punkte $P_3(-22|6)$ und $P_4(27|19)$. Dies ergibt

$$\frac{49}{8} \cdot x_1 - \frac{13}{8} \cdot x_0 = 1. \quad (101)$$

Daraus folgen die Parameter

$$\begin{aligned} \omega_{47} &= -\frac{13}{8} \\ \omega_{57} &= \frac{49}{8} \\ \theta_7 &= 1. \end{aligned} \quad (102)$$

Die dritte Trennlinie (grüne Trennlinie) schneidet die Punkte $P_5(7|63)$ und $P_6(4|12)$. Dies ergibt die Parameterdarstellung

$$-\frac{1}{56} \cdot x_1 + \frac{17}{56} \cdot x_0 = 1 \quad (103)$$

und

$$\begin{aligned} \omega_{48} &= \frac{17}{56} \\ \omega_{58} &= -\frac{1}{56} \\ \theta_8 &= 1. \end{aligned} \quad (104)$$

Sollte ein RGB-Farbwert oberhalb aller Trennlinien sein, handelt es sich um einen Pflanzenpixel. Daher können alle drei Schwellwertelemente mit einem logischen „und“ verknüpft werden. Wird das Transformationsnetzwerk aus Abbildung 58 mit diesen Schwellenwertelementen erweitert, ergibt sich das neuronale Netz zur Segmentierung in der nachfolgenden Abbildung.

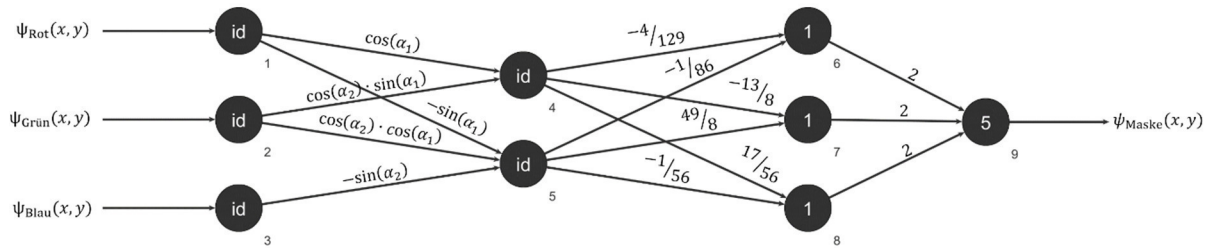


Abbildung 61: Künstliches neuronales Netz zur pixelweisen Segmentierung

Die Eingangsneuronen leiten die RGB-Werte in das Netzwerk, anschließend erfolgt die Transformation in den u-v-Raum mit den folgenden Verbindungen zu den Neuronen 4 und 5. Danach erfolgt über die Neuronen 6, 7 und 8 die Trennlinienberechnung (blau, orange und grün). Das Neuron 9 bildet mit den Gewichten eine logische „und“-Verknüpfung. Das logische „und“ ist ein bekanntes Segment aus Kapitel 2.2.2. Die Neuronen 6 bis 9 verwenden die Schwellwertfunktion als Aktivierungsfunktion, wobei der Schwellwert selbst in den Neuronen dargestellt ist. Das Netzwerk erzeugt aus den Eingangswerten eine 1 sofern es sich um eine Pflanze handelt und eine 0, wenn es sich um Hintergrund handelt. Eine mögliche Erweiterung des Netzwerkes bestünde darin, die Gewichte so zu modifizieren, dass sie mit einem Shift-Operator abgearbeitet werden können. Durch diese Erweiterung wäre ein Geschwindigkeitsvorteil gegenüber der zeitaufwendigeren Multiplikation gegeben.

Für die Untersuchung dieses Netzes können nicht alle Permutationen von Trenngeraden untersucht werden. Daher soll im Test lediglich eine Kombination für jedes Bild verwendet werden. Es handelt sich bei dieser Kombination nicht zwangsweise um das Optimum. Zur Einstellung der Trenngeraden für das jeweilige Bild wurde das Ground-truth-Bild verwendet, um die Trenngerade auf Genauigkeit hin zu optimieren. Es existiert derzeit kein Algorithmus, welcher die Trenngeraden automatisch optimal berechnen kann. Für eine reale Nutzung könnten vor der Messung die Trenngeraden manuell festgelegt werden und für das Feld verwendet werden, solange es keine großen Veränderungen über dem Feld gibt. Für einen automatischen echtzeitfähigen Algorithmus, das heißt für jedes separat aufgenommene Bild, könnten verschiedene Aspekte verwendet werden, die aus der Familie der Cluster-Verfahren stammen. Durch die Betrachtung eines RGB-Bildes der JAI-Kamera nach der Koordinatentransformation (Abbildung 62) wird ein Problem der Kamera sichtbar.

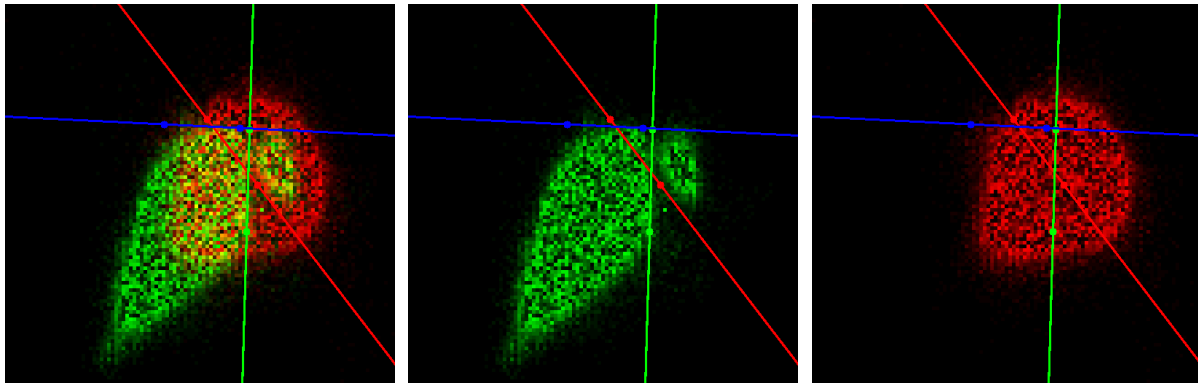


Abbildung 62: Darstellung der verschiedenen Pixel nach der Koordinatentransformation. Die Geraden zeigen die Trenngeraden. Links: vereinigte Darstellung, grün sind Pflanzenpixel, rot sind Hintergrundpixel; Mitte nur Pflanzenpixel; Rechts nur Hintergrundpixel.

Anders als zum Beispiel bei der Nikon Kamera, ist der Farbwertabstand bei der JAI-Kamera der drei Parameter Rot, Grün und Blau nicht so groß. Dies bedeutet, dass ein Pflanzenpixel der JAI-Kamera nur einen geringen Unterschied zwischen dem roten, grünen und blauen Farbkanal besitzt. Dadurch ist dieser Farbwert nach der Koordinatentransformation jedoch in der Nähe oder im selben Bereich wie die Hintergrundpixel, die sich durch ihre Genauigkeit auszeichnen. Der Algorithmus, der die Trenngeraden anhand dieser Bilder einstellt, setzt die blaue und die grüne Trennlinie. Anschließend verschiebt er die rote Trennlinie und übernimmt die Einstellung, welche die beste Genauigkeit hervorbrachte. Wie bereits erwähnt, muss dies nicht die optimalen Parameter hervorbringen. Im grünen Bereich rechts von der grünen Trennlinie sind in der Regel Pflanzenpixel, die direkt an der Kante zu den Hintergrundpixeln sitzen. Diese Pixel werden für die Statistik, wie weiter oben beschrieben, nicht verwendet. Durch den separaten Bereich, die diese Pixel innerhalb der Hintergrundpixel abdecken, könnte davon ausgegangen werden, dass dies eventuell wirklich Hintergrundpixel sind und keine Pflanzenpixel.

Für die Untersuchung über die Einzelbilder (lokale Genauigkeitsanalyse) mit einer Skalierung zwischen Pflanzen und Hintergrund ergibt sich die nachfolgende Tabelle¹⁵:

Tabelle 14: Statistische Auswertung des KNN-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild mit Skalierung

Datensatz	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	0,97 (0,022)	0,98 (0,059)	0,96 (0,039)	0,97 (0,061)
nass	0,97 (0,022)	0,96 (0,102)	0,97 (0,032)	0,96 (0,103)
normal	0,97 (0,021)	0,99 (0,010)	0,95 (0,042)	0,97 (0,023)
trocken	0,97 (0,022)	0,98 (0,013)	0,95 (0,042)	0,96 (0,023)

¹⁵ Eine Messung über alle Pixel aller Bilder soll hier nicht durchgeführt werden, da dieselben Parameter für alle Bilder eines Datensatzes nicht verwendet werden können. Da jedoch keine gemeinsamen Parameter vorliegen, kann auch keine Signifikanzanalyse wie bei den anderen Verfahren durchgeführt werden.

Für die Auswertung, in der alle Bilder zusammengefasst werden (globale Genauigkeitsanalyse) folgt die Tabelle:

Tabelle 15: Statistische Auswertung des KNN-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild und der Auswertung über alle Bilder mit Skalierung

Datensatz	Genauigkeit	Precision	Recall	F1-Score
alle	0,97	0,98	0,97	0,97
nass	0,97	0,98	0,97	0,97
normal	0,98	0,99	0,96	0,98
trocken	0,97	0,98	0,96	0,97

Beide Auswertungen zeigen, dass die Genauigkeit, Precision sowie der Recall im selben Bereich liegen wie bei vorhergegangenen Verfahren zu Segmentierung. Auch hier gibt es zwischen den einzelnen Datensätzen keine nennenswerten Unterschiede und die durchschnittliche Genauigkeit über den Datensatz, der alle Bilder beinhaltet, liegt für beide Messmethoden bei 97%. Die Standardabweichung liegt dabei etwas über zwei Prozentpunkten. Wird die Messung ohne die Verwendung einer Skalierung wiederholt, ergeben sich folgende Messwerte:

Tabelle 16: Statistische Auswertung des KNN-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild ohne Skalierung

Datensatz	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	0,98 (0,015)	0,48 (0,288)	0,96 (0,039)	0,59 (0,029)
nass	0,98 (0,019)	0,59 (0,263)	0,97 (0,032)	0,70 (0,232)
normal	0,99 (0,011)	0,32 (0,296)	0,95 (0,042)	0,41 (0,313)
trocken	0,98 (0,012)	0,55 (0,227)	0,95 (0,042)	0,66 (0,209)

Wenn für die Auswertung nicht einzelne Bilder betrachtet werden, ergeben sich die Ergebnisse aus Tabelle 17.

Tabelle 17: Statistische Auswertung des KNN-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild und der Auswertung über alle Bilder ohne Skalierung

Datensatz	Genauigkeit	Precision	Recall	F1-Score
alle	0,98	0,70	0,97	0,81
nass	0,98	0,77	0,97	0,86
normal	0,99	0,65	0,96	0,78
trocken	0,98	0,59	0,96	0,73

Die Daten wurden ohne Skalierung aufgenommen. Wie zu sehen ist, ist die Genauigkeit zur vorherigen Messung in etwa gleichgeblieben. Jedoch ist der Precision-Wert eingebrochen. Dies liegt an der nicht zu ignorierenden Zunahme der False-Positiv-Werte, die durch die hier fehlende Skalierung auftritt, da auf einem Bild etwa 93% aller Pixel im Durchschnitt Hintergrund-Pixel sind. Weiter wurde für die Optimierung der Trenngeraden lediglich die Genauigkeit als Optimierungsgröße verwendet und kein weiterer Wert, wie zum Beispiel die Precision. Der Recall-Wert zeigt weiter denselben Wert an, da dieser von der Skalierung unbeeinflusst ist. Jedoch zeigt dieser, dass nahezu (ca. 97%) alle Pflanzenpixel auch als Pflanzenpixel klassifiziert worden sind. Die Precision-Werte zeigen, dass in einer realen Messung auf einen klassifizierten richtigen Pflanzenpixel circa ein falscher Pflanzenpixel kommt. Es sollte hier nicht ausgeschlossen werden, dass durch einen automatischen Algorithmus nicht ein

vernünftiger Zwischenwert erreicht werden kann, in dem die Genauigkeit, die Precision und der Recall-Wert besser werden als hier dargestellt.

3.6.8 Multilayer Perceptron flächenweise

Die bisherigen Segmentierungsverfahren waren Verfahren, die alle als Punktoperatoren arbeiten. Flächenbezogene Verfahren können jedoch vorteilhaft sein, da diese die umliegenden Regionen mit einbeziehen und so nicht nur auf einen einzigen Farbwert triggern. Daher soll im Folgenden die Möglichkeit untersucht werden, ein neuronales Netz wie im Kapitel 2.2.4 in Hinblick auf die Möglichkeit der Segmentierung zu untersuchen. Das Training geschieht dabei mit dem Backpropagation-Verfahren aus Kapitel 2.2.3. Es sollte jedoch nicht davon ausgegangen werden, dass das MLP vor dieser Arbeit noch nicht für die Segmentierung in der Bildverarbeitung verwendet wurde, es dient hier jedoch als Vergleich für die anderen Segmentierungsalgorithmen.

3.6.8.1 Multilayer Perceptron flächenweise Segmentierung im RGB-Farbraum

Um ein mögliches Netz zu finden, wurden verschiedene Permutationen automatisiert geprüft. Das Eingangsbild besitzt dabei eine Größe von 3 x 3 Pixeln im RGB-Format, wobei der mittlere Pixel untersucht werden soll. Die untersuchten Netze verwenden ein bis zwei verdeckte Schichten. Dabei liegt die Anzahl der Neuronen zwischen 1 und 5. Als Ausgangsschicht existieren zwei Neuronen, die jeweils eine Wahrscheinlichkeit für Hintergrund und Pflanze zwischen 0 und 1 ausgeben. Ein Beispielaufbau ist in Abbildung 63 dargestellt.

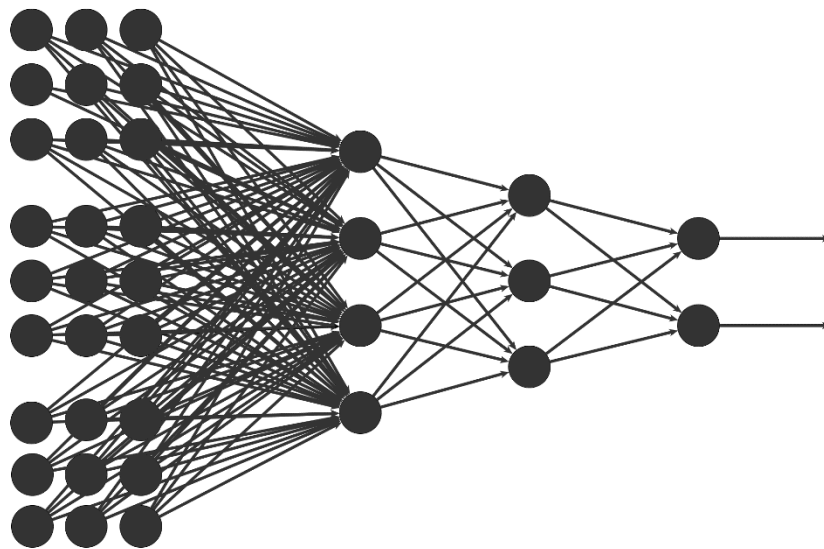


Abbildung 63: MLP Netzaufbau zur Segmentierung mit RGB Eingangsbild, zwei verdeckten Schichten und zwei Neuronen als Ausgang

Die Trainingsbildmenge umfasst eine Datenmenge von 100.000 Bildern mit je einer Auflösung von 3x3 Pixeln im RGB-Format. Die Lernbilder beinhalten dabei nicht die Bilder, welche für die statistische Auswertung verwendet wurden, besitzen jedoch ebenfalls Proben aus nassen, normalen und trockenen Wetterbedingungen. Die Aufteilung der Trainingsbildmenge beinhaltet 50.000 Bilder von Pflanzen und ebenso viele vom Hintergrund, so dass eine Gleichverteilung beider Klassen im Datensatz enthalten ist. Zum Lernen wurde das Batch durch Iterieren von Hintergrund- und Pflanzenpixel aufgebaut. Zum Training selbst wurden 90.000 Bilder mit einer Batchgröße von 64 Bildelementen verwendet. Die weiteren 10.000 Bildelemente wurden für die periodische Evaluation des Trainingserfolgs verwendet. Die Anzahl der Lernepochen ist fest auf 100.000 Epochen gesetzt worden. Durch eine stichprobenartige Analyse verschiedener Netze stellte sich heraus, dass ab circa 30.000 Epochen keine Verbesserung der Gewichte erfolgt. Zur Sicherheit wurden daher 100.000 Epochen zum Lernen angesetzt.

Nach dem Training wurden alle untersuchten Netze auf die unbekannten Datensätze angewendet. Die Darstellung der Genauigkeit des Datensatzes, welcher alle Bilder beinhaltet und keine Analyse über die Einzelbilder vornimmt, ist in Abbildung 64 dargestellt. Dabei wurde erneut die Skalierung zwischen Pflanzenpixeln und Hintergrundpixeln verwendet. Die Darstellung der Ergebnisse erfolgt hier anders als bei den anderen Segmentierungsverfahren, durch die Korrelation zwischen Genauigkeit und der Anzahl der verwendeten Gewichte des jeweiligen Netzwerkes. Außerdem handelt es sich bei der Darstellung um eine Punktwolke und nicht um einen zusammenhängenden Graphen, da es sich hier um voneinander unabhängige Netze handelt. Die Darstellung in der Klammer gibt die Anzahl der Neuronen in der ersten und zweiten verdeckten Schicht an. Jedes Netz besitzt 27 Eingangsneuronen (3x3 Pixel x 3 Farben) und 2 Ausgangsneuronen. Die Farben der Messpunkte stellen farblich die Anzahl der ersten verdeckten Schicht dar. Rot entspricht einem Neuron, grün 2 Neuronen, blau 3 Neuronen, gelb 4 Neuronen und rosa 5 und 10 Neuronen. Zu beachten wäre auch, dass die Darstellung der Abszisse sich den Ergebnissen anpasst und sich daher nicht in nachfolgenden Darstellungen konstant verhält. In Abbildung 65 ist die Messreihe dargestellt, welche die Analyse über die Einzelbilder erfasst. Auch hier wurde der Datensatz verwendet, der alle Bilder beinhaltet. Hier ist zusätzlich zu den oben genannten Darstellungsmethoden die Standardabweichung der Genauigkeit als schwarzer Balken für die einzelnen Messpunkte erfasst. Anders als bei den vorhergegangenen Analysen der Segmentierungsverfahren soll hier auf eine separate Darstellung der Datensätze der nassen, normalen und trockenen Bilder verzichtet werden. Eine gemeinsame Darstellung ist aus Gründen der Übersichtlichkeit nicht sinnvoll.¹⁶ Sämtliche Graphen wurden jedoch erzeugt und es konnte festgestellt werden, dass keine nennenswerten Unterschiede zwischen den einzelnen Wetterbedingungen herausstechen.

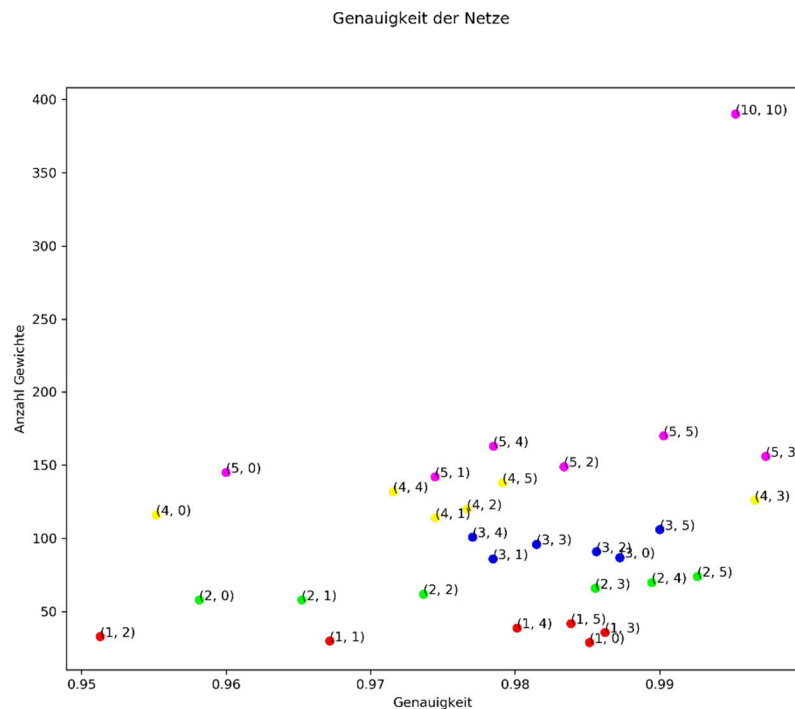


Abbildung 64: Darstellung der Genauigkeit der untersuchten Netze auf den Datensatz, der alle Bilder beinhaltet und keine Analyse der Einzelbilder vornimmt.

¹⁶ Sämtliche Graphen von der Genauigkeit, der Precision, des Recalls und des F1-Scores, sowie über die Analyse über alle Bilder und Einzelbilder würde 64 Graphen erzeugen ohne notwendige Informationen zu generieren.

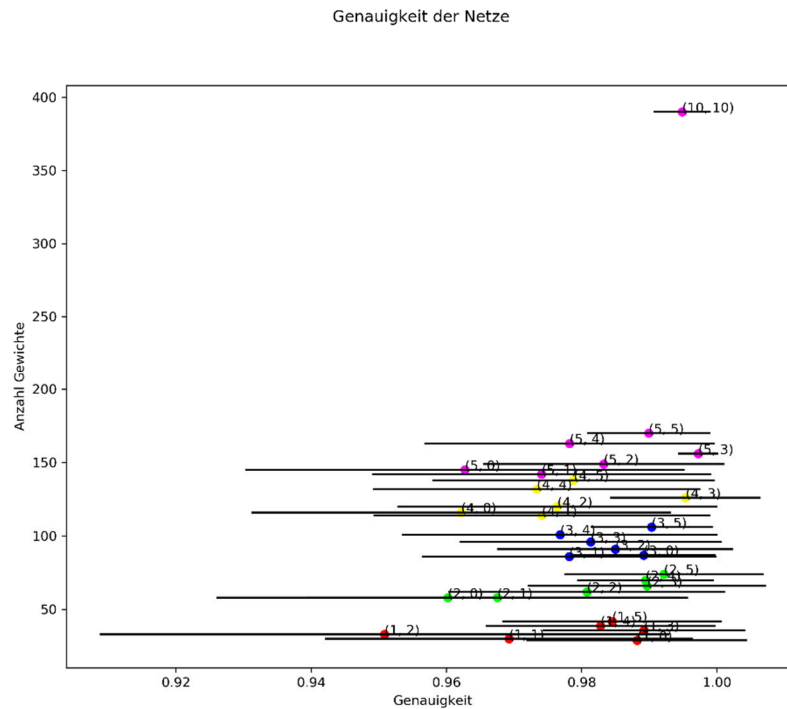


Abbildung 65: Darstellung der Genauigkeit der untersuchten Netze auf den Datensatz, der alle Bilder beinhaltet und die Analyse über die Einzelbilder vornimmt.

Wie in den beiden Graphen zu sehen ist, gibt es keinen kausalen Zusammenhang zwischen der Anzahl der Neuronen der Netze und der erreichten Genauigkeit. Alle Netze erreichen mehr als 90% der Genauigkeit in beiden Messmethoden. Durch Betrachtung der Verteilung der Genauigkeit sind sowohl kleine, als auch große Netze im Maximalbereich, daher kann davon ausgegangen werden, dass die Verteilung nicht mit der Netzwerkstruktur zusammenhängt. Die Unterschiede werden eher durch das Training der Netze zu Stande kommen. Die Netze wurden zwar alle mit denselben Lernbildern trainiert, jedoch ist ihre anfängliche Gewichtsinitialisierung zufällig. Dadurch könnte beim Backpropagation-Verfahren anhand dieser unterschiedlichen Startbedingungen ein lokales Minimum statt des globalen erreicht worden sein. Die Werte für den Recall liegen in der Nähe von 100% für beide Messmethoden. Lediglich die Precision-Werte (siehe Abbildung 66) unterscheiden sich. In der Messung, in der die Analyse keine Einzelbilder betrachtet, liegen alle Werte über 90%, bei der Einzelbild-Analyse bricht die Messung jedoch zusammen.

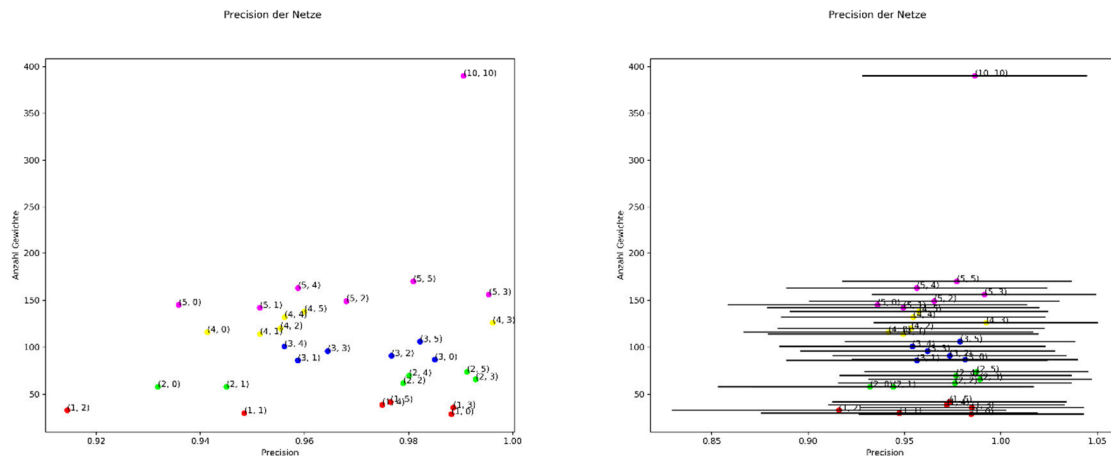


Abbildung 66: Darstellung der Precision aller untersuchten Netze; Links: Analyse ohne Einzelbildbetrachtung; Rechts: Analyse mit Einzelbildbetrachtung.

Die Diskrepanz der beiden Messungen entsteht durch eine unterschiedliche Gewichtung der Bilder. Bei der Einzelbildmessung besitzt ein Precision-Wert, der aus kleinen Parametern (True-Positiv und False-Positiv) besteht, dieselbe Wertigkeit beim Mitteln über die Bilder, als wenn der Precision-Wert aus hohen Parametern berechnet wurde. Es spielt lediglich das Verhältnis zwischen den Parametern selbst eine Rolle. Wenn jedoch erst alle True-Positiv und False-Positiv-Werte über alle Bilder summiert werden und anschließend der Precision-Wert berechnet wird, spielen Bilder mit kleinen True-Positiv und False-Positiv-Werten eine kleinere Rolle. Um eine bessere Analyse des Precision-Wertes durchführen zu können, soll die Art der False-Positiv Werte genauer betrachtet werden.

Das Problem der Hintergrundpixel, die als Pflanzenpixel klassifiziert werden (False-Positiv), lässt sich mit Hilfe des in Abbildung 67 dargestellten Diagramms verdeutlichen. Auf der Abszisse sind alle Pixel, die durchschnittlich pro Bild als False-Positiv eingestuft wurden dargestellt. Die Ordinate zeigt die durchschnittliche Anzahl von zusammenhängenden Strukturen aus False-Positiv eingestuften Pixeln von allen Bildern. Eine Fläche, beziehungsweise seine Pixel, wurde in der Messung als False-Positiv klassifiziert, wenn von einer gesamten zusammenhängenden Struktur weniger als die Hälfte der Pixel nicht durch den Segmentierer der richtigen Klasse zugeordnet wurden. Eine Skalierung ist hier nicht notwendig, da es sich einzig auf Hintergrundpixel bezieht, die als Pflanzenpixel segmentiert worden sind (False-Positiv).

Die Zahlen neben den Messpunkten geben, wie bei den anderen Graphen auch, die Anzahl der ersten und zweiten verdeckten Schicht an. Der ideale Wert, d.h. ohne False-Positive Pixel und damit auch ohne Flächen, läge im Ursprung des Diagramms, also unten links.

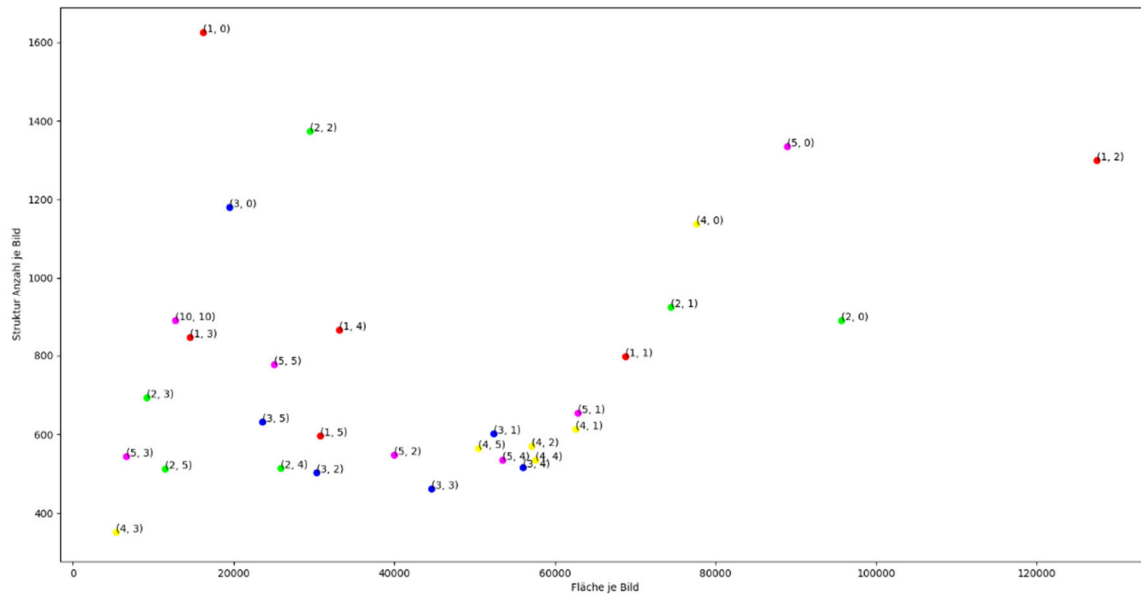


Abbildung 67: Darstellung der False-Positiv Werte aller untersuchten Netze anhand der Größe der Pixelfläche und Anzahl der Strukturen

Auch in dieser Grafik zeigt sich, dass kein Muster zwischen Komplexität und Güte der Ergebnisse zu erkennen ist. Das beste Netz (Dargestellter Text: (4, 3)) besteht aus 27 Eingangsneuronen. Es liegen 4 und 3 Neuronen in der verdeckten Schicht und 2 Ausgangsneuronen befinden sich circa am Punkt 5.000 Flächenpixel und 250 Strukturen. Dieses Netz hat auch die besten Genauigkeitswerte in beiden Messvarianten. Das am einfachsten aufgebaute und damit schnellste Netz (Dargestellter Text: (1, 0)), dass lediglich aus 27 Eingangsneuronen, einem verdeckten Neuron und 2 Ausgangsneuronen besteht, steht ungefähr am Punkt mit 18.000 Flächenpixel und 1.650 Strukturen. Es erzeugt also eine Vielzahl von kleinen fehlerhaften Pflanzenpixeln. Wie oben erwähnt, scheint das Ergebnis jedoch stark vom Lernergebnis abzuhängen. Daher soll im nächsten Schritt das Netz, welches nur ein verdecktes Neuron besitzt, näher untersucht werden. Dieses Netz wurde weitere 25-mal neu trainiert. Die Ergebnisse der Strukturflächendarstellung sind in Abbildung 68 dargestellt. Die roten Punkte geben die 25 neu untersuchten Netze an, die lediglich ein verdecktes Neuron nutzen. Im Hintergrund sind die vorher untersuchten Netzwerke transparent zum Vergleich dargestellt.

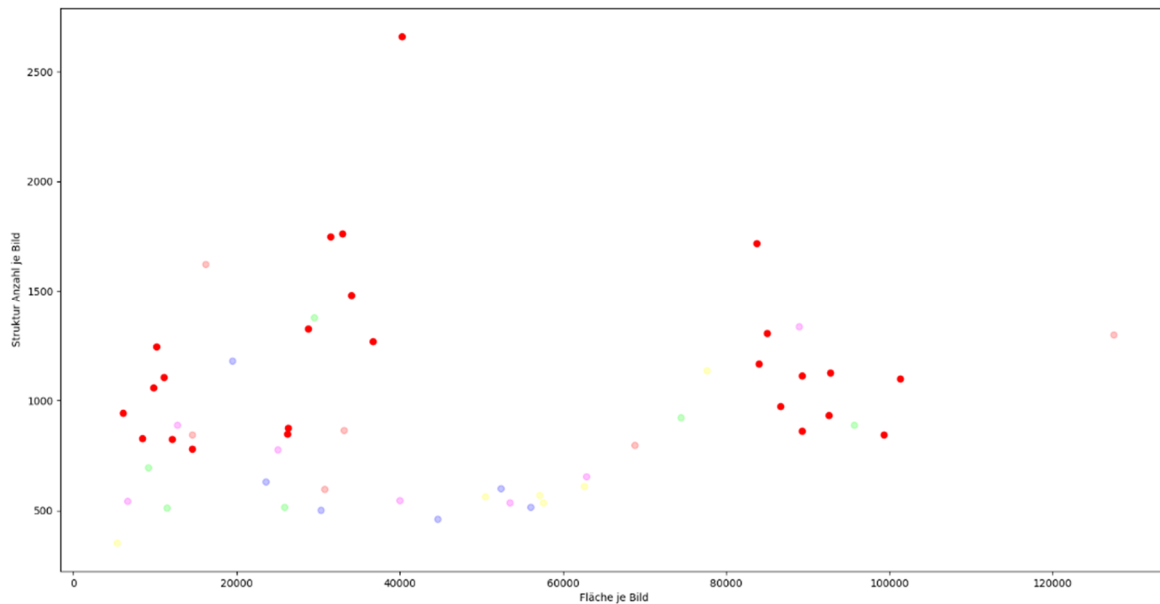


Abbildung 68: Darstellung der Strukturflächenanalyse der neu untersuchten Netze

Es sollte beachtet werden, dass die Skalierung der Ordinate anhand der Ergebnisse angepasst worden ist. Weiter fällt auf, dass es die einzelnen Ergebnisse des Netzes verbessert hat, jedoch noch nicht an das beste Netz herangekommen ist¹⁷. Zum Vergleich mit den anderen Netzen soll daher das Netz mit 4 und 3 verdeckten Neuronen verwendet werden. Für die globale Genauigkeitsanalyse ergibt sich:

Tabelle 18: Statistische Auswertung des MLP zur Segmentierung mit der Analyse ohne einzelne Bildelemente

Datensatz	Genauigkeit	Precision	Recall	F1-Score
alle	0,997	0,996	0,997	0,997
nass	0,995	0,994	0,996	0,995
normal	0,998	0,998	0,998	0,998
trocken	0,997	0,996	0,998	0,997

und für die lokale Genauigkeitsanalyse:

Tabelle 19: Statistische Auswertung des MLP zur Segmentierung mit der Analyse über einzelne Bildelemente

Datensatz	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	0,995 (0,011)	0,992 (0,058)	0,995 (0,022)	0,992 (0,059)
nass	0,995 (0,005)	0,983 (0,103)	0,997 (0,006)	0,984 (0,103)
normal	0,994 (0,018)	0,998 (0,003)	0,990 (0,037)	0,994 (0,020)
trocken	0,997 (0,002)	0,996 (0,002)	0,999 (0,003)	0,997 (0,002)

¹⁷ Ein nächster Schritt läge darin, die Netze, die nur ein Neuron in der verdeckten Schicht haben, erneut zu trainieren, jedoch alle mit derselben Initialisierung der Gewichte beginnen zu lassen. So könnte ein weiterer Lerneinfluss ausgeschlossen werden. Leider fehlten für diese Untersuchung bislang die Zeit und Rechenressourcen.

Alle Messwerte zeigen bei der skalierten Messung Ergebnisse im oberen Neunzigprozent-Bereich. Wie bei den anderen Verfahren haben die Umwelteinflüsse keine besondere Auswirkung.

Für die Signifikanzanalyse soll das Netzwerk mit der besten Genauigkeit von 99,7% verwendet werden. Dieser Genauigkeitswert wurde über alle Bilder ermittelt, ohne einzelne Bilder zu betrachten und soll nun mit den jeweiligen Genauigkeitswerten verglichen werden, die über die Einzelbildanalyse berechnet werden. Für die Berechnung wird ein Skalierungsfaktor verwendet, der auch hier über alle Bilder des Datensatzes berechnet und anschließend auf jedes einzelne Bild angewendet wird. Die Signifikanzanalyse des Genauigkeitswertes ergibt, dass 24% der Bilder nicht in den signifikanten Grenzen und 23% der Bilder nicht in den hochsignifikanten Grenzen liegen. Jedoch sind schon bei einer angenommenen Genauigkeit von 97,2% alle Bilder in den signifikanten Grenzen und für 97,4% innerhalb der hochsignifikanten Grenzen.

Die Messung der Einzelbildanalyse¹⁸ ohne Skalierung zwischen Pflanzenpixel und Hintergrundpixel, wie sie in Tabelle 20 dargestellt ist, zeigt für die Genauigkeit des Recall-Wertes ein ähnliches Verhalten.

Tabelle 20: Statistische Auswertung des MLP zur Segmentierung mit der Analyse über einzelne Bildelemente ohne Skalierung

Datensatz	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
alle	0,996 (0,003)	0,750 (0,212)	0,995 (0,022)	0,836 (0,168)
nass	0,994 (0,004)	0,823 (0,154)	0,997 (0,005)	0,891 (0,125)
normal	0,998 (0,002)	0,624 (0,244)	0,990 (0,037)	0,937 (0,198)
trocken	0,996 (0,002)	0,807 (0,161)	0,999 (0,003)	0,882 (0,120)

In der Tabelle fällt jedoch auf, dass die Precision wie bei den anderen Segmentierungsverfahren ohne die Skalierung schlechter wird. Das Netz hat Probleme, in allen Datensätzen die hellen weißen Stellen sauber als Hintergrund zu segmentieren. Zwei Beispielbilder sind in Abbildung 69 dargestellt. Diese hellen und weißen Stellen existieren unterschiedlich stark in den verschiedenen Datensätzen. Bei dem normalen Datensatz kommt noch hinzu, dass die Pflanzen hier kleiner sind, als in den anderen Datensätzen, dadurch hat ein False-Positiv segmentierter Pixel einen stärkeren Einfluss. Die hier dargestellte Analyse der Netze ist bereits der zweite Messzyklus (Training und Auswertung). Im ersten Messzyklus wurde bei den Lernbildern kein besonderer Fokus auf die weißen Stellen gelegt, so dass diese im ersten Durchlauf lediglich mit ihrer statistischen Auftrittswahrscheinlichkeit in den Lernmasken vorkamen. Das Resultat war, dass alle weißen Stellen falsch segmentiert worden sind. Im zweiten Durchlauf wurde darauf geachtet, dass mindestens ein Drittel der Hintergrundpixel den Bereich der weißen Stellen umfasst. Dadurch konnten die Ergebnisse auf die Werte, wie sie hier dargestellt sind, verbessert werden. Es soll daher nicht ausgeschlossen werden, dass eine weitere Verbesserung durch gezielteres Auswählen der Lernmasken erreicht werden könnte.

¹⁸ Die Analyse über alle Bilder zeigt ähnliche Werte, daher soll hier nur eine Tabelle präsentiert werden.

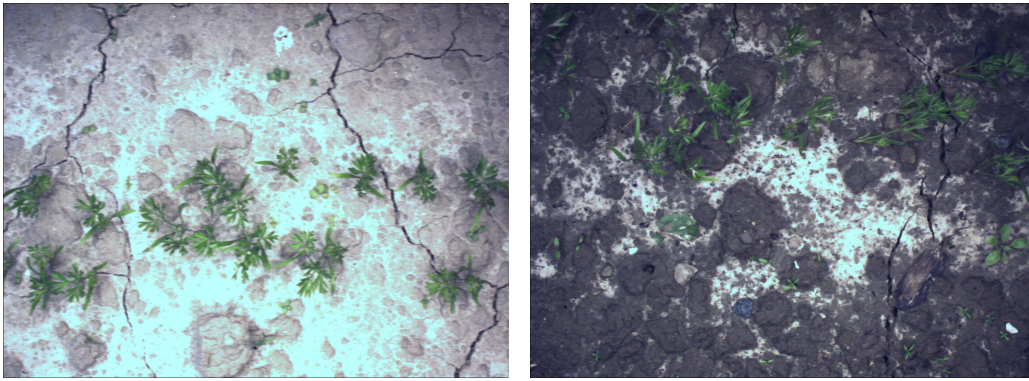


Abbildung 69: Beispielbilder (links: Datensatz trocken, rechts: Datensatz nass) mit weißen Flecken in der Hintergrundstruktur

Diese Netze verwenden lediglich ein RGB-Eingangsbild von 3x3 Neuronen. Hier kann nicht wirklich von einer komplexen Struktur wie bei einer Pflanze gesprochen werden, die sich im Eingangsbild befindet. Diese Netze werden lediglich die Farbeigenschaften zur Segmentierung verwenden. Aus diesem Grund soll nachfolgend eine weitere Netzwerkstruktur untersucht werden.

3.6.8.2 Multilayer Perceptron flächenweise Segmentierung mittels Graustufenbilder

Dieses Netzwerk besitzt erstens ein größeres Eingangsbild von 23x23 Pixel und zweitens bestehen die Eingangsbilder lediglich aus Graustufen und nicht aus RGB-Werten. Dadurch soll das Netz dazu gezwungen werden, auf die Struktur selbst zu schauen und nicht allein auf die Farbwerte. Die untersuchten Netzwerke bestehen wieder aus der Eingangsschicht mit 23x23 Neuronen, einer bis drei verdeckten Schichten und einer Ausgangsschicht aus zwei Neuronen. Wird der jeweilige minimale Pixelabstand von jedem Pflanzenpixel zum nächsten Unkrautpixel grafisch dargestellt, entsteht der in Abbildung 70 dargestellte Graph. Die Abszisse stellt den jeweiligen minimalen Pixelabstand dar. Die Ordinate ist als Wahrscheinlichkeit dargestellt. Der Graph zeigt an, wie viele Pixel mindestens den jeweils angezeigten Abstand von der Kante haben. Es haben mindestens 90% aller Pflanzenpixel einen Abstand von 12 Pixel. Soll der Abstand von 12 Pixel mindestens von dem Eingabebild abgedeckt werden, damit das Netzwerk mindestens eine Kante zum Hintergrund sieht, muss der mittlere Pixel vom Eingangsbild mindestens 12 Pixel bis zum Rand haben. Das ergibt eine Bildgröße von 23 Pixel Seitenlänge.

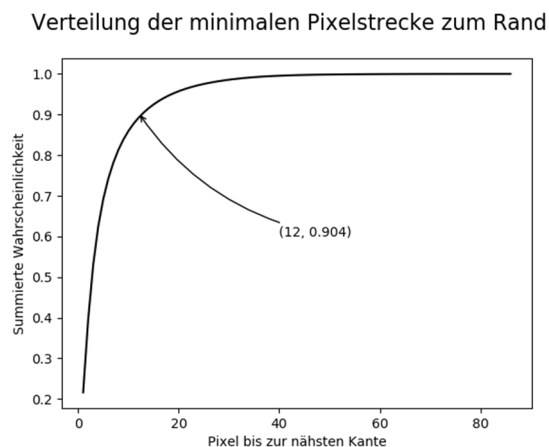


Abbildung 70: Darstellung der Verteilung des Pixelabstandes vom Pflanzenpixel zum Hintergrund

Durch eine Stichprobenuntersuchung der Variationen der Neuronen der verdeckten Schicht konnte lediglich eine maximale Genauigkeit von circa 70% erreicht werden. Der Ansatz war nun, dass ein zweites Netz hinter das vorherige Netz geschaltet wird und dieses Netz die Ergebnisse des ersten Netzes

verwendet, um die Genauigkeit zu verbessern. Die beiden untersuchten Variationen sind in Abbildung 71 dargestellt.

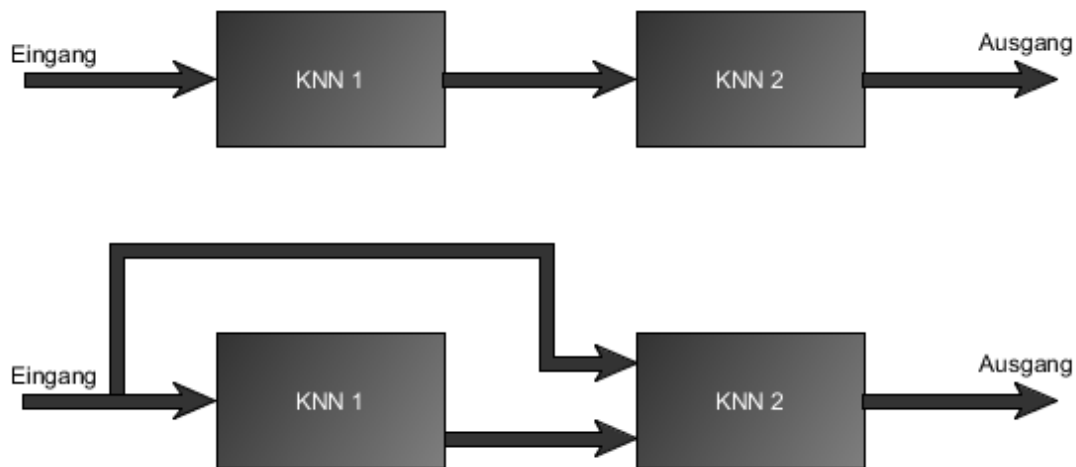


Abbildung 71: *Signalpfad der möglichen hintereinander geschalteten Netze.*

Die obere Darstellung zeigt die Variante, in der das hintere geschaltete Netz nur das Ausgangssignal des vorherigen Netzes bekommt. In der zweiten unteren Variante bekommt das zweite Netz zusätzlich noch das Eingangssignal. Das erste Netzwerk wurde ebenfalls mit festen 100.000 Epochen und einer Batchsize von 64 trainiert. Die verwendeten 100.000 Lernbilder bestehen ebenfalls gleichmäßig aus Hintergrund und Pflanzenpixel und wurden auch iterierend zusammengesetzt. Auch das zweite Netz setzt sich aus 100.000 Bildern zusammen. Dabei wurde die Verteilung auf je ein Viertel der Bilder auf True-Positiv-, True-Negativ-, False-Positiv- und False-Negativ-Ergebnisse des vorherigen Netzwerkes gelegt, so dass alle Fälle gleichmäßig vorkamen. Auch hier wurden 100.000 Epochen zum Training verwendet und eine Batchsize von 64.

Bei der oberen Variante blieb die Genauigkeit in derselben Größenordnung wie bei der Genauigkeit vom ersten Netz. In der unteren Variante verbesserte sich die Genauigkeit von circa 70% des ersten Netzes auf 74% des zweiten Netzes. Die weitere Überlegung war, das zweite Netz erneut zur Verbesserung einzusetzen, so dass iterierend das zweite Netz als Eingang neben dem Originalbild in Graustufen seinen Ausgang erhielt. Das Ergebnis führte zu einem Zusammenbrechen der Genauigkeit. Das zweite Netz schneidet zunehmend Pflanzenpixel in den Randbereichen ab und segmentiert diese zu Hintergrundpixel.

Versucht wurde außerdem noch, ein Nachhilfe-Lernverfahren einzusetzen. Hier wurden, nachdem das Training keine Veränderung der Genauigkeit mehr erzeugte, gezielt nur weitere Eingangsbilder verwendet, die das Netz zu diesem Zeitpunkt falsch klassifizierte. Die Genauigkeit veränderte sich zwar nicht, jedoch wurden je nach Trainingsschritt andere Bilder richtig und falsch klassifiziert.

Das Problem liegt am Informationsgehalt der Graustufenbilder der Jai-Kamera. Die Farbinformation der Pixel liegen für die Pflanzenpixel sehr dicht beieinander (Der Abstand¹⁹ zwischen rot/blau und dem grünen Kanal liegt bei circa 7%. Bei der Nikon liegt der Abstand bei 15% zwischen dem roten und grünen Kanal und 30% zwischen dem blauen und grünen Kanal). Dadurch hebt sich bei der einfachen Mittelung des RGB-Wertes für das Grauwertbild die Pflanze nicht vom Hintergrund ab. Das Problem ist in Abbildung 72 dargestellt, das linke Bild zeigt das RGB-Jai-Bild und das rechte Bild zeigt das resultierende Grauwertbild.

¹⁹ $(\text{Grüner Farbwert} - \text{Roter Farbwert}) / 255 = \text{Abstand in Prozent}$

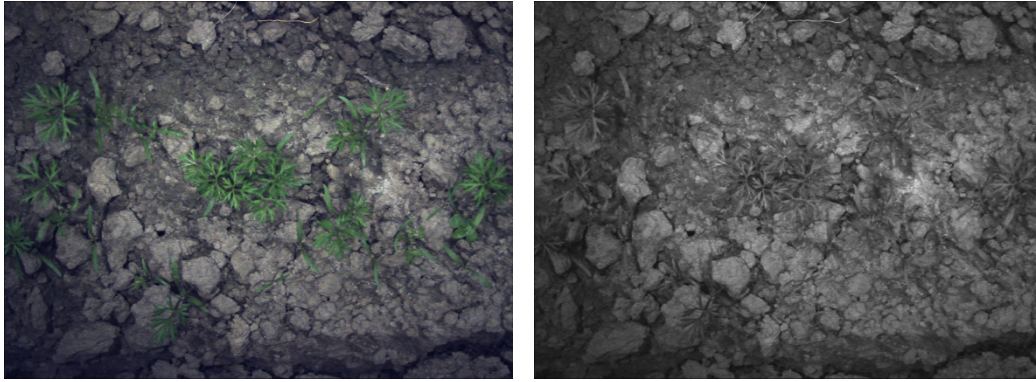


Abbildung 72: Darstellung des Grauwertproblems der Jai-Kamera; Links: RGB-Bild; Rechts: resultierendes Grauwertbild

3.6.9 Zusammenfassende Auswertung der Segmentierung

Im Folgenden soll eine Gegenüberstellung und Zusammenfassung der verschiedenen Segmentierungsverfahren erfolgen, um eine Auswahl zu treffen. Die präsentierten Ergebnisse zeigen, dass die Wetterbedingungen keinen großen Einfluss auf die Segmentierung haben. Daher soll nun ein Vergleich nur in Hinblick auf den Datensatz, der alle Bilder beinhaltet, betrachtet werden. Die nachfolgende Tabelle zeigt erneut die besten Ergebnisse der vorgestellten verschiedenen Segmentierungsverfahren unter Verwendung der jeweils besten Parameter für jedes Bild.

Tabelle 21: Auflistung aller Segmentierungsverfahren mit den jeweils besten Parametern für jedes Bild mit Skalierung

Segmentierungsverfahren	Genauigkeit (SD)	Precision (SD)	Recall (SD)	F1-Score (SD)
Einfacher Schwellwert	0,72 (0,082)	0,70 (0,088)	0,77 (0,135)	0,72 (0,098)
NDVI	0,98 (0,017)	0,98 (0,061)	0,99 (0,021)	0,98 (0,060)
RGB	0,97 (0,027)	0,97 (0,059)	0,97 (0,033)	0,97 (0,060)
HSV	0,995 (0,0069)	0,992 (0,0551)	0,996 (0,0128)	0,992 (0,0555)
MLP pixelweise	0,97 (0,022)	0,98 (0,059)	0,96 (0,039)	0,97 (0,061)
MLP flächenweise	0,995 (0,011)	0,992 (0,058)	0,995 (0,022)	0,992 (0,059)

Abgesehen von dem einfachen Schwellwert, der lediglich der Vollständigkeit halber mit aufgeführt wird, sind alle statistischen Größen im gleichen Wertebereich. Daher soll eine Analyse der Anzahl der Strukturen und der Flächen erfolgen, um die False-Positive-Werte ohne Skalierung darzustellen. Dafür werden die jeweils besten Parameter von den verschiedenen Segmentierungsverfahren in Abbildung 73 dargestellt.

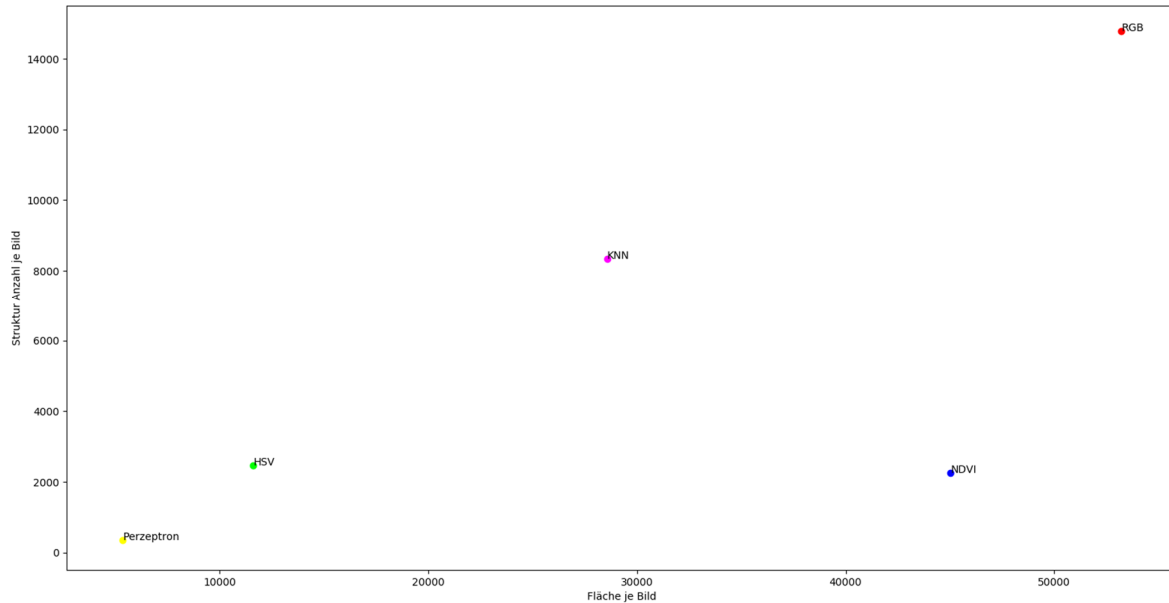


Abbildung 73: Zusammenfassende Darstellung des Struktur-Flächen-Diagramms aller Segmentierungsverfahren.

Das Diagramm zeigt, dass das MLP-flächenweise-Verfahren die besten Resultate liefert, gefolgt vom HSV-Verfahren. Es folgen mit etwas Abstand das MLP-pixelweise-Verfahren, das NDVI-Verfahren und das RGB-Verfahren.

Anhand der gewonnenen Ergebnisse soll das HSV-Verfahren für die Segmentierung verwendet werden. Das MLP-flächenweise-Verfahren zeigt zwar teilweise bessere Ergebnisse, benötigt jedoch eventuell ein Trainingsverfahren für den Fall, dass sich die Eingangsbilder stark ändern. Für die Verwendung des MLP-pixelweise-Verfahrens wird ein Algorithmus benötigt, der echtzeitfähig die Trenngeraden für jedes Bild bestimmt. Dieser Algorithmus existiert derzeit nicht. Das RGB-Verfahren zeigt im Vergleich schlechtere Ergebnisse als das HSV-Verfahren und das NDVI-Verfahren benötigt eine Bispektral-Kamera. Weiter haben die Ergebnisse gezeigt, dass das HSV-Verfahren eine große Flexibilität im Schwellwert zulässt.

Zur Untersuchung der Grenzen des HSV-Verfahrens sollen vier verschiedene Problemfälle betrachtet und weiter untersucht werden:

- Unschärfe durch Translation der Kamera.
- Unschärfe durch einen falschen Fokus des Objektivs
- Unter- und Überbelichtung
- Unter- und Übersättigung der Farbe des Bildes

Diese vier Fälle kommen in der Praxis vor. Für die Untersuchung wurde jedoch der vorher verwendete Datensatz, bestehend aus allen 293 Bildern, künstlich verändert, um eine Vergleichsprobe zu schaffen. Die Störungen sind in den Testfällen stärker vertreten als sie in der Praxis vorkommen, jedoch sollen die Grenzbereiche untersucht werden. Einige Störungen könnten auch durch eine automatische Kalibrierung der Kameraparameter verringert werden. Dies ist jedoch beim derzeitigen Aufbau nicht enthalten.

3.6.9.1 Untersuchung des Verhaltens des HSV-Segmentierungsverfahrens auf Translationsunschärfe

Für die Translationsunschärfe wurden die Pixel in der Zeile über eine variable Pixelanzahl gemittelt. Variationen der Translationsunschärfe sind in Abbildung 74 dargestellt. In der Abbildung nimmt die Translationsunschärfe von links nach rechts zu. Insgesamt wurden 30 Variationen verwendet. Das bedeutet, dass eine Mittelung vom Originalbild bis maximal 30 Pixeln in der Schrittweite von je 1 Pixel angewendet worden ist.



Abbildung 74: Darstellung der verschiedenen simulierten Translationsunscharfen am Beispiel eines Bildes; von links: Original, Integration über 10 Pixel, 20 Pixel, 30 Pixel

Wird dieser Testdatensatz, bestehend aus 30 verschiedenen Translationsbewegungen, verwendet, um erneut mit dem HSV-Segmentierungsverfahren alle Schwellwerte durchzufahren, zeigt sich die in Abbildung 75 dargestellte Variation der Genauigkeit.

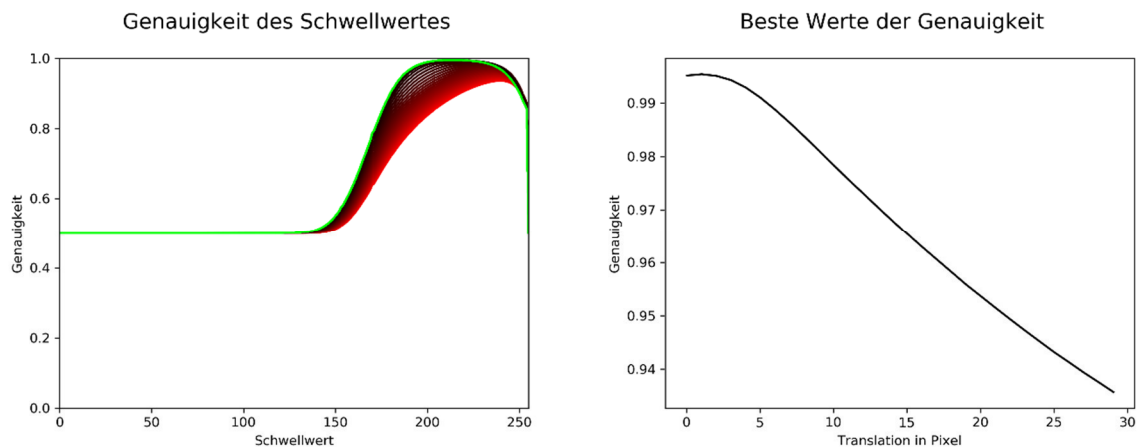


Abbildung 75: Darstellung des Zusammenhangs zwischen Genauigkeit und Translationsunschärfe; Links: Verlauf über Schwellwerte; Rechts: Veränderung der Genauigkeit über die maximalen Werte

Die Darstellung zeigt die Messung²⁰ über den Datensatz, der alle Bilder beinhaltet und keine Einzelbildanalyse verwendet. Die Analyse über Einzelbilder weist zu dieser Analyse kaum Unterschiede auf und wird daher nicht dargestellt. Der grüne Graph zeigt den Verlauf der Genauigkeit über die verschiedenen Schwellwerte über die originalen Bilder, also ohne Translationsunschärfe. Die anderen Graphen sind durch einen Farbverlauf von Schwarz zu Rot durch zunehmende Translation gekennzeichnet. Das heißt, je höher die Translationsunschärfe, desto roter ist der Graph. Es zeigt sich, dass durch Zunahme der Translationsunschärfe die Genauigkeit von circa 100% auf 93,5% abfällt²¹. Der Schwellwert für die besten Genauigkeitswerte wird mit zunehmender

²⁰ Auch hier wurde wieder eine Skalierung zwischen Pflanzen und Hintergrund verwendet. Außerdem wurden der Randbereich des Bildes sowie die zwei Pixel breiten Kanten zwischen Pflanzen und Hintergrund ignoriert, um die Sonderfälle, die mit dem angewendeten Filter und der Ungenauigkeit des Ground truth auftreten, nicht in die Messung einfließen zu lassen.

²¹ Wenn der Verlauf des rechten Bildes in Abbildung 75 ganz genau betrachtet wird, ist zu erkennen, dass die Genauigkeit erst leicht ansteigt. Dieser Anstieg entspricht jedoch drei Promille und soll daher nicht weiter beachtet werden.

Unschärfe größer. Durch die Unschärfe werden die einzelnen Werte gemittelt. Das bedeutet, dass Pflanzenpixel am Randbereich einen niedrigeren Grünwert bekommen und einen höheren Rot- und Blauwert. Der Hintergrund bekommt hingegen einen höheren Grünwert und einen niedrigeren Rot- und Blauwert, so dass sich der Abstand vom Grünwert zum Rot- und Blauwert verringert. Dadurch muss auf einer Seite der Schwellwert höher sein und das hat zur Folge, dass die Auswahl des Schwellwertes nicht mehr so robust ist, da eine feinere Einstellung erfolgen muss.

3.6.9.2 Untersuchung des Verhaltens des HSV-Segmentierungsverfahrens auf allgemeine Unschärfe

Sollte der Fokus nicht richtig eingestellt werden, verhalten sich die Bilder ähnlich. Die Simulation wendet auf die Bilder einen Gauß-Filter mit den Kernelgrößen vom Originalbild und 1, 3, 5, 7 bis 25 an. Das bedeutet, dass mit dem größten Kernel auf einer Fläche von 25 x 25 Pixel über eine Gaußkurve gemittelt wird. Beispielbilder sind in Abbildung 76 dargestellt.

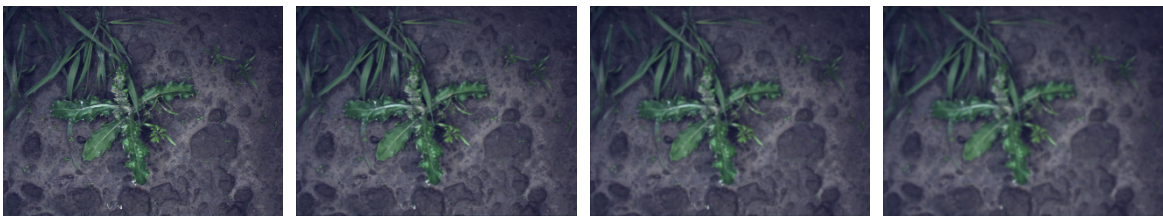


Abbildung 76: Darstellung der verschiedenen simulierten allgemeinen Unschärfen am Beispiel eines Bildes; Von links: Original, Kernel 9, 17 und 25

Anders als bei der Translationsunschärfe zeigt sich, dass die Auswirkungen auf den Algorithmus minimal sind. Die Genauigkeit²², wie sie in Abbildung 77 dargestellt ist, weist kleine Schwankungen auf und wenn die besten Schwellwerte verwendet werden, wie im rechten Bild, ist der Abfall minimal. Bei genauer Betrachtung kann auch erkannt werden, dass durch die Verwendung des Gaußfilters der Schwellwert noch robuster wird.

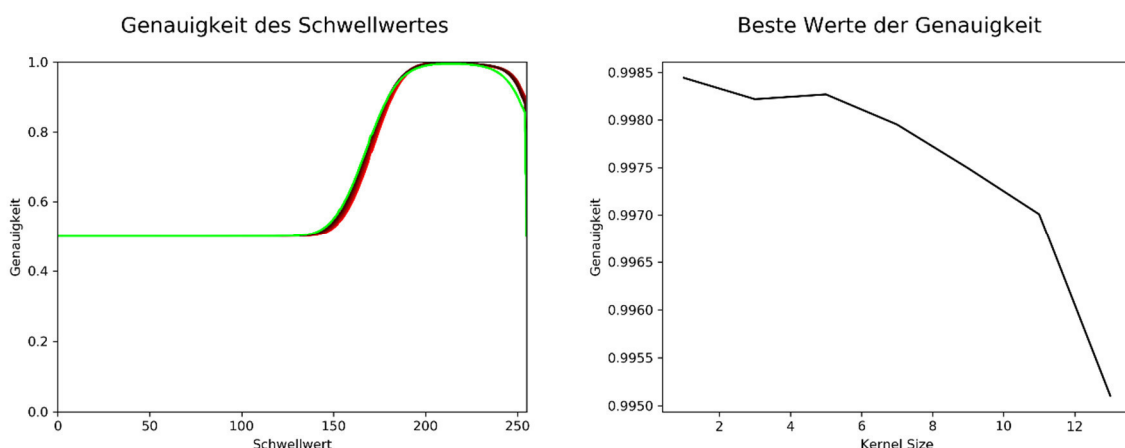


Abbildung 77: Darstellung des Zusammenhangs zwischen Genauigkeit und Unschärfe; Links: Verlauf über Schwellwerte; Rechts: beste Genauigkeit der einzelnen Kernelgrößen

²² Auch hier wurde wieder eine Skalierung zwischen Pflanzen und Hintergrund verwendet und der Kantenbereich zwischen Pflanze und Hintergrund, der im ground truth fehlerhaft sein kann, wurde ebenfalls bei der Auswertung ignoriert.

Dass es hier keinen Abfall der Genauigkeit mit zunehmendem Schwellwert gibt, wie es bei der Translationsunschärfe der Fall ist, liegt zum einen daran, dass hier über eine Fläche von maximal 25 x 25 Pixel mit einer Gaußkurve gemittelt wird. Damit besitzen die jeweiligen Klassen einen höheren Einfluss, als wenn über einen 1 Pixel breiten Streifen, aber mit einer Länge von 30 Pixeln, linear gemittelt wird. Die jeweiligen Pixel der Klasse werden dadurch etwas homogener. Der Abfall der Genauigkeit ist dadurch im Promillebereich. Auch hier zeigt der grüne Graph den Verlauf über die Originalbilder und die zunehmende Kernelgröße des Gaußfilters wird durch die Zunahme des roten Farbwertes der Graphen dargestellt. Der Gaußfilter mit einer Kernelgröße von 1 wird schwarz dargestellt und der Gaußfilter mit einer Kernelgröße von 25 wird rot dargestellt.

3.6.9.3 Untersuchung des Verhaltens des HSV-Segmentierungsverfahren auf Unter- und Überbelichtung

Ein weiteres Problem, das in der Praxis auftaucht, ist eine Unter- oder Überbelichtung des Bildes. Dies kann zum Beispiel passieren, wenn die Blende des Objektivs falsch eingestellt worden ist oder auch durch die falsche Einstellung der künstlichen Beleuchtung, beziehungsweise der Integrationszeit der Kamera. Für die Untersuchung wurden die Bilder von 0% bis 300% in 10% Schritten belichtet. Das Originalbild hat in der Skala einen Wert von 100%. In der Abbildung 78 sind vier Fälle dargestellt, 30%, 100%, 200% und 300% der Belichtung.

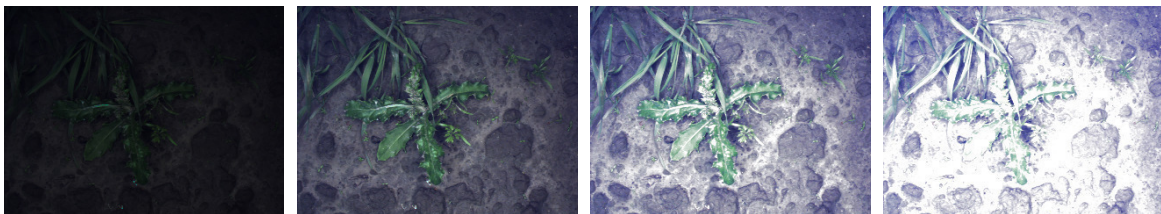


Abbildung 78: Darstellung der verschiedenen simulierten Belichtungsfälle; Von links: 30%, Original, 200% und 300% Belichtung

Die in Abbildung 79 dargestellten Graphen zeigen die Analyse, die keine Einzelbildbetrachtung vornimmt. Auch hier ist der Graph über die Originalbilder (100% Belichtung) grün dargestellt. Mit zunehmender Belichtung besteht ein Farbverlauf von Schwarz zu Rot. Das heißt, dass 0% Belichtung dem schwarzen Graphen entsprechen, der konstant bei einer Genauigkeit von 0,5 verläuft. Eine Belichtung von 0% bedeutet, dass in den Bildern keine Informationen mehr vorhanden sind, da die Bilder komplett schwarz sind. Die Ergebnisse des Segmentierens, die mit einer Skalierung zwischen Pflanzen und Hintergrund ausgewertet werden, sind zu 50% der Segmentierungsfälle richtig. Mit zunehmender Helligkeit nimmt auch der Rotwert des Graphen zu. Der dunkle Graph, der treppenförmig verläuft, zeigt ein Verhalten, in dem durch 10% Helligkeit wenige Informationen zu erkennen sind und diese durch den Schwellwert sprunghaft detektiert werden. Der Schwellwert des HSV-Segmentierungsalgorithmus hat keinen direkten Zusammenhang mit dem eigentlichen Farbwert. Durch die Skalierung des Schwellwertes in der Segmentierungsformel des HSV-Algorithmus können auch kleine Farbwerte einen hohen Schwellwert triggern. Wird die Genauigkeit wie im rechten Bild wieder über den jeweiligen besten Wert der Belichtungsreihe dargestellt, ist ersichtlich, dass mit zunehmender Belichtung oberhalb der Originalbelichtung von 100% die Genauigkeit abnimmt. Durch eine Zunahme der Belichtung werden zunehmend Informationen im Bild auf einen maximalen Wert gesetzt und so nimmt die Information im Bild ab. Wird die Variation bei 100% der Belichtung betrachtet, zeigt sich, dass der HSV-Segmentierungsalgorithmus zwar bei 100% der Belichtung seinen besten Genauigkeitswert hat, jedoch mit unterbelichteten Bildern immer noch eine Genauigkeit von über 0,9 möglich ist. Bei keiner Belichtung, also 0%, fällt der Wert auf 0,5 wie bereits oben beschrieben.

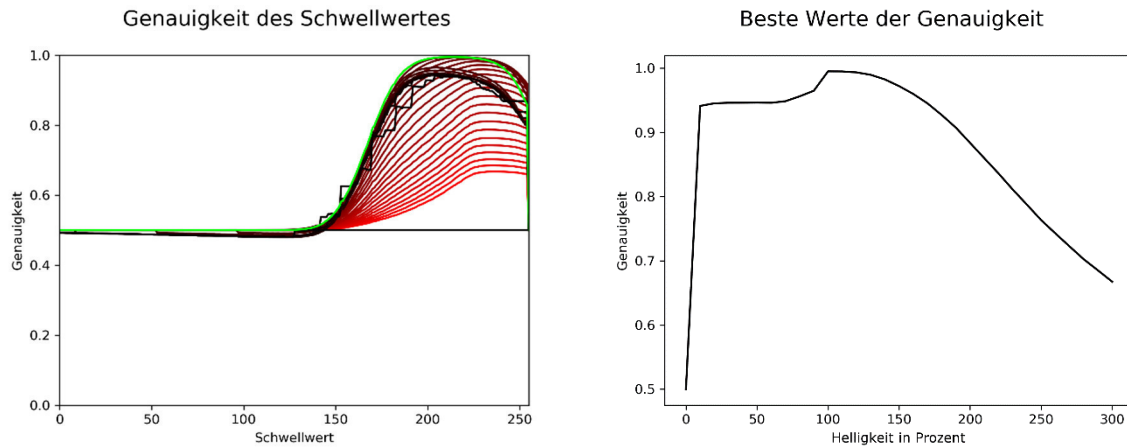


Abbildung 79: Darstellung des Zusammenhangs zwischen Genauigkeit und Belichtungsstärke; Links: Verlauf über Schwellwerte; Rechts: beste Genauigkeit über einzelne Belichtungsstufen

3.6.9.4 Untersuchung des Verhaltens des HSV-Segmentierungsverfahrens auf Unter- und Übersättigung der Farbe

Abschließend soll der Einfluss der Farbsättigung untersucht werden. Auch diese Problematik kann zum Beispiel durch eine falsche Einstellung der Kameraparameter entstehen. Die Farbsättigung soll genauso wie die Helligkeit im Bereich der Untersättigung und der Übersättigung untersucht werden. Auch hier sollen vier Beispielbilder gezeigt werden. In Abbildung 80 ist ein Bild mit 0%, 100%, 200% und 300% Farbsättigung dargestellt. Dabei stellt der 100%-Wert das Originalbild dar. Alle Bilder wurden wie bei der Helligkeit in 10% Schritten von 0% bis 300% variiert.

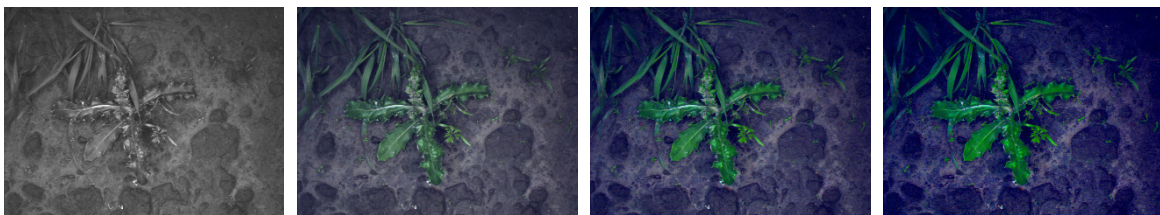


Abbildung 80: Darstellung der verschiedenen simulierten Farbsättigkeiten; Von links: 0%, Original, 200% und 300% Farbsättigung

Wie bei den anderen Analysen stellt in Abbildung 81 die Analyse ohne Einzelbildbetrachtung dar. Die Messung verwendet eine Skalierung zwischen Pflanzen und Hintergrund und es wurden auch in dieser Messung die Kanten zwischen Pflanzen und Hintergrund ignoriert. Die Ergebnisse sind wieder farblich dargestellt. Mit zunehmendem Rotwert der Graphen nimmt auch die Farbsättigung der Datensätze zu. Der schwarze Graph zeigt die Farbsättigung von 0%, was Grauwertbilder repräsentiert. Dieser Fall ist mit dem HSV-Algorithmus nicht segmentierbar, dadurch liegt der Graph konstant bei einer Genauigkeit von 0,5. Der 100%-Graph, der die Originalbilder repräsentiert, ist auch hier in grün dargestellt. Die Analyse zeigt, dass der Schwellwert mit zunehmender Farbsättigung einen größeren Toleranzbereich zulässt, also ein größerer Bereich für den Schwellwert gewählt werden kann. Wenn die besten Genauigkeitswerte des jeweiligen Datensatzes dargestellt werden (wie im rechten Bild), dann ist ersichtlich, dass die Farbsättigung von 0% einen Genauigkeitswert von 0.5 erreicht, anschließend jedoch die Genauigkeit nahezu konstant bis zum Originalfarbwert bleibt, dann leicht abfällt und erneut einen konstanten Verlauf annimmt.

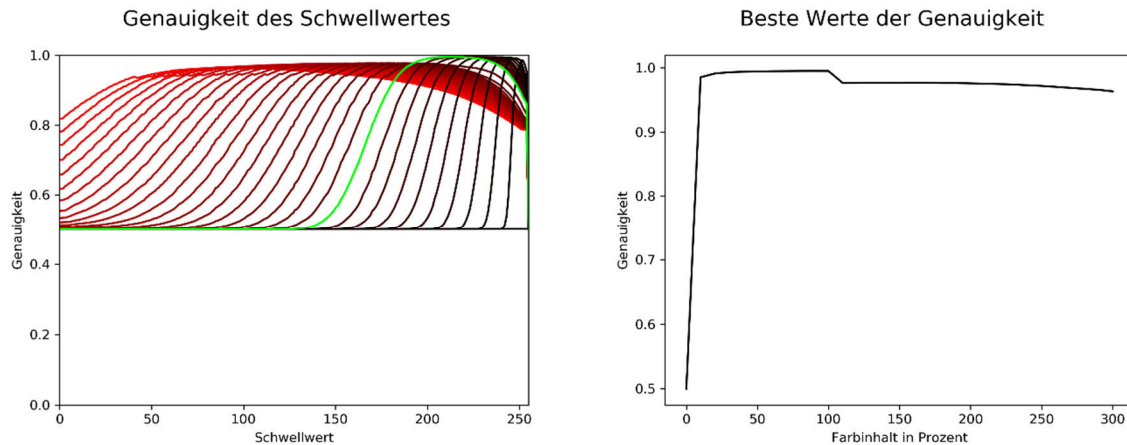


Abbildung 81: Darstellung des Zusammenhangs zwischen Genauigkeit und Farbintensität; Links: Verlauf über Schwellwerte; Rechts: beste Genauigkeit über einzelne Farbintensitätsstufen

3.6.10 Fazit zu der Segmentierung

Es wurden neben dem einfachen Schwellwert auf dem grünen Kanal und dem Verfahren, das auf dem state-of-the-art NDVI-Vegetationsindex beruht, vier weitere Methoden vorgestellt, um Pflanzen vom Hintergrund zu segmentieren. Alle Verfahren wurden mit derselben Datengrundlage untersucht, um einen direkten Vergleich zu gewährleisten. Dabei wurde auf die Grenzen der Datengrundlage eingegangen. Nach der Gegenüberstellung der verschiedenen Segmentierungsverfahren wurde das HSV-Segmentierungsverfahren unter Begründung der Messwerte für eine weitere Untersuchung ausgewählt. Dieser Algorithmus verknüpft die Ergebnisse verschiedener Schwellwerte, die er auf das S- und das H-Bild des HSV-Farbraums anwendet. Durch eine mathematische Zusammenführung der einzelnen Berechnungsschritte des Verfahrens ergibt sich eine einfache Formel, die auf jeden Pixel des Bildes angewendet werden kann. Das HSV-Verfahren wurde dann tiefer auf reale Probleme hin untersucht, um die Grenzen zu erfassen. Dabei zeigte sich, dass das HSV-Segmentierungsverfahren äußerst robust auf Einflüsse reagiert und nur einen leichten Genauigkeitsabfall erfährt. Auch zeigt die Signifikanzanalyse, dass die Schwankung der Genauigkeiten minimaler ist als bei den anderen Verfahren. Einflüsse durch Unschärfe und eine Unter- oder Übersättigung der Farbe kann die Methode kompensieren. Eine zu starke Überbelichtung führt durch den Verlust der Informationen im Bild zu einem Zusammenbrechen des Verfahrens. Dies hätte jedoch auf jedes Verfahren einen Einfluss. Daher soll das HSV-Segmentierungsverfahren in der Arbeit zur Segmentierung verwendet werden.

Eine interessante Frage, die zukünftig beantwortet werden sollte, ist, wie genau ein Segmentierer arbeiten muss, damit die nachfolgende Klassifizierung ohne Probleme vollzogen werden kann. Hier gibt es zwei Aspekte, die zu berücksichtigen sind. Einmal sollte ein Segmentierer möglichst wenige Hintergrundpixel (False-Positiv) erzeugen, damit nicht unnötige Hintergrundfragmente einer Pflanzen-Klasse zugewiesen werden und damit unter Umständen die Vernichtungseinheit blockieren. Der zweite Aspekt wäre, dass der Segmentierer die Kanten einer Pflanze möglichst gut maskiert, damit der Klassifikator den Umriss der Pflanze verwenden kann.

3.7 Fazit zur Sensorauswahl und Segmentierung

In diesem Kapitel wurde am Anfang beschrieben, unter welchen Bedingungen die Arbeit auf dem Feld verrichtet werden muss. Dabei wurden das Trägersystem und die Module für das autonome Jäten gezeigt. Neben den Möglichkeiten, die zur Vernichtung der Beikräuter verwendet werden können, wurden die untersuchten Sensoren vorgestellt. Die Sensoren wurden jede Saison mehrere Wochen getestet, um eine Datengrundlage mit verschiedenen Wachstumsstadien, Pflanzen und Wetterbedingungen zu erstellen. Bezüglich der Messergebnisse wurden die verschiedenen Sensoren verglichen und das Resultat ist, dass für die Arbeit eine RGB-Kamera zum Klassifizieren verwendet werden soll. Bevor der Klassifikator aus den RGB-Bildern die Pflanzen klassifizieren kann, müssen diese erst segmentiert werden, damit die Datenlast auf die wichtigsten Informationen beschränkt wird. Hierfür wurden im Kapitel verschiedene Segmentierungsmethoden vorgestellt und untereinander, sowie mit dem State-of-the-Art NDVI-Verfahren verglichen. Dafür wurde jede Methode aus verschiedenen Blickwinkeln statistisch ausgewertet. Auch eine Aussage über die Datengrundlage wurde getroffen, um die Grenzen der Statistik zu zeigen. Für die verschiedenen Messungen wurde ein Hypothesentest durchgeführt, um eine Aussage über die Signifikanz der Datensätze zu treffen, die für die Auswertung der verschiedenen Methoden verwendet worden sind. Am Ende wurde anhand der Ergebnisse der Algorithmus ausgewählt, der die Segmentierung über den HSV-Farbraum berechnet. Dieser Algorithmus verknüpft die Ergebnisse verschiedener Schwellwerte, die er auf das S- und das H-Bild des HSV-Farbraums anwendet. Durch eine mathematische Zusammenführung der einzelnen Berechnungsschritte des Verfahrens ergibt sich eine einfache Formel, die auf jeden Pixel des Bildes angewendet werden kann. Dieser Algorithmus wurde daher tiefergehend auf reale Problemfälle untersucht, um die Grenzen der Verfahren zu definieren. Der nächste Schritt ist nun, die segmentierten Daten auf Karotten und Beikraut zu klassifizieren. Dies soll tiefergehend im nächsten Kapitel hergeleitet und beschrieben werden.

4. Der Klassifikator

4.1 Einleitung

Die Klassifikation, also die Unterscheidung zwischen Nutzpflanze und Beikraut, ist in diesem Projekt die komplexeste Aufgabe und soll im nachfolgenden Kapitel ausführlich diskutiert werden. Die gesamte Vorverarbeitung, wie die Sensorenauswahl und das Untersuchen der Segmentierung, ist im vorherigen Kapitel erläutert worden. Die Komplexität der Klassifikation der Pflanzen ist schwierig, da eine Pflanzenart, zum Beispiel die Karotte, zwar untereinander ähnliche Merkmale besitzt, jedoch jede Pflanze auch anders aussieht. Dazu kommen große Unterschiede durch das Wachstumsstadium und Wetterbedingungen. Versuche mit einem Random-Forest-Klassifikator zeigten zwar, dass eine Genauigkeit von circa 94 Prozent²³ erreicht werden kann, jedoch lag die Geschwindigkeit mit diesem Klassifikator auf einem Feld bei lediglich 4,5 cm pro Sekunde [Haug, et al., 2014]. Der Geschwindigkeitsmalus des Random-Forest-Klassifikators beruht auf der aufwändigen Featureberechnung im Bild, die der Klassifikator zur eigentlichen Entscheidung braucht. In dieser Arbeit sollen lernfähige Datenanalysen verwendet werden, auf Basis der künstlichen neuronalen Netze. Durch die steigenden Rechenkapazitäten erlangten die neuronalen Netze eine neue Renaissance. Es gibt mittlerweile verschiedene Architekturen von neuronalen Netzen, um die komplizierten Bildklassifikationen durchzuführen. Daher sollen im ersten Teil des Kapitels verschiedene neuronale Netze auf die Klassifikation von Karotte und Beikraut untersucht werden, anschließend folgen eine Gegenüberstellung der verschiedenen Ergebnisse und die Auswahl einer Architektur. Dabei soll vorerst ausschließlich auf die erreichbare Genauigkeit hin untersucht und nicht auf die Geschwindigkeit Rücksicht genommen werden. Die ausgewählte Architektur soll dann genauer untersucht werden. Im zweiten Teil des Kapitels sollen die Algorithmen dieser Architektur auf Geschwindigkeit optimiert werden. Dazu soll eine selbst entwickelte CPU vorgestellt, sowie eine Optimierung der Berechnungen auf einer GPU präsentiert werden. Abschließend wird die Netzwerktopologie der Architektur auf die Optimierungen angepasst. Am Ende wird ein Vergleich zwischen der herkömmlichen Abarbeitung und der optimierten Abarbeitung durchgeführt, sodass ein direkter Vergleich der Genauigkeit und Geschwindigkeit möglich ist.

²³ Der in der Veröffentlichung [Haug, et al., 2014] verwendete Datensatz unterscheidet sich von der Häufigkeit des Vorkommens der Nutzpflanze, sowie der Größe der Beikräuter sehr stark von den in dieser Arbeit vorliegenden Daten. Dies hat unter anderem den Grund, dass es sich um Testfelder in einer anderen Region handelt und auch andere Arbeitsabläufe als auf dem Westhof verwendet worden sind. Außerdem ist die Genauigkeit und Geschwindigkeit als Mittelwert angegeben. Ohne jede Kenntnis über die Verteilung oder Signifikanz der Werte kann eine Güte dieser Aussage nicht nachvollzogen werden.

4.2 Ground Truth Erstellung und Aufteilung des Datensatzes für die Klassifikation

Für das Training und Testen der neuronalen Netze müssen die zur Verfügung stehenden Bilder von Hand klassifiziert werden, wie bereits bei der Segmentierung²⁴ des Hintergrundes. Für die Handklassifikation wurde ein speziell dafür geschriebenes Programm entwickelt, das es erlaubt, jede Pflanze auf einem Bild einer Art und Identifikationsnummer zuzuweisen²⁵. Mit diesem Programm wurden 1324 Bilder klassifiziert. Dies ergibt eine Anzahl von 41.626 Pflanzen, die sich aufteilt in Karotten mit 21.812 Pflanzen und Beikräuter mit 19.814 Pflanzen. Insgesamt ergibt dies eine Gesamt-Pixelanzahl von 105.345.104 Pflanzenpixeln. Die Bilder stammen dabei aus einem Datensatz von drei Jahren und zeigen Pflanzen bei verschiedenen Wetterbedingungen und in unterschiedlichen Wachstumsstadien. Auch hier sollte berücksichtigt werden, dass durch die menschliche Komponente für die Ground Truth Erstellung nicht von einer hundertprozentig fehlerfreien Arbeit ausgegangen werden kann.

Die 1324 Bilder wurden in die Lerndaten (249 Bilder) und die Validierungsdaten (1075 Bilder) geteilt. Damit kann sichergestellt werden, dass eine Vermischung von Trainings- und Validierungsdaten später nicht die Ergebnisse verfälscht. Für alle folgenden Messungen wurde die gleiche Aufteilung des Datensatzes verwendet. Die Bilder wurden per Hand in 23 verschiedene Pflanzenarten eingeteilt, die sich weniger an biologischen Kriterien, sondern eher an bildverarbeiteten Objekteigenschaften wie Größe und Blattstruktur orientieren. Dadurch wurden ähnliche Pflanzen auch in einer ähnlichen Art zusammengefasst²⁶. Dadurch, dass einige Pflanzenarten jedoch im gesamten Datensatz nur wenige Male vorkamen, wurde diese Einteilung zu größeren Gruppen zusammengefasst, um eine bessere statistische Aussage treffen zu können. Eine Einteilung nach Wetterbedingungen erfolgt hier nicht, da die Anzahl der Pflanzen in den Gruppen noch weiter verkleinert worden und eine statistische Aussage damit nicht mehr gegeben wäre. Außerdem zeigte die Untersuchung bei dem Segmentierer, dass das Wetter zumindest auf den Segmentierer keinen großen Einfluss besitzt. Für den Validierungsdatsatz ergibt sich damit die Aufteilung:

Tabelle 22: Auflistung der zusammengefassten Pflanzengruppen für die Auswertungen

Pflanzengruppen	Pflanzenanzahl	Pflanzenanzahl	Pflanzenanzahl
Karotte ohne 3. Blatt	13.247	17.867	32.983
Karotte mit mind. 3 Blatt	4.620		
Beikraut klein	13.786	15.116	
Beikraut mittel	581		
Beikraut Knöterich	422		
Beikraut groß	327		

Die Karotten wurden in zwei wesentliche Gruppen eingeteilt: Einmal in die Gruppe „Karotte ohne 3. Blatt“ (Dies sind die Karottenpflanzen wie sie gerade aus dem Erdboden kommen.) und in die andere Gruppe mit Karotten mit mindestens 3 Blättern. Dadurch, dass der Roboter ziemlich früh nach dem

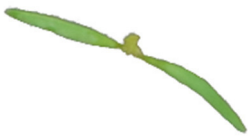
²⁴ Kapitel 3.6.1 auf Seite 43

²⁵ Es wurde ebenfalls die Idee getestet, durch einen Sticking-Algorithmus [Holtorf, et al., 2016] das gesamte Feld zu einem Bild zusammenzusetzen. Anschließend wird dann jede Pflanze auf dem zusammengesetzten Bild von Hand klassifiziert. Spätere Aufnahmen von demselben Feld (andere Wachstumsperiode) könnten dann durch die zusammengesetzten Bilder übereinander gelegt werden und müssten nicht neu manuell klassifiziert werden. Neue Pflanzen wären höchstwahrscheinlich Beikräuter. Durch zu viel Toleranz der einzelnen Bilder war das Verfahren jedoch zu unpräzise.

²⁶ Von einer exakten biologischen Untersuchung wurde hier abgesehen, da die studentischen Hilfskräfte keine großen biologischen Kenntnisse über die Pflanzen mitbrachten. Da es auch unerheblich für den Erfolg des Projektes ist, die Beikräuter an sich exakt unterscheiden zu können wurde der Arbeitsschritt dahingehend vereinfacht.

ersten Herauskommen der Karotten jätet, bildet die erste Gruppe statistisch gesehen die Mehrheit. Das Beikraut wird in drei wesentliche Gruppen eingeteilt. Diese Gruppen orientieren sich an der Größe der jeweiligen Pflanzen. Das mittlere Beikraut ist dabei speziell noch weiter unterteilt worden, so dass der Knöterich (in der Größe des mittleren Beikrauts) als separate Gruppe abgespalten wurde. Dies hat den Grund, dass diese Pflanze der Karotte sehr ähnlich sieht und hier eine genauere Betrachtung erfolgen soll. Außerdem kann durch diese Einteilung ein eventueller menschlicher Fehler beim Labeln besser ausgeschlossen werden, da diese Gruppe separat geprüft werden kann. Diese Einteilung wurde manuell durchgeführt, daher kann die Einteilung bei der Datenmenge verständlicherweise auch Fehler beinhalten. Für das bessere Verständnis sind in Tabelle 23 einige Pflanzen aufgelistet für die jeweilige Gruppe. Es sollte jedoch bedacht werden, dass die Gruppen wesentlich mehr Variationen in Größe, Form und Struktur beinhalten als die drei hier jeweils aufgezeigten Varianten. Die Bildgrößen besitzen dabei, bis auf die Gruppe „Beikraut groß“ dieselbe Skalierung, so dass die Größenunterschiede ersichtlich sind. Damit die Gruppe „Beikraut groß“ dargestellt werden kann, wurden diese Bilder für die Darstellung verkleinert. Außerdem sollte bedacht werden, dass die Pflanzen im Datensatz nicht immer so sauber wie hier extrahiert werden können, da viele Pflanzen sich überlappen. Viele der Pflanzen weisen an bestimmten Strukturen ähnliche Muster auf, so ist die Blattspitze teilweise identisch. Auch die Kamillepflanze (Beikraut mittel, Variante #2 in Tabelle 37) zeigt Ähnlichkeiten mit dem dritten Blatt der Karottenpflanze. Durch die starken Variationen der Pflanzen in sämtlichen Größen und Formen kann ein einfacher Größenvergleich (zum Beispiel: Pixelfläche oder Umfang) hier nicht helfen, da die Karotte auch in fast allen Größen vorkommt.

Tabelle 23: Auflistung der zusammengefassten Pflanzengruppen für die Auswertungen

Pflanzen- gruppen	Variante #1	Variante #2	Variante #3
Karotte ohne 3. Blatt			
Karotte mit mind. 3 Blatt			
Beikraut klein			
Beikraut mittel			
Beikraut Knöterich			
Beikraut groß			

4.3 Grundkonzept zu den Klassifikatoren

Eine Möglichkeit, ein Bild zu klassifizieren, ist es, das zu klassifizierende Objekt in dem Bild zu lokalisieren und anschließend zu klassifizieren. Dieser Ansatz erwies sich hier leider nicht als effizient, da eine Unterscheidung von zwei benachbarten Pflanzen, sofern sich diese beiden Pflanzen überlappen, nicht trivial ist. Daher folgende Überlegung: Ein Bild besteht aus Pixeln und jeder Pixel kann in einem idealen Bild²⁷ nur genau eine Klasse beinhalten. Daher wäre ein trivialer Ansatz, das gesamte Bild pixelweise zu klassifizieren [Knoll, et al., 2018]. Dies soll heißen, dass jeder Pixel nacheinander einer Klasse zugeordnet wird. Da jedoch ein einzelner Pixel nicht die nötige Information enthält, um eine Aussage über die Klassenzugehörigkeit zu treffen, muss eine umliegende Fläche zur Klassifikation herangezogen werden. Dies liegt daran, dass allein die Pixelfarbe nicht ausreicht, um eine Entscheidung zu fällen, es wird zum Beispiel die Geometrie benötigt. Diese Fläche kann dann mit einem Klassifikator genauer untersucht werden, um eine Klassifizierung vorzunehmen. Damit ergibt sich für diese Art der Klassifikation der in der Abbildung 82 dargestellte Signalfluss zu dem Klassifikator, der im Folgenden als pixelweise Klassifikation bezeichnet werden soll.

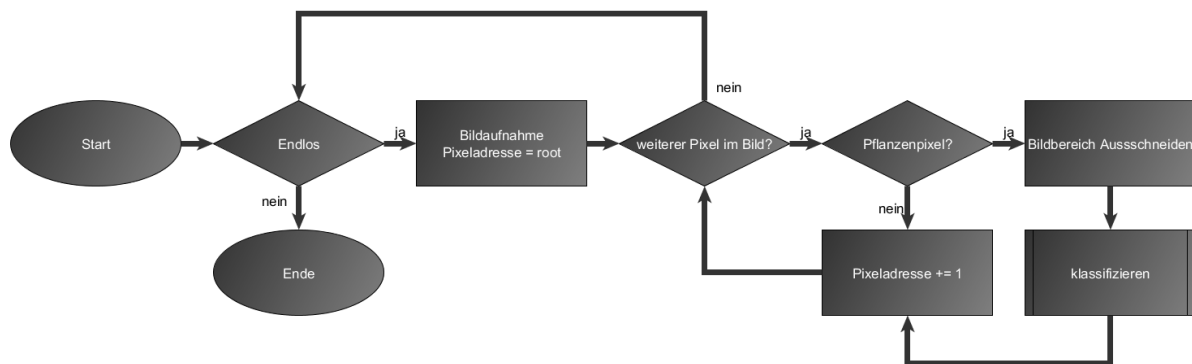


Abbildung 82: Signalflussgraph der pixelweisen Klassifikation

Der Algorithmus arbeitet wie abgebildet in einer Endlosschleife und liest als Erstes ein neues Bild ein. Nun wird für jeden Pixel separat mit dem im Kapitel 3.6 untersuchten Segmentierungsalgorithmus geprüft, ob es sich um einen relevanten Pflanzenpixel handelt. Sollte es sich um ein Pflanzenpixel handeln, wird ein Segment aus dem Bild ausgeschnitten. Dieses Segment hat als Center-Koordinate den gefundenen Pflanzenpixel. Der Center-Pixel wird anschließend mit Hilfe eines Klassifikators klassifiziert. Der Vorgang wiederholt sich dann solange, bis alle Pflanzenpixel in dem Bild klassifiziert worden sind. Es entsteht damit ein klassifiziertes Bild. Anschließend wird ein neues Bild geladen und der Vorgang beginnt von neuem. Dieses Konzept soll im Folgenden mit verschiedenen neuronalen Netzen untersucht werden. Ein anderes untersuchtes Konzept verwendet den Klassifikator, um eine gesamte Fläche auf einmal zu klassifizieren. Dieses Konzept soll jedoch in dem entsprechenden Kapitel genauer vorgestellt werden [Knoll, et al., 2018].

²⁷ Durch Unschärfe könnte darüber philosophiert werden, dass in einem Pixel zwei Klassen vorkommen.

4.4 Die verwendeten Auswertungen für die Pflanzenklassifikation

Im vorhergehenden Kapitel²⁸ wurde beschrieben, wie sich der Datensatz zusammensetzt und sich für das Training und die Validierung aufteilt. Die hier untersuchten Klassifikatoren sind alle lernfähigen Algorithmen. Damit sichergestellt werden kann, dass die Klassifikatoren nicht die einzelnen Klassifikationsfälle lediglich auswendig lernen, ist es unter anderem wichtig, dass die Lerndaten und die Validierungsdaten sauber voneinander getrennt und unabhängig sind. Dies wird dadurch erreicht, dass für beide Datensätze unterschiedliche Bilder verwendet werden. Aus den Bildern des Lerndatensatzes werden anschließend die Trainingsfälle extrahiert. Der erste Schritt besteht darin, dass 100.000 Eingangsbilder der Größe von 101x101 Pixel aus den Trainingsbildern extrahiert werden. Die beiden Klassen Karotte und Beikraut kommen dabei jeweils mit 50.000 Fällen vor. Die Verteilung der Eingangsbilder ist zufällig über die Trainingsbilder verteilt. Das bedeutet auch, dass zwei Eingangsbilder lediglich um einen Pixel verschoben sein können. Werden kleinere Bildausschnitte benötigt (zum Beispiel 51x51 Pixel), werden diese aus den Bildern mit 101 x 101 Pixel extrahiert. Alle Netze wurden dann mit demselben Datensatz trainiert.

Im nächsten Schritt werden die 100.000 Bilder in 10 Trainingsdatensätze getrennt. In einem Datensatz sind dann 10.000 Bilder, 5000 Karotten und 5000 Beikräuter immer iterierend angeordnet. Die 10 Trainingsdatensätze werden anschließend für eine 10-Fache Kreuzvalidierung beim Training verwendet. Die Kreuzvalidierung sieht vor, dass der Klassifikator 10 mal neu trainiert wird, immer mit 9 der 10 Trainingsdatensätzen, bei jedem neuen Training wird ein neuer Trainingsdatensatz nicht verwendet. Der jeweils nicht verwendete Datensatz wird während des Trainings zum Testen verwendet. Durch dieses Vorgehen wird sichergestellt, dass eine gewisse Unabhängigkeit des Trainingsdatensatzes besteht. Der Trainingserfolg am Ende wird durch die Berechnung der Genauigkeit angegeben. Die Genauigkeit ist dabei der arithmetische Mittelwert der Einzelgenauigkeiten der 10 Trainingsdurchläufe. Dadurch, dass die Einzelwahrscheinlichkeiten sich kaum unterscheiden, kann ein Mittelwert gebildet werden. Außerdem zeigt sich, dass die Datensätze keine statistischen Ausreißer der Testfälle beinhalten. Diese Vorgehensweise soll lediglich den Trainingsfortschritt anzeigen. Dadurch, dass zwei Trainingsbilder lediglich um einen Pixel verschoben sein können und dass diese zwei Bilder in zwei unterschiedlichen der 10 Datensätze sein können, kann die Auswertung angreifbar sein.

Für eine genauere und sicherere Untersuchung wurden ein trainierter Gewichtsatz aus den 10 Trainingsdurchläufen der Kreuzvalidierung gewählt und die 1075 unabhängigen Validierungsbilder komplett klassifiziert. Anschließend wurden die Klassifikationsergebnisse ausgewertet mit dem im Folgenden beschriebenen Verfahren. Da die Bildausschnitte, welche die 1075 Validierungsbilder zeigen, keine statistische Relevanz besitzen, wird diese Untersuchung die 32.983 Pflanzen betrachten, beziehungsweise ihre 82.514.098 Pflanzenpixel.

Globale Genauigkeit

Die erste Untersuchung wurde bereits bei der Segmentierung²⁹ verwendet. Hier werden die statistischen Größen (True-Positiv, True-Negativ, False-Positiv und False-Negativ) über alle Bilder summiert. Anschließend werden diese Werte skaliert, damit eine Unabhängigkeit der Häufigkeit der Klassen nicht die Untersuchung beeinflusst. Aus den skalierten Größen werden anschließend für jede Klasse (Karotte und Beikraut) die Genauigkeit, die Precision, der Recall und der F1-Score berechnet. Die Berechnung ist in Formel (59) angegeben.

²⁸ Kapitel 4.2 auf Seite 98

²⁹ Kapitel 3.6.2 auf Seite 46

Lokale Pflanzengenauigkeit

Anschließend soll eine Aussage über die Genauigkeit innerhalb einer Pflanze getroffen werden: die Pixelgenauigkeit. Dafür werden wieder die statistischen Größen True-Positiv, True-Negativ, False-Positiv und False-Negativ berechnet, jedoch diesmal für jede einzelne der 32.983 Pflanzen. Dadurch kann für jede Pflanze separat die Genauigkeit, die Precision, der Recall und der F1-Score berechnet werden. Für eine bessere Übersichtlichkeit werden die Größen skaliert über die Pflanzengruppen (Karotte ohne 3. Blatt, Karotte mit mind. 3 Blatt, Beikraut klein, Beikraut mittel, Beikraut Knöterich und Beikraut groß) gemittelt und angegeben. Abschließend wird mittels Histogrammanalyse die Verteilung der Genauigkeiten aufgezeigt, um die Signifikanz aufzuzeigen. In dem Histogramm wird auf der Ordinate die Anzahl und auf der Abszisse die Genauigkeit aufgetragen. Damit kann im Histogramm abgelesen werden, wie viele Pflanzen welche Pixelgenauigkeit erreicht haben.

4.5 Die untersuchten Pflanzen-Klassifikatoren

4.5.1 Multilayer Perceptron-Netz zur Pflanzenklassifikation

Der erste Ansatz zur Klassifikation bestand darin, ein MLP aufzubauen, wie es bereits bei der Segmentierung in Kapitel 3.6.8 verwendet wurde³⁰. Die untersuchten Netzwerke besitzen dabei eine Input und Output Schicht sowie 1 bis 3 verdeckte Schichten (Hidden-Layer). Die Eingabeschicht verwendet eine Bildgröße von 101x101 Pixel. Die Eingangsgröße wurde so gewählt, dass auch große Pflanzen repräsentativ in dem Ausschnitt abgebildet sind. Wenn eine Seitenlänge des Eingangsbildes von 81 Pixel gewählt wird, ist der mittlere Pixel 40 Pixel vom Bildrand entfernt. In Abbildung 70³¹ kann abgelesen werden, dass circa 98% aller Pflanzenpixel nicht weiter als 40 Pixel vom Pflanzenrand entfernt sind. Es wurde daher ein Abstand von 50 Pixel (also eine Seitenlänge von 100 Pixel³²) gewählt, damit ein gewisser Bereich von Nicht-Pflanzenpixeln im Bildausschnitt enthalten ist, wodurch Konturen besser erkennbar sind. Die Ausgangsschicht besitzt zwei Neuronen, eins für die jeweilige Klasse (Karotte und Beikraut). Für die Suche nach einer funktionierenden Netzstruktur wurden mehrere Rechner parallel verwendet, um verschiedene zufällige Netzwerkpermutationen auszuprobieren. Als Abbruchkriterium wurde für die Voruntersuchung eine feste Epochenzahl von 100.000 gewählt. Die Batchgröße wurde auf 64 Bilder festgelegt. Die Netze nahmen nach circa 40.000 Epochen ihr Maximum an. Für das Training wurden aus den oben beschriebenen Trainingsbildern 100.000 Bildelemente von 101x101 Pixeln Größe extrahiert und mittels einer zehnfachen Kreuzvalidierung trainiert. Beide Klassen waren in der Trainingsmenge gleichverteilt und wurden alternierend in das Batch geladen. Das Trainingsergebnis war dabei identisch³³. Es zeigte sich jedoch, dass schon bei dieser Untersuchung die Netzwerkarchitektur nicht ausreichte, um höhere Genauigkeiten zu erreichen.

³⁰ Es wurde eine Bachelorarbeit vom Autor zu diesem Thema an Herrn Dennis Pfau [Pfau, 2016] vergeben und betreut. Die Ergebnisse dieser Arbeit konnten jedoch nicht in dieser Dissertation verwendet werden.

³¹ Abbildung 70 befindet sich auf Seite 87

³² Damit ein mittlerer zu klassifizierender Pixel existiert, muss die Seitenlänge ungerade sein, daher wurden 101 Pixel verwendet.

³³ Die Genauigkeitsergebnisse von funktionierenden Netzwerken schwankten um circa 1,5 Prozent.

Tabelle 24: Darstellung der Auswertungsergebnisse über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung

Klasse	Neuronen			Genauigkeit	Precision	Recall	F1-Score
	H ₁	H ₂	H ₃				
Karotte	100	-	-	0,76	0,71	0,86	0,78
Beikraut					0,83	0,66	0,73
Karotte	100	100	-	0,79	0,74	0,92	0,82
Beikraut					0,89	0,67	0,77

In Tabelle 24 sind die Ergebnisse zweier MLP-Netze dargestellt. Die Ergebnisse unterliegen einer Skalierung, um die unterschiedliche Anzahl der Karottenpixel und Beikrautpixel auszugleichen. Diese beiden Netze sollen als Auszug aller untersuchten MLP-Netzwerke das Problem darstellen. Es sind in der Tabelle dafür jeweils die Ergebnisse der Karottenpixel und der Beikrautpixel separat dargestellt. In der Spalte Neuronen sind mit H₁, H₂ und H₃ die Anzahl der Neuronen der verdeckten Schicht dargestellt. Beide Netze verwenden das RGB Eingangsbild von 101x101 Pixel. Dabei ist der Hintergrund (in der Regel der Erdboden) nicht auf Schwarz gesetzt, sondern natürlich vorhanden. Ebenfalls besitzen beide Netze zwei Ausgangsneuronen, eins für jede Klasse. Die Schwellwertfunktion ist hier die ReLu-Funktion³⁴ und über die Ausgangsinformationen wird noch eine Softmax-Funktion³⁵ angewendet.

Die Auswertung in der Tabelle betrachtet die Ergebnisse aller Pixel der 1075 Evaluierungsbilder (globale Genauigkeitsanalyse). Dabei ist keine Einteilung in Bilder oder Pflanzen mehr vorhanden. Es wurde jedoch eine Skalierung zwischen den Karottenpixeln und den Beikrautpixeln vorgenommen, damit ein gleichmäßiges Vorkommen beider Klassen für die Auswertung vorliegt. Die Genauigkeiten zeigen mit 0,76 und 0,79, dass die Netzwerke kein zufälliges Ergebnis liefern und Ansätze von richtiger Klassifikation zeigen. Auch die Precision, der Recall und damit der F1-Score liegen über der 50% Marke, die ein reines Zufallsergebnis ausschließen lässt. Die beiden Netzwerke bilden die besten Ergebnisse der gesamten Untersuchung ab. Das Netzwerk, das aus zwei verdeckten Schichten mit je 100 Neuronen besteht, ist marginal besser als das Netzwerk, das nur eine Schicht verwendet. Der Einfachheit halber soll daher mit diesem Netzwerk weiter argumentiert werden. Die Ergebnisse sind jedoch vergleichbar mit dem anderen Netzwerk. Drei Schichten brachten hier keine weitere Verbesserung der Ergebnisse. Wie verwertbar dieses Ergebnis ist, zeigt folgende Analyse.

Im Folgenden wird die Betrachtung der einzelnen Pflanzen auf den 1075 Bildern durchgeführt. Dafür werden die etwa 32.000 Pflanzen separat ausgewertet und für jede Pflanze eine Genauigkeit berechnet. Diese Genauigkeit gibt an, wie viel Prozent der Pixel einer Pflanze richtig klassifiziert wurden. Anschließend werden die einzelnen Pflanzengenauigkeiten in einem Histogramm dargestellt. Das Histogramm ist in Abbildung 83 zu sehen.

³⁴ Die **Rectifier Linear Unit** ist in Kapitel 2.2.5.1 beschrieben und in Abbildung 17 dargestellt.

³⁵ Die Softmax-Funktion ist in Kapitel 2.2.5.3 beschrieben.

Verteilung der Genauigkeiten

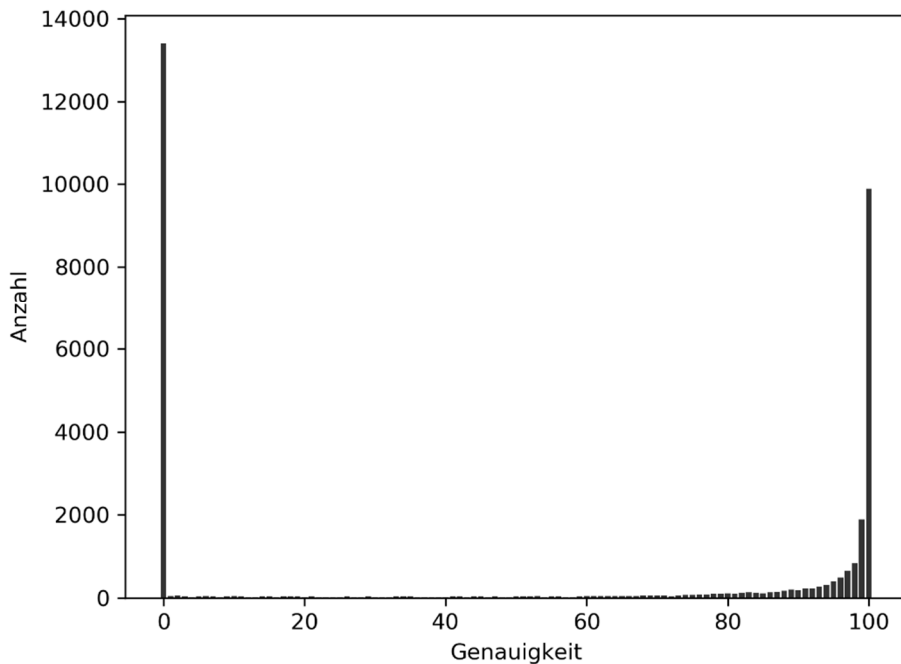


Abbildung 83: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des ersten Klassifikators

Das Histogramm zeigt, dass sich weit mehr als 70% aller vorkommenden Genauigkeiten (sowohl von Karottenpflanzen, als auch von Beikrautpflanzen) auf zwei Peaks verteilen. Dabei stellen beide Peaks auch die Extrema der Genauigkeiten da. Die meisten Pflanzen besitzen nach der Klassifikation 0% richtig klassifizierte Pixel. Der zweite Peak zeigt, dass circa $\frac{1}{3}$ aller Pflanzen zu 100% richtig klassifiziert werden. Das heißt, dass alle Pixel dieser Pflanzen der richtigen Klasse zugeordnet werden können. Da es sich hier nicht um eine Normalverteilung handelt, kann nicht ein Hypothesentest wie bei der Segmentierung angewendet werden. Wird das Histogramm in die beiden Klassen (Karotte und Beikraut) unterteilt, wie dies in Abbildung 84 dargestellt ist, wird ersichtlich, dass vor allem das Klassifikationsergebnis eher für die Klasse „Karotte“ ausfällt.

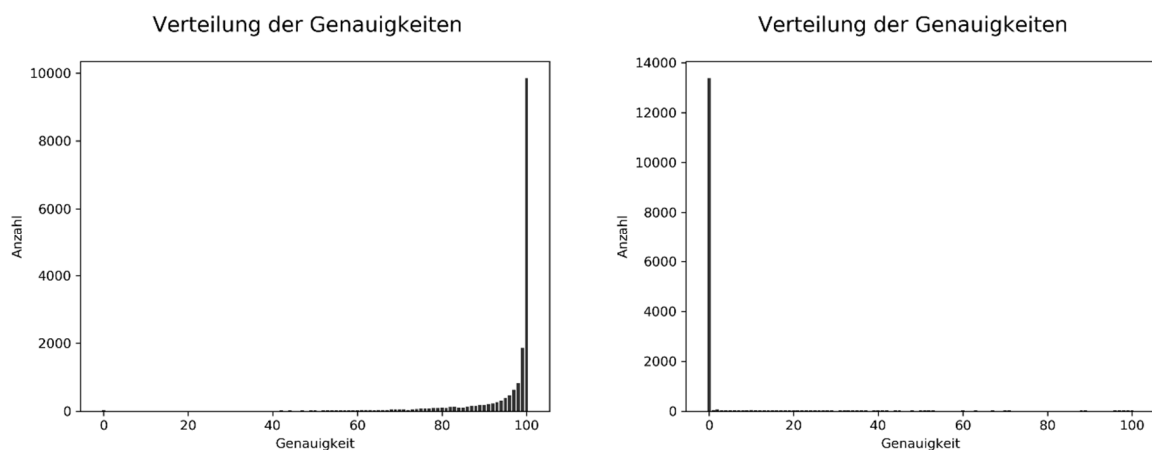


Abbildung 84: Darstellung der Genauigkeitshistogramme des ersten Klassifikators, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“

Bei der Abbildung sollte bedacht werden, dass die Achsendarstellung unterschiedlich ist. Im rechten Bild, das die Verteilung der Beikrautgenauigkeit angibt, existieren auch Genauigkeiten größer als 0%.

Da diese jedoch lediglich ein paar Hundert Mal vorkommen, ist dies durch die Skalierung in der Abbildung nur schwer zu erkennen. Auf eine weitere Darstellung soll aber aus Gründen der Übersichtlichkeit verzichtet werden. Genauso wie die Ergebnisse für die einzelnen Kategorien der Klassen die eine analoge Charakteristik aufweisen. Das Netzwerk ist so nicht für die Klassifikationsaufgabe geeignet.

Zur Verbesserung der Klassifikation sollen die MLP-Netzwerke erneut trainiert werden. Jedoch soll dabei der Hintergrund durch den Segmentierer auf den Eingangsbildern auf den Wert 0 (Schwarz) gesetzt werden. Damit könnte das KNN sich leichter auf die Struktur einer Pflanze konzentrieren und muss diese nicht erst selbst vom Hintergrund filtern. Auch dafür wurden wieder verschiedene Permutationen von Netzwerken trainiert, wieder mit einem RGB Eingangsbild von 101x101 Pixeln und mit zwei Ausgangsneuronen. Außerdem wurden wieder 100.000 Epochen als Grenze gesetzt. Auch hier war das Maximum von allen Netzwerken nach circa 40.000 Epochen erreicht. Die Trainingsbilder waren dabei identisch mit denen des ersten Trainings, lediglich der Hintergrund wurde auf Schwarz gesetzt. Ein Teil der trainierten Netze ist in Tabelle 25 mit ihren Ergebnissen dargestellt. Auch hier wurde wieder eine Skalierung verwendet, um die unterschiedliche Anzahl der Klassenpixel auszugleichen.

Tabelle 25: Darstellung der Auswertungsergebnisse über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung und schwarzem Hintergrund

Klasse	Neuronen			Genauigkeit	Precision	Recall	F1-Score
	H ₁	H ₂	H ₃				
Karotte	100	-	-	0,86	0,86	0,87	0,86
Beikraut					0,87	0,86	0,86
Karotte	100	100	-	0,86	0,82	0,92	0,87
Beikraut					0,91	0,80	0,85
Karotte	100	100	100	0,87	0,84	0,90	0,87
Beikraut					0,90	0,84	0,87
Karotte	200	-	-	0,87	0,86	0,89	0,87
Beikraut					0,88	0,85	0,87
Karotte	200	200	-	0,86	0,82	0,90	0,86
Beikraut					0,89	0,81	0,85
Karotte	200	200	200	0,87	0,85	0,91	0,88
Beikraut					0,90	0,83	0,87

Die Tabelle zeigt, dass alle Netzwerke bessere Ergebnisse erzielen, als beim ersten Versuch ohne schwarzen Hintergrund. Es zeigt sich auch, dass alle Netzwerke identische Ergebnisse liefern und mehrere Schichten hier keine Verbesserung bringen. Damit jedoch ein direkter Vergleich zur ersten Auswertung besteht, soll hier wieder das Netzwerk verwendet werden, das zwei verdeckte Schichten mit je 100 Neuronen verwendet. Auch hier wurden von den Netzwerken die Genauigkeiten der jeweiligen Pflanzen berechnet und mit einem Histogramm die Verteilung dargestellt (siehe Abbildung 85). Auch hier ist wieder wichtig, dass zum Vergleich der einzelnen Abbildungen auf die unterschiedliche Skalierung der Achse geachtet werden muss.

Verteilung der Genauigkeiten

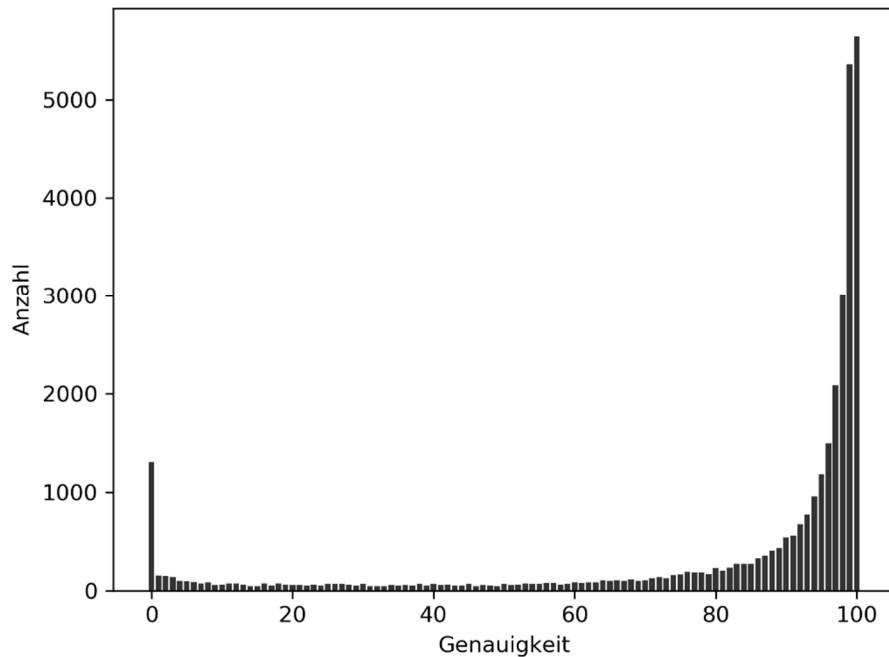


Abbildung 85: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des ersten Klassifikators mit segmentierten Bildern.

Das Histogramm zeigt, dass sich die 0% Klassifikationen (komplett falsch) verringert haben. Jedoch ist auch die Anzahl der 100% Klassifikationen (komplett richtig) von nicht ganz 10.000 auf circa 5500 zurückgegangen. Die Klassifikationsergebnisse haben sich über die Skala verteilt. Auch die Einzelbetrachtung der Klassen wie in Abbildung 86 zeigt, dass sich die Beikrautgenauigkeit verbessert hat, jedoch auch, dass die Karottengenauigkeitswerte, die bei 100% liegen, sich anders verteilen. Die Charakteristik der anderen Kategorien in der Histogrammdarstellung ist ähnlich und soll daher nicht dargestellt werden. Es zeigt sich, dass durch die Segmentierung der Eingangsbilder bessere Ergebnisse erzielt werden können. Weiter zeigt sich, dass der Klassifikatoransatz weiter verfeinert werden muss. Ob dies mit Hilfe eines optimierten Lernens geschieht, in dem die Lernbilder gezielter ausgewählt werden oder mit Hilfe von vor- oder nachgeschalteten separaten Filtern die Ergebnisse verbessert werden, muss untersucht werden.

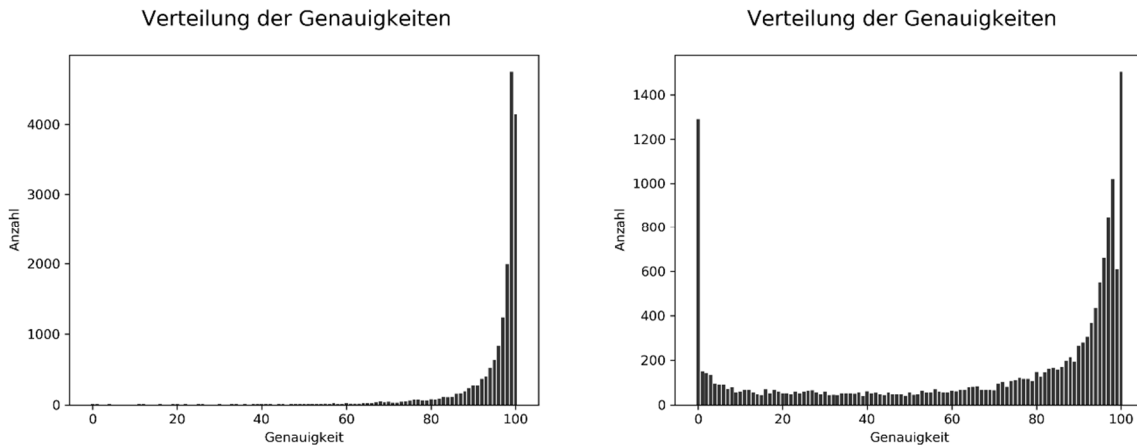


Abbildung 86: Darstellung der Genauigkeitshistogramme des ersten Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“

Ein Versuch, den Klassifikator mittels „Best of Three“- oder „Best of Five“-Lösung arbeiten zu lassen, brachte kein zufriedenstellendes Ergebnis. Mit einer „Best of Three“- oder „Best of Five“-Lösung ist gemeint, dass drei oder fünf verschiedene Netzwerke unabhängig voneinander klassifizieren und das Ergebnis dann mittels einfacher Mehrheit aus allen drei oder fünf Klassifikationsergebnissen ermittelt wird³⁶.

Dass diese Verbesserung keine Auswirkung auf die Genauigkeit hatte, liegt daran, dass die verschiedenen Netze ähnliche Klassifikationen aufweisen, so dass die Netzwerke auf identischen Strukturen ein identisches Klassifikationsergebnis aufweisen. Es könnte hier eventuell eine Verbesserung erzielt werden, indem verschiedene Netze auf verschiedene Probleme trainiert werden. Andere neuronale Netzwerkstrukturen, die auf diese oder ähnliche Weise die Daten verarbeiten sind viel leistungsfähiger, daher wurde dieser Ansatz nicht weiter verfolgt und schon bestehende Strukturen verwendet.

Würde dieser Klassifikator zum Jäten eingesetzt werden, würden sehr viele Beikräuter nicht erkannt werden. Auch müsste ein Algorithmus, der die klassifizierten Pixel auswertet, um die Beikräuter für die Vernichtung zu lokalisieren, die vielen False-Positiv-Klassifikationen (Beikraut, das als Karotte klassifiziert ist) korrigieren.

³⁶ Später stellte sich heraus, dass LeCun ein ähnliches Vorgehen in seinem Boosted LeNet-4 Netzwerk verwendet [LeCun, et al., 1995].

4.5.2 Convolutional-Neural-Network zur pixelweisen Pflanzenklassifikation

Das Convolutional-Neural-Network besitzt durch seinen Aufbau eine komplexere Möglichkeit, um Strukturen wiederzuerkennen. Für dieses Projekt wurde die Netzwerkstruktur des LeNet [LeCun, et al., 1995], [LeCun, et al., 1998] als Grundlage verwendet. Die Struktur setzt sich aus einer seriellen Verschaltung von parallel arbeitenden Convolution-, Pooling- und Dense-Schichten zusammen. Eines der ersten funktionierenden untersuchten CNN für den pixelweisen Klassifikator ist in der Abbildung 114 dargestellt und wurde in [Knoll, et al., 2018]³⁷ veröffentlicht.

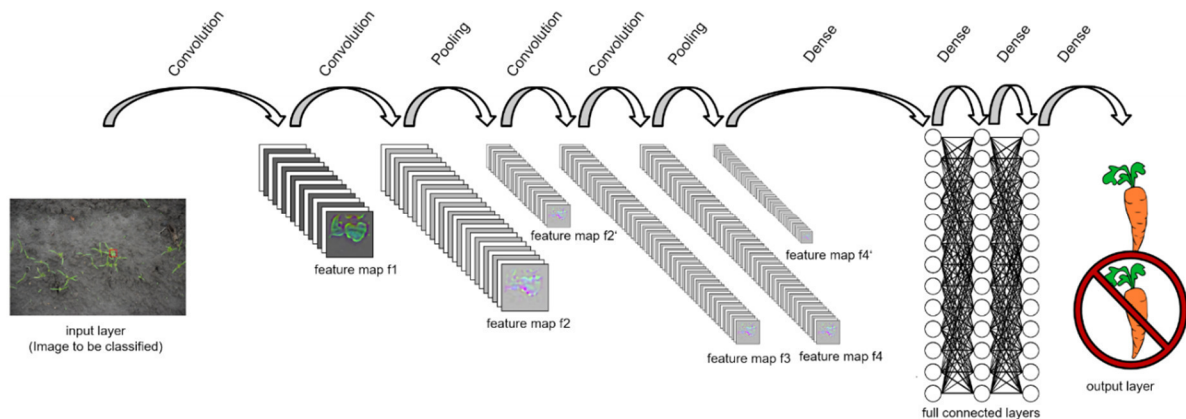


Abbildung 87: Struktur des Convolutional-Neural-Networks für die Pflanzenklassifikation [Knoll, et al., 2018]

Das Netzwerk verwendet als Eingangsbild ein RGB Bild mit der Größe von 101x101 Pixel. Dies ist als kleiner roter Kasten im linken Bild markiert. Auf dieses Bildelement wird eine Convolution-Schicht von 16 Features mit einer Größe von 5x5 angewendet. Auf die Ergebnisse der ersten Convolution-Schicht folgt eine weitere Convolution-Schicht mit einer Featuregröße von 7x7 Pixel und 32 Features. Nach einem Pooling werden erneut zwei Convolution-Schichten verwendet, wieder mit der Größe von 5x5 Pixeln und 7x7 Pixel, jedoch mit 32 und 64 Features. Nach einem weiteren Pooling folgen drei Dense-Schichten mit je 64 Neuronen. Alle Schichten verwenden dabei die ReLu-Schwellwertfunktion³⁸. Die folgende Ausgangsschicht besitzt zwei Neuronen für jede Klasse („Karotte“ und „Beikraut“). Es wurde hier auf weitere Pooling-Schichten zwischen den Convolution-Schichten verzichtet, da vermieden werden sollte, dass die Featuremaps zu klein werden für die letzten Schichten. Es sollten jedoch aus Gründen der Performance nicht noch größere Eingangsbilder gewählt werden³⁹. Die jeweils zweite Convolution-Schicht besitzt daher eine etwas größere Featuregröße, um das fehlende Pooling auszugleichen. Die Werte der einzelnen Featuregrößen und die Anzahl sind dabei rein experimentell bestimmt worden.

Das segmentierte Bildelement, das aus dem Gesamtbild extrahiert wird, besitzt hier wie bereits erwähnt die Größe von 101x101 Pixel im RGB Format. Wie bereits bei der Segmentierung in Abbildung 70 dargestellt⁴⁰, ist in der Grafik zu erkennen, dass die Pixel von 98% aller Pflanzen maximal 40 Pixel vom Hintergrund entfernt sind. Das Eingangsbild hat die Größe 101x101, damit zu jeder Seite vom Center Pixel ausgehend ein Übergang der Pflanze zu sehen ist. Der mittlere Pixel des segmentierten Bildes ist

³⁷ Die Idee des Autors, ein künstliches neuronales Netz zur Pflanzenklassifikation zu verwenden, wurde in der Abschlussarbeit von Sascha Pocizhoski untersucht [Pocizhoski, 2016].

³⁸ Die Rectifier Linear Unit ist in Kapitel 2.2.5.1 beschrieben und in Abbildung 17 dargestellt.

³⁹ Neben der Performance der Geschwindigkeit hat ein größeres Eingangsbild eine höhere Wahrscheinlichkeit, nicht nur eine Klasse von Pflanzen zu beinhalten. Dieses kann jedoch zu Fehlklassifikationen führen. Später dazu mehr.

⁴⁰ Abbildung 70 befindet sich auf Seite 87.

das zu klassifizierende Objekt und die umliegenden Pixel dienen als zusätzliche Informationen über das zu klassifizierende Objekt. Das CNN untersucht die Geometrie eines Objektes tiefgreifender als das Multilayer Perceptron. Dafür werden die Convolution-Schichten verwendet. Diese suchen zum Beispiel Kanten oder Flächen in einem Objekt. Diese gefundenen Merkmale werden in einer späteren Convolution-Schicht weiter zusammengefasst zu komplexeren Zusammenhängen. Die Dense-Schicht setzt alle gefunden Merkmale zu einer Klasse zusammen. In der Abbildung 88 ist links das Kamerabild dargestellt und rechts daneben das Ergebnisbild des Klassifikators.

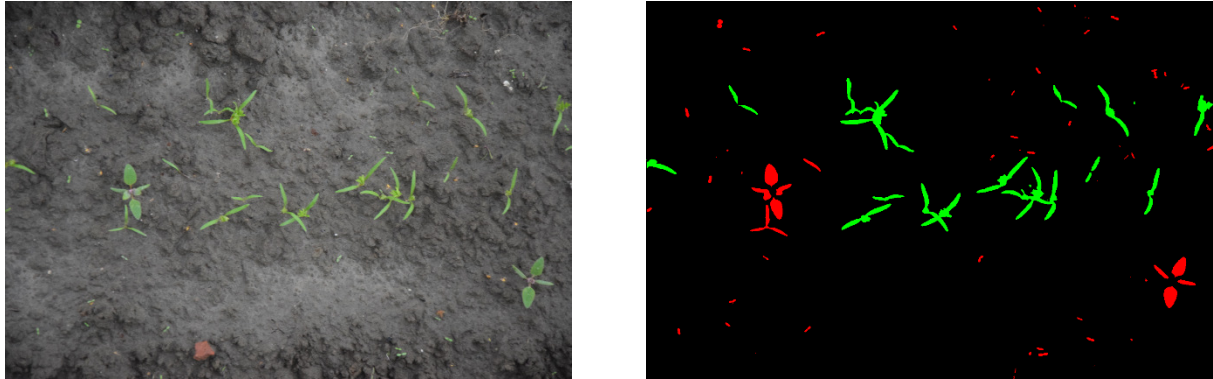


Abbildung 88: Kamerabild und Klassifikationsergebnis des Pixelweise-Klassifikators

Das Netzwerk wurde am Anfang mit dem Lasagne-Framework trainiert. Später wurde aus Performancegründen auf das neu aufkommende Tensorflow-Framework umgestiegen, da dieses schneller ist und bessere Lernerfolge in kürzerer Zeit generiert. Zum Training wurden insgesamt 100.000 Bildelemente aus den oben beschriebenen Trainingsbildern mit Hintergrund extrahiert. Die Bildelemente wurden dann in 10 Pakete für die Kreuzvalidierung aufgeteilt. Es handelt sich dabei um dieselben Bildelemente wie bei der Multilayer Perceptron-Untersuchung. Die verwendete Batchgröße wurde auf 64 festgelegt, innerhalb der Batches wurden die beiden Klassen immer alternierend hinterlegt. Die Trainingsdauer wurde dabei auf 100.000 Epochen festgelegt, da diese stichprobenartig nach circa 40.000 Epochen ihr Maximum erreichte. Durch die zusätzlichen Epochen soll eine Varianz der benötigten Epochen ausgeglichen werden und sichergestellt werden, dass kein Overfitting⁴¹ erfolgte.

Tabelle 26: Darstellung der Auswertungsergebnisse des CNNs über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung

Klasse	Genauigkeit	Precision	Recall	F1-Score
Karotte	0,97	0,97	0,97	0,97
Beikraut		0,95	0,96	0,97

Die Tabelle zeigt, dass eine hohe Pixelgenauigkeit von 0,97 in der globalen Genauigkeitsanalyse erreicht wird. Dies bedeutet, dass 97% aller Pixel in den gesamten 1075 Bildern richtig klassifiziert werden. Damit ein unterschiedliches Vorkommen der Klassen nicht das Ergebnis beeinflusst, wurde hier eine einfache lineare Skalierung⁴² zwischen dem Auftreten der Pixel vorgenommen. Die nicht skalierten Ergebnisse weichen jedoch auch nur maximal 1% ab. Auch die Precision, der Recall und der F1-Score zeigen sehr gute Ergebnisse. Zur besseren Betrachtung der Ergebnisse soll auch hier wieder eine Analyse der einzelnen Pflanzen erfolgen. Die Genauigkeitswerte der einzelnen Kategorien (lokale Genauigkeitsanalyse) sind in der Tabelle 27 aufgelistet. Auf die Berechnung der Precision und des Recalls soll hier verzichtet werden. Innerhalb einer Pflanze existiert kein Pixel mit einem True-Negativ-

⁴¹ auswendig lernen

⁴² $\text{Skalierungsfaktor} = \frac{\text{TruePositiv} + \text{FalseNegativ}}{\text{TrueNegativ} + \text{FalsePositiv}}$

und keinem False-Positiv-Wert, da immer aus der jeweiligen Pflanze auf die Auswertung geschaut wird. Daher wäre die Precision immer 1 und der Recall-Wert identisch mit der Genauigkeit. In der Klammer neben der Genauigkeit ist die Standardabweichung angegeben und in der letzten Spalte steht die Anzahl der ausgewerteten Pflanzen.

Tabelle 27: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des CNN-Klassifikators

Kategorie	Genauigkeit (SD)	Anzahl gesamt	Genauigkeit ≥0,90	Genauigkeit ≥0,95
gesamt	0,92 (0,22)	32.983	86%	84%
Karotte gesamt	0,98 (0,10)	17.867	95%	93%
Beikraut gesamt	0,86 (0,30)	15.116	78%	73%
Karotte ohne 3. Blatt	0,98 (0,10)	13.247	95%	94%
Karotte mit mind. 3 Blatt	0,97 (0,11)	4.620	93%	88%
Beikraut klein	0,85 (0,30)	13.786	77%	73%
Beikraut mittel	0,86 (0,27)	581	75%	70%
Beikraut Knöterich	0,98 (0,06)	422	95%	90%
Beikraut groß	0,90 (0,22)	327	81%	68%

Nach der Tabelle werden im Durchschnitt 92% der Pixel einer Pflanze richtig klassifiziert. Wie die Standardabweichungen jedoch zeigen, sind die mittleren Genauigkeitswerte so nicht verwertbar. Daher soll hier zusätzlich der prozentuale Anteil der Pflanzen angegeben werden, die über einer Genauigkeit von 0,9 und 0,95 liegen. Dies bedeutet, dass 86% aller Pflanzen eine Genauigkeit von mindestens 0,9 aufweisen und immer noch 84% aller Pflanzen eine Genauigkeit von 0,95. Ähnlich wie bei dem MLP-Netzwerk ergibt sich auch hier eine Charakteristik des Histogramms, das eine Nicht-Normalverteilung der Genauigkeiten der einzelnen Pflanzen aufzeigt, wie in Abbildung 89 dargestellt.

Verteilung der Genauigkeiten

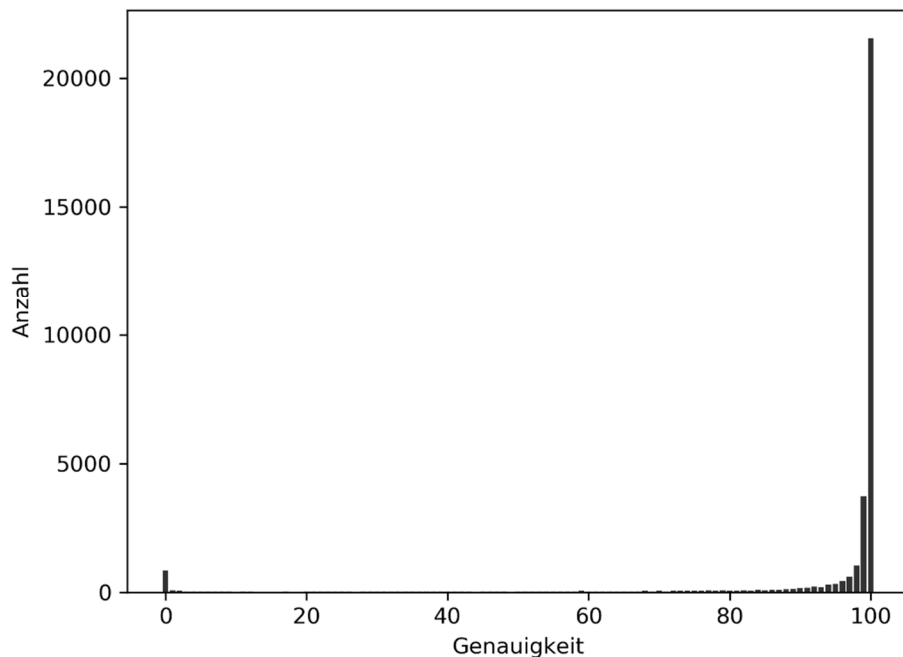


Abbildung 89: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des ersten CNN-Klassifikators

In dem Histogramm sind sowohl die Karotten als auch die Beikräuter dargestellt. Das Histogramm zeigt, dass bei 77% aller Pflanzen eine Genauigkeit von mindestens 0,99 aller Pixel erreicht wurde. Ab einer Pixelgenauigkeit innerhalb der Pflanze von mindestens 0,95 sind 84% aller Pflanzen der insgesamt 32.983 Pflanzen inbegriffen. Jedoch ergibt sich auch ein Bereich wie beim MLP-Netzwerk, in dem Pflanzen komplett falsch klassifiziert werden. So weisen 2,6% aller Pflanzen eine Pixelgenauigkeit von unter 0,01 auf und 3,5% aller Pflanzen eine Pixelgenauigkeit von weniger als 0,1.

Die Auswertung zeigt daher ein extremes Verhalten. Viele Pflanzen werden komplett richtig klassifiziert oder zeigen nur wenige falsche Pixel. Außerdem werden einige Pflanzen komplett falsch klassifiziert. Die Betrachtung der einzelnen Klassen des Histogramms zeigt ein noch eindeutigeres Bild.

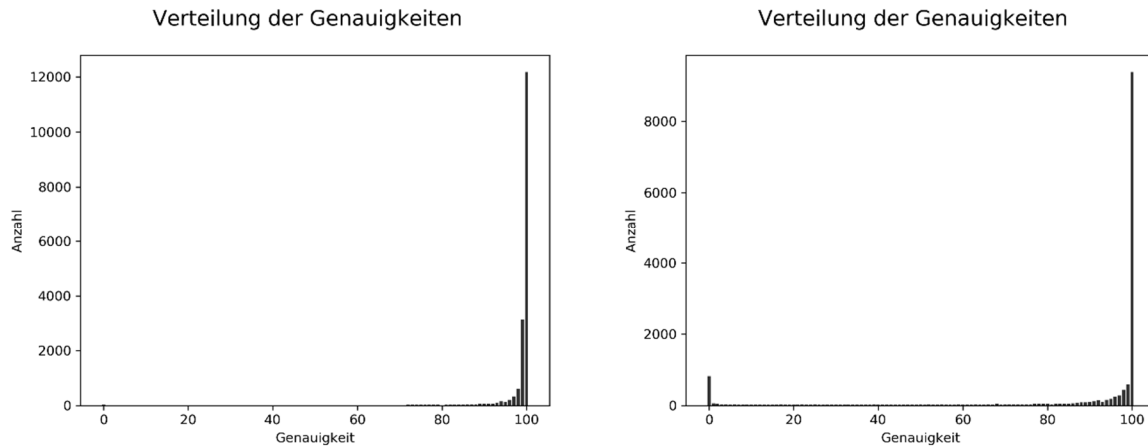


Abbildung 90: Darstellung der Genauigkeitshistogramme des CNN-Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“

Das rechte Histogramm zeigt die Verteilung der Pixelgenauigkeiten in den Karottenpflanzen. Hier haben 95% aller Karotten mindestens eine Pixelgenauigkeit von 0,9. Über eine Pixelgenauigkeit innerhalb einer Pflanze von 0,99 kommen 86% aller Karotten. Beim Beikraut sind fast alle Fälle der Genauigkeiten von 0 enthalten. Aber auch hier weist der obere Bereich der Skala die meisten Fälle auf. Eine Genauigkeit innerhalb der Pflanze von mindestens 0,9 haben 77% aller Beikräuter und eine Genauigkeit von über 0,99 immer noch 66% aller Beikräuter.

Anders als bei den MLP-Netzwerken weisen die extrem schlechten Fälle auf einen strukturierten und reproduzierbaren Fehler hin und nicht auf ein zufälliges Auftreten. Bei der Betrachtung eines klassifizierten Bildes wie in Abbildung 91 ist der Fehler ersichtlich. Die grünen Pixel stellen Pixel von Karotten dar, die roten sind Beikrautpixel. Auf die Darstellung des korrekten Labels soll hier verzichtet werden. Der Fall mit den extrem schlechten Fällen befindet sich im blauen Kreis. Die drei kleinen Pflanzen neben den Karotten sind als Karotten klassifiziert worden, obwohl sie Beikraut sind. Die Ursache besteht vermutlich darin, dass, sobald eine kleine Beikrautpflanze zu nah an einer größeren Karottenpflanze liegt und der Klassifikator beides in seinem Eingangsbild registriert, er das Beikraut als Karotte klassifiziert. Eine genauere Problembetrachtung soll später erfolgen.

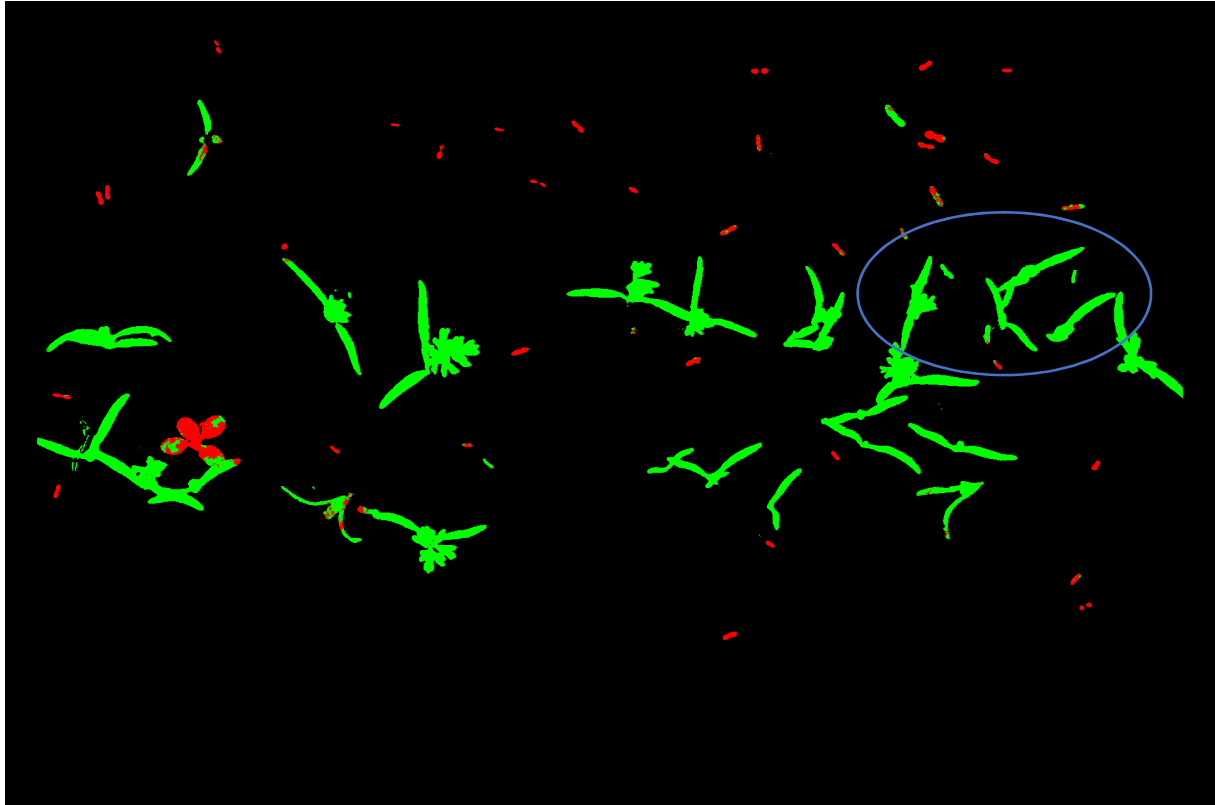


Abbildung 91: Ergebnis eines CNN-Klassifikationsergebnisses; Grün: überwiegend Karottenpixel; Rot: überwiegend Beikrautpixel; Blauer Kreis: stellt den Fall der extremen Klassifikationsergebnisse dar

Die erste Netzwerkoptimierung sah vor, zwei Convolution-Schichten zu entfernen. Das optimierte Netzwerk ist in der Abbildung 92 dargestellt und verwendet immer noch ein Eingangsbild von 101x101 Pixel im RGB Format. Auf das Eingangsbild wirkt eine Convolution-Schicht aus 16 Features mit der Größe von 5x5 Pixeln. Direkt danach wird eine Pooling Schicht durchgeführt und erneut eine Convolution-Schicht aus 32 Features mit der Größe von 7x7 Pixeln. Nach einem erneuten Pooling bilden drei Dense-Schichten mit je 64 Neuronen die Verbindung zu den beiden Ausgangsneuronen. Es wurde hier lediglich die Anzahl der Convolution-Schichten verändert, damit genau dieser Aspekt untersucht werden konnte.

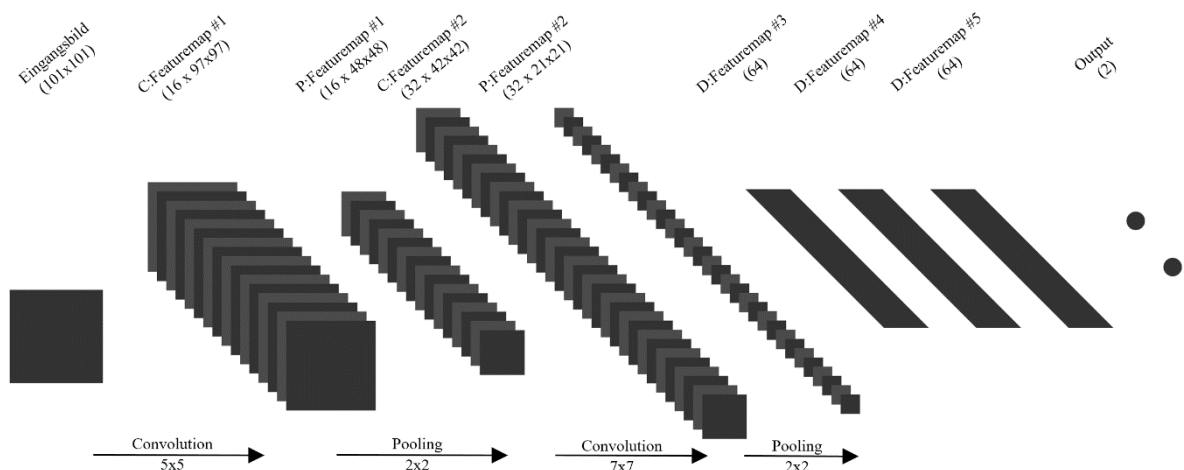


Abbildung 92: Struktur des optimierten Convolutional-Neural-Networks für die Pflanzenklassifikation

Auch mit dieser Netzwerkstruktur wurden die 1075 Bilder klassifiziert. Die Ergebnisse der Klassifikation weisen dabei einen ähnlichen Charakter auf wie das erste Netzwerk. Die statistischen Größen für die globale Genauigkeitsanalyse sind in der Tabelle 28 abgebildet.

Tabelle 28: Darstellung der Auswertungsergebnisse des optimierten CNNs über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung

Klasse	Genauigkeit	Precision	Recall	F1-Score
Karotte	0,96	0,95	0,97	0,96
Beikraut		0,97	0,95	0,96

Die Werte sind nahezu identisch mit denen des ursprünglichen CNN-Klassifikators. Der kleine Unterschied kann eventuell in der unterschiedlichen Initialisierung der Gewichte der beiden Netzwerke ihren Ursprung haben. Die Betrachtung der Genauigkeiten der einzelnen Pflanzen weist dabei größere Unterschiede auf, wie Tabelle 29 mit der lokalen Genauigkeitsanalyse zeigt. Die Unterschiede sind jedoch so gering, dass mit dem optimierten Netzwerk weiter gearbeitet werden könnte.

Tabelle 29: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des CNN-Klassifikators

Kategorie	Genauigkeit ursprünglich (SD)	Genauigkeit optimiert (SD)	Anzahl	Genauigkeit optimiert ≥0,90	Genauigkeit optimiert ≥0,95
gesamt	0,92 (0,22)	0,91 (0,23)	32.983	85%	81%
Karotte gesamt	0,98 (0,10)	0,97 (0,12)	17.867	92%	88%
Beikraut gesamt	0,86 (0,30)	0,85 (0,30)	15.116	76%	72%
Karotte ohne 3. Blatt	0,98 (0,10)	0,97 (0,12)	13.247	93%	90%
Karotte mit mind. 3 Blatt	0,97 (0,11)	0,96 (0,12)	4.620	90%	84%
Beikraut klein	0,85 (0,30)	0,85 (0,31)	13.786	76%	72%
Beikraut mittel	0,86 (0,27)	0,82 (0,29)	581	64%	57%
Beikraut Knöterich	0,98 (0,06)	0,96 (0,08)	422	89%	81%
Beikraut groß	0,90 (0,22)	0,87 (0,25)	327	79%	65%

Auch die Darstellung des Histogramms zeigt, dass die Ergebnisse ähnlich sind, wie bei der ursprünglichen Netzwerkgröße. Das Histogramm, das die Genauigkeitsverteilung aller Pflanzen zeigt, ist in Abbildung 95 dargestellt. Das Histogramm, das nur die Verteilung für die Karotten zeigt, ist links und das Histogramm nur für die Beikräuter ist rechts in Abbildung 96 dargestellt.

Verteilung der Genauigkeiten

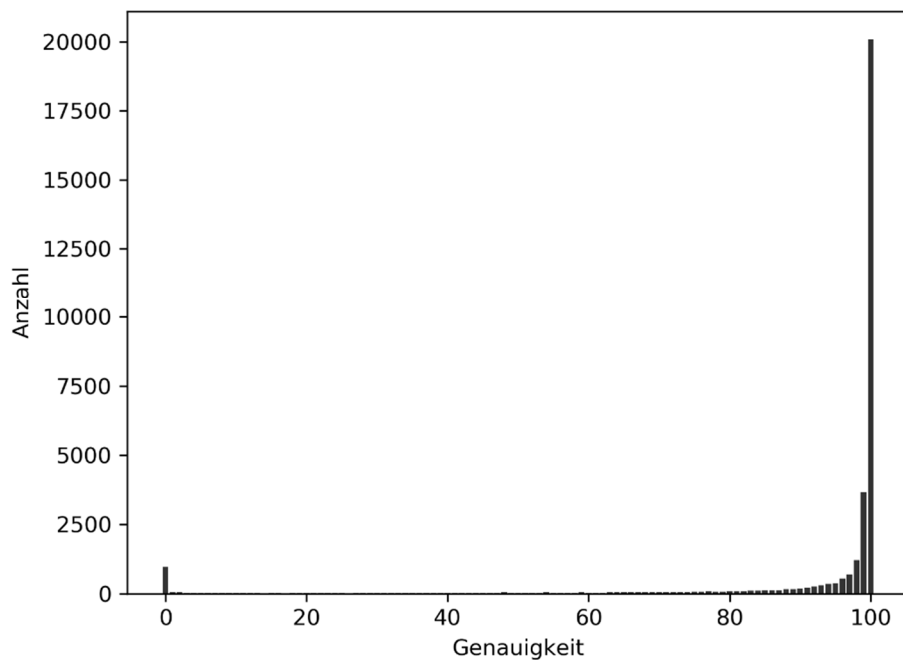


Abbildung 93: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des verkleinerten CNN-Klassifikators.

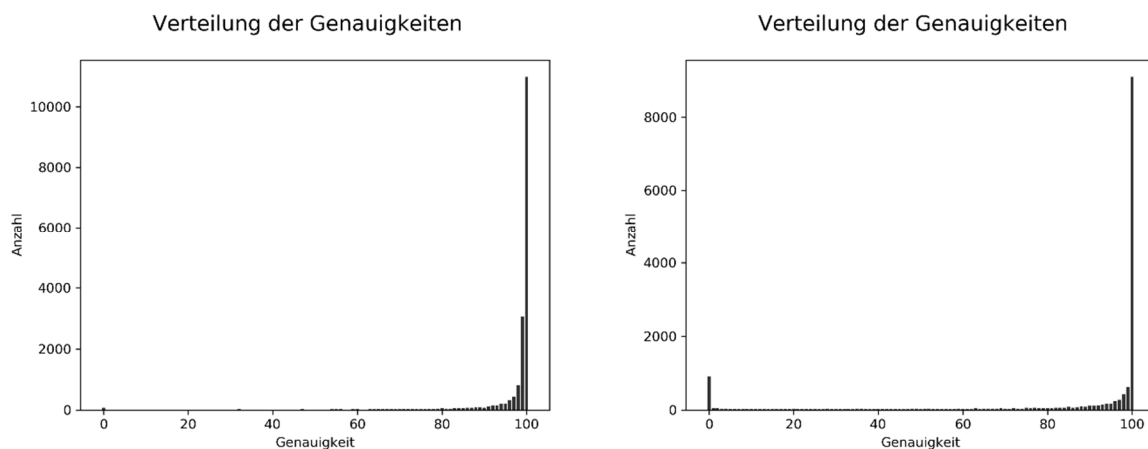


Abbildung 94: Darstellung der Genauigkeitshistogramme des verkleinerten CNN-Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“

Der Klassifikator ist besser als das vorgestellte MLP, er besitzt jedoch noch einige False-Positiv-Klassifikationen (Beikraut, das als Karotte klassifiziert ist); für die Werte wurde auch eine Theorie beschrieben, die zur Vermeidung dieses Fehlers eingesetzt werden könnte. Das oben dargestellte und untersuchte CNN erfüllte zwar die Anforderungen an die Genauigkeit, jedoch ist neben der Genauigkeit auch die Geschwindigkeit für die Klassifikation von Bedeutung. Die Geschwindigkeit soll erst später optimiert werden, wenn eine Klassifikatorarchitektur ausgewählt wurde.

4.5.3 GoogLeNet

Eine weitere Entwicklung des LeNets ist das GoogLeNet, das in [Szegedy, et al., 2015] präsentiert wurde. In dieser Architektur wird versucht, tiefere Netzwerke zu erzeugen. Das GoogLeNet verwendet in seiner Architektur Module, die sich aus Convolution- und Pooling-Schichten zusammensetzen. Ein erster Ansatz mit dem GoogLeNet bestand darin, verschiedene Featuregrößen parallel in einer Schicht zu verarbeiten. Die verwendeten Module werden als Inceptionmodule bezeichnet. Das Inceptionmodul der ersten Version des GoogLeNet ist in Abbildung 95 dargestellt.

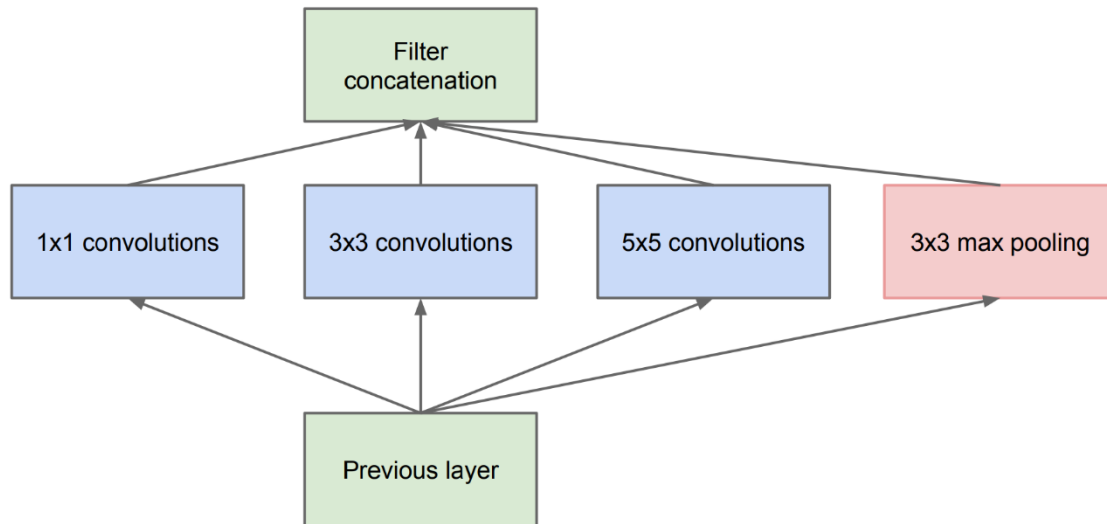


Abbildung 95: Inceptionmodul des GoogLeNet Version 1 [Szegedy, et al., 2015]

Dieses Modul verwendet drei parallel arbeitende Convolutions mit den Featuregrößen 1x1, 3x3 und 5x5. Außerdem wird ein Maxpooling der Größe 3x3 auf das Ergebnis der vorherigen Schicht angewendet. Der Aufbau resultiert aus der Idee, die passenden Featuregrößen für jede auftretende Problemstellung bereitzustellen, anstatt sich für eine Größe entscheiden zu müssen. Das Pooling soll denselben positiven Effekt erzeugen wie bei der LeNet Struktur. Außerdem ermöglicht es der nachfolgenden Schicht, auf die zusammengefassten Informationen der vorherigen Schicht zuzugreifen [Szegedy, et al., 2015]. Dies bedeutet, dass die Informationen der jeweiligen Schicht lediglich über das Pooling zusammengefasst werden, aber ansonsten unverändert tieferen Schichten zur Verfügung stehen. Die Convolution-Schicht mit der Größe von 1x1 kann als Dimensionsverringerung aufgefasst werden. Die Schicht fasst die einzelnen Ergebnisse der vorherigen Schicht zusammen, so dass die nachfolgenden Schichten nicht mehr auf die volle Anzahl der einzelnen Featuremaps zugreifen müssen, sondern Zugriff haben auf eine Zusammenfassung mit verringerter Dimension. Genau wie beim Pooling werden so Informationen von der vorherigen an die nächste Schicht weitergeben, ohne dass eine direkte Veränderung durch eine komplexere Featureanwendung herbeigeführt wurde. Diese Methode wird in der Weiterentwicklung des Inceptionmoduls ebenfalls für die anderen Features und das Pooling angewendet. Damit soll unter anderem der Rechenaufwand verringert werden [Szegedy, et al., 2015]. Das erweiterte Modul ist in Abbildung 96 dargestellt.

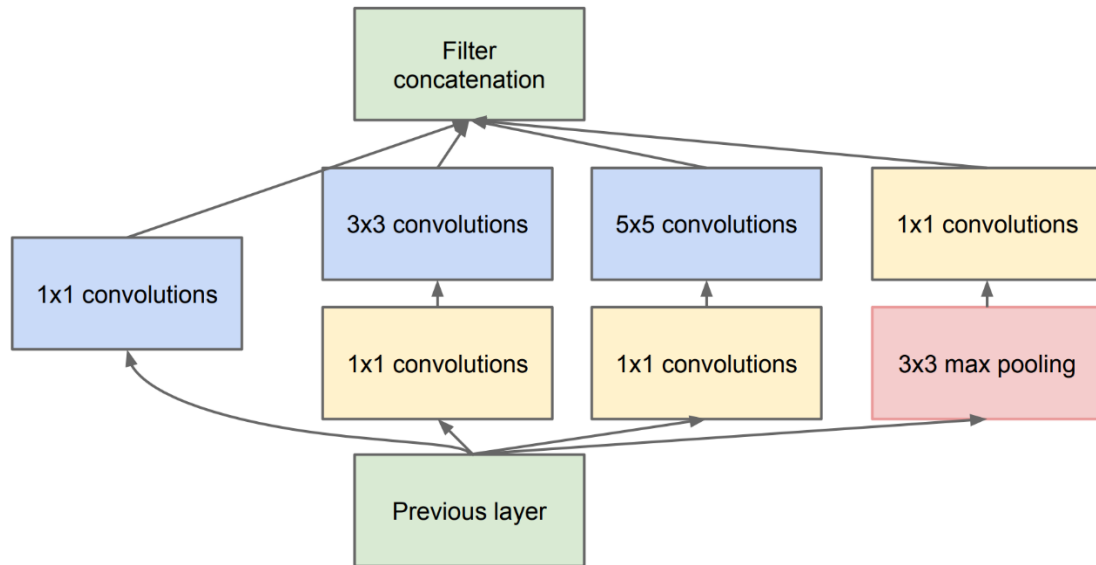


Abbildung 96: Weiterentwicklung des Inceptionmoduls mit Dimensionsreduktion [Szegedy, et al., 2015]

Diese Module werden anschließend zu einem komplexen Aufbau verschaltet. Der Aufbau ist in [Szegedy, et al., 2015] abgebildet. Es werden hier neun Inceptionmodule hintereinander geschaltet. Damit ergibt sich eine Tiefe von 27 Schichten. Damit das Netzwerk die Tiefe beim Lernen nutzen kann, wurden zusätzliche Hilfsausgänge geschaffen, die nach je drei Inceptionmodulen das Netzwerk verlassen. Über die Hilfsausgänge soll verhindert werden, dass der mittlere Teil des Netzwerkes nicht eingeht. Das GoogLeNet wird dabei von Google vortrainiert, lediglich die Dense-Struktur am Ende wird vom Benutzer auf das jeweilige Problem angepasst. Hierfür wurde eine Eingangsbildgröße von 101x101 Pixeln gewählt.

Die Verwendung des GoogLeNet ist rechenintensiver als die anderen Klassifikatoren. So kann die Auswertung eines gesamten Bildes zwischen 30 und 60 Minuten benötigen. Der GoogLeNet Klassifikator ist auch nicht für diese Art der einzelnen Pixelklassifikation ausgelegt, sondern soll eher für eine Klassifikation wie bei der ImageNet Large Scale Visual Recognition Competition (ILSVRC) mehrere 1000 Klassen wiederfinden [Wartala, 2018]. Damit die 1075 Testbilder in einer vernünftigen Zeit ausgewertet werden können, wurde hier von jedem Bild nur jeder zehnte Pflanzenpixel ausgewertet. Für die Auswertung der circa 8,5 Millionen Pflanzenpixel wurden fast vier Tage benötigt. Die Auswertung jedes zehnten Pflanzenpixels in den 1075 Bildern für die globale Genauigkeitsanalyse sind in Tabelle 30 dargestellt. Zum Training wurde eine Batchgröße von 64 und 100.000 Epochen verwendet. Das Netz erreichte vor den 100.000 Epochen ein konstantes maximales Niveau der Genauigkeit.

Tabelle 30: Darstellung der Auswertungsergebnisse des GoogLeNet über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung

Klasse	Genauigkeit	Precision	Recall	F1-Score
Karotte	0,95	0,94	0,97	0,95
Beikraut		0,97	0,94	0,95

Die Ergebnisse weisen ähnliche Erfolge auf, wie bei der LeNet Struktur, die ebenfalls für alle Größen im oberen neunzig Prozent Bereich liegen. Werden die Ergebnisse für die lokale Genauigkeitsanalyse wieder auf die einzelnen Pflanzen bezogen, ergibt sich die Tabelle 31.

Tabelle 31: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien der GoogLeNet Klassifikators

Kategorie	Genauigkeit (SD)	Anzahl	Genauigkeit ≥0,90	Genauigkeit ≥0,95
Gesamt	0,93 (0,22)	32.983	90%	87%
Karotte gesamt	0,98 (0,08)	17.867	95%	92%
Beikraut gesamt	0,88 (0,29)	15.116	84%	82%
Karotte ohne 3. Blatt	0,98 (0,08)	13.247	96%	93%
Karotte mit mind. 3. Blatt	0,96 (0,10)	4.620	91%	84%
Beikraut klein	0,88 (0,30)	13.786	84%	83%
Beikraut mittel	0,92 (0,21)	581	85%	81%
Beikraut Knöterich	0,97 (0,10)	422	92%	88%
Beikraut Groß	0,81 (0,26)	327	58%	39%

Die Tabelle zeigt, dass innerhalb einer Karotte 98% der Pixel im Mittel richtig klassifiziert werden. Beim Beikraut sind es je Pflanze lediglich 88% der Pixel. Die große Standardabweichung, insbesondere bei den Beikräutern, zeigt jedoch erneut, dass hier nicht von einer normalen Verteilung ausgegangen werden kann und die Werte so nicht verwertbar sind. Daher wird auch hier wieder eine Histogrammanalyse erzeugt. Die nachfolgenden Histogramme in diesem Kapitel zeigen, von wie vielen Pflanzen (Ordinate), wie viel Prozent der sich in der Pflanze befindenden Pixel (Abszisse) richtig klassifiziert wurden.

Verteilung der Genauigkeiten

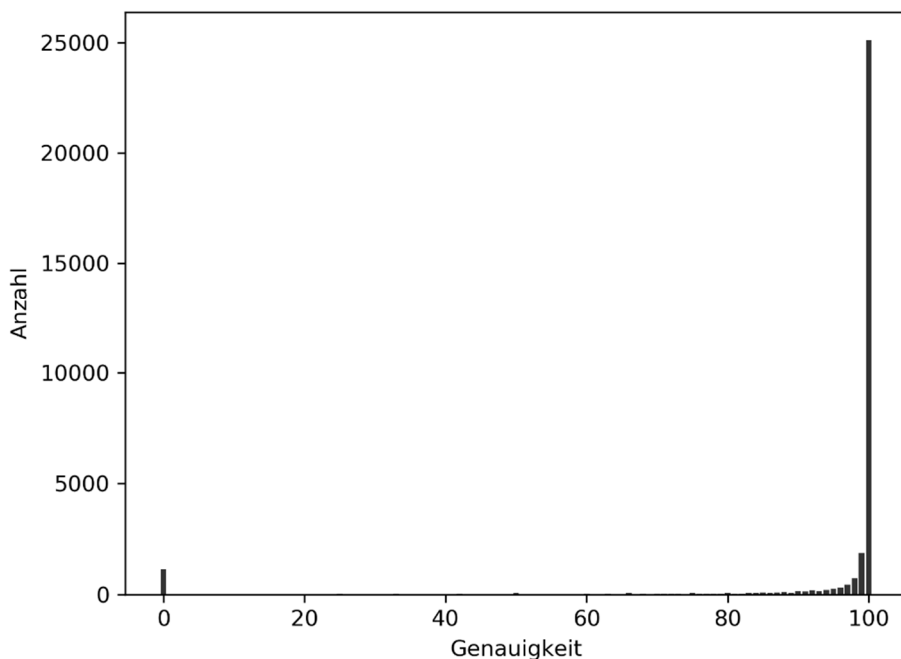


Abbildung 97: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des GoogLeNet-Klassifikators

Das Histogramm für alle Pflanzenarten zeigt ein ähnliches Verhalten wie das der LeNet Struktur. Auch hier bilden sich erneut zwei Extrema ab. Die 32.983 Pflanzen werden entweder komplett falsch oder fast komplett richtig klassifiziert. Hier weisen 82% aller Pflanzen eine Genauigkeit von mindestens 0,99 auf und 90% besitzen eine Genauigkeit von mindestens 0,9. Dabei haben 4% der Pflanzen eine Genauigkeit von unter 0,01. Die Betrachtung der einzelnen Klassen („Karotte“ und „Beikraut“) ist in

Abbildung 98 dargestellt. Das linke Histogramm bildet die Verteilung über alle Karotten und das rechte Histogramm die Verteilung über alle Beikräuter ab. Auch hier zeigt sich ein Verhalten wie bei der LeNet Struktur.

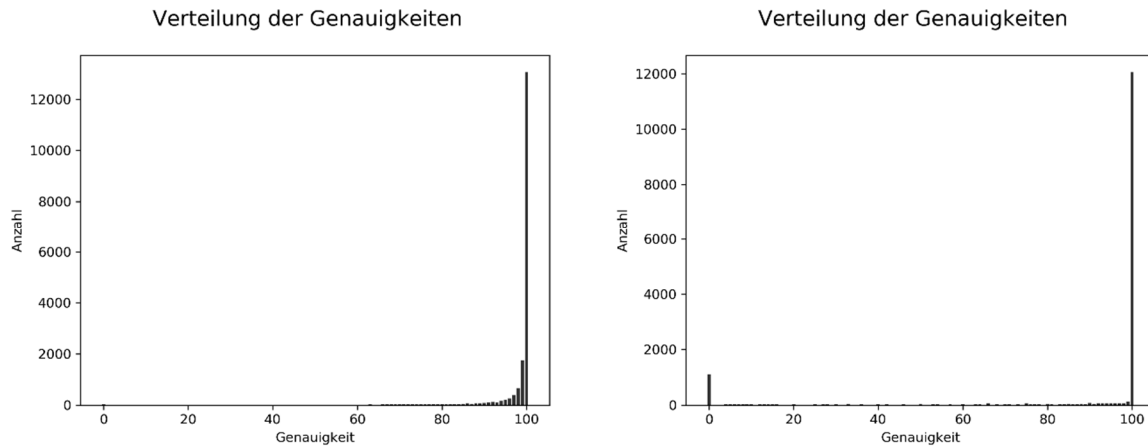


Abbildung 98: Darstellung der Genauigkeitshistogramme des GoogLeNet-Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“

Es zeigt sich, dass 94% der Karotten mindestens mit einer Pixelgenauigkeit von 0,9 richtig klassifiziert werden. Eine Pixelgenauigkeit von unter 0,8 weisen nur 3% aller Karotten auf. Das Beikraut bildet hingegen die zwei Extrema ab. Dabei werden circa 7% der Beikrautpflanzen komplett falsch (Pixelgenauigkeit von unter 0,01) klassifiziert und 84% aller Beikräuter werden mit einer Genauigkeit von mindestens 0,9 richtig klassifiziert. Auch hier ist das negative Extrem der Beikräuter durch die Fälle begründet, in denen eine große Karotte in der Nähe eines kleinen Beikrauts wächst.

4.5.4 Alex-Netz

Ein weiteres Netzwerk, das das LeNet als Basis verwendet, ist das Alex-Netz und wurde in [Krizhevsky, et al., 2012] präsentiert. Die Netzwerkstruktur ist in Abbildung 99 dargestellt. Das Netzwerk verwendet ein Eingangsbild von 224x224 Pixel im RGB Format. Es folgt eine Convolution-Schicht von 96 Features mit der Featuregröße von 11x11. Die erste Convolution-Schicht verwendet dabei eine Schrittweite (Stride) von vier Pixeln. Auf die Featuremap wird ein Pooling angewendet, hier jedoch mit Überlappung (Schrittweite ein Pixel). Außerdem wird eine Normalisierung der Ergebnisse durchgeführt (später mehr dazu in diesem Kapitel). Es folgt eine weitere Convolution-Schicht mit 256 Features der Größe 5x5. Nach einem erneuten Normalisierungsvorgang und Pooling mit der Schrittweite von zwei Pixel folgen direkt drei Convolution-Schichten. Diese drei Schichten sind direkt miteinander verbunden, verwenden also keine Normalisierung oder Pooling untereinander. Die erste der drei Convolution-Schichten verwendet 384 Features mit 3x3 Pixeln, es folgen weitere 384 Features mit 3x3 Pixel Größe und abschließend 256 Features ebenfalls mit der Größe von 3x3 Pixeln. Das Netzwerk bildet zum Schluss zwei Dense-Schichten mit je 4096 Neuronen. Der Übergang von der letzten Convolution- zur ersten Dense-Schicht bildet dabei wieder eine Normalisierung und ein Pooling. Der ursprüngliche Ausgang bildet dann eine Softmax-Schicht, die 1000 Klassen repräsentiert. Das Netzwerk verwendet die ReLu-Funktion als Aktivierungsfunktion der einzelnen Schichten [Krizhevsky, et al., 2012].

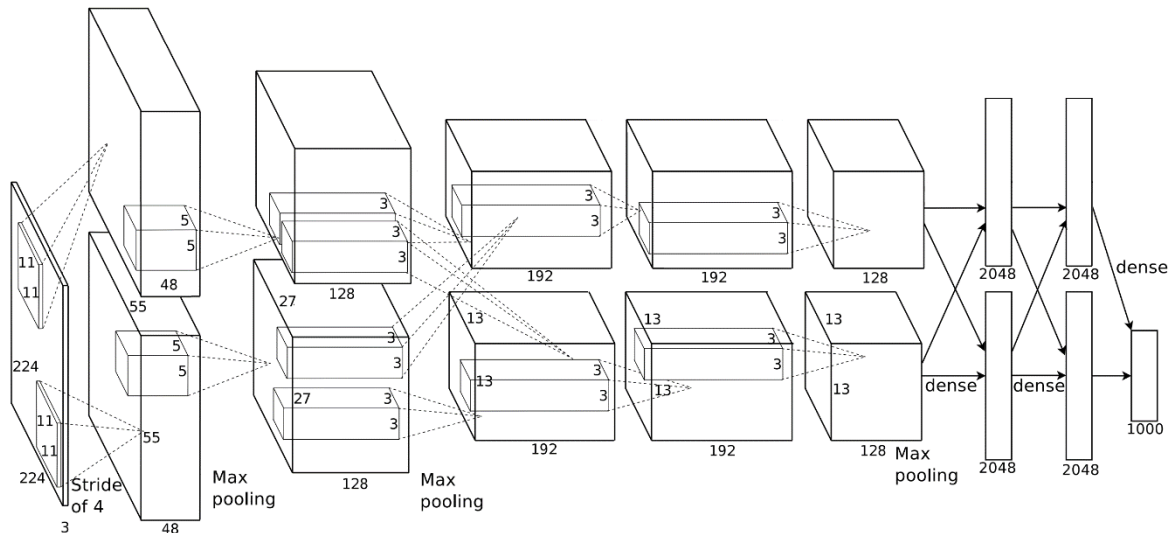


Abbildung 99: Aufbau der AlexNet Struktur [Krizhevsky, et al., 2012]⁴³

Die Abbildung 99 stellt das Netz wie in dem Paper vorgestellt dar. Dabei haben die Autoren das AlexNet so aufgeteilt, dass es auf zwei Grafikkarten parallel verwendet werden kann. So wird der obere Zweig auf einer und der untere Zweig auf einer zweiten Grafikkarte berechnet. Der Datenaustausch zwischen den beiden Grafikkarten wurde dabei auf ein Minimum reduziert, um Berechnungszeit zu sparen. Die Autoren waren zu diesem Vorgehen gezwungen, da die verwendete GPU (GTX 580) lediglich 3GB Arbeitsspeicher benutzte und dieser Speicherplatz nicht ausreichte für die Berechnung [Krizhevsky, et al., 2012]. Die hier in dieser Arbeit verwendete Grafikkarte (GeForce GTX 1080 Ti) bietet genügend Ressourcen, um dies auf einer Grafikkarte zu berechnen, daher musste die Aufteilung nicht verwendet werden.

Das AlexNet verwendet Normalisierungsschichten. Diese Schichten verwenden dabei die Formel

$$\psi_{x,y}^i = \frac{\psi_{x,y}^i}{\left(k + \alpha \cdot \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (\psi_{x,y}^j)^2\right)^\beta} \quad (105)$$

mit

$\psi_{x,y}^i$	Pixelwert an der Stelle des Ausgangsbildes ψ' an der Koordiante x und y , sowie der Dimension i
$\psi_{x,y}^i$	Pixelwert an der Stelle des Eingangsbildes ψ an der Koordinate x und y , sowie der Dimension i
k, n, α, β	sind Hyperparameter
N	Anzahl aller Features in der Schicht.

Die Hyperparameter sind in der Veröffentlichung [Krizhevsky, et al., 2012] mit den Werten $k = 2, n = 5, \alpha = 10^{-4}$ und $\beta = 0,75$ angeben. Die Normalisierung soll einen Wettbewerb zwischen den verschiedenen Features erzeugen und eindeutigere Aktivierungen bewirken⁴⁴.

Für die Untersuchung über die 1075 Bilder wurde das AlexNet mit der Bildgröße von 101x101 Pixeln trainiert. Dabei wurde das Netz ebenfalls mit 100.000 Epochen trainiert und es wurde eine Batchgröße

⁴³ Das Originalbild in der Veröffentlichung ist unvollständig (abgeschnitten) und wurde hier wieder vervollständigt

⁴⁴ Die Verwendung der Normalisierung wird unter anderem in den Stanford-Kursen (zum Beispiel in: [github, 2019]) diskutiert. Die vertretene Meinung ist, dass die Art der Normalisierung nur einen kleinen positiven Einfluss auf die Ergebnisse hat, jedoch zum Beispiel durch die Verwendung der Dropout-Methode mittlerweile abgelöst wurde.

von 64 verwendet. Auch bei diesem Netz wurde das Genauigkeitsmaximum vor den 100.000 Epochen erreicht. Die Ergebnisse für die globale Genauigkeitsanalyse, in denen alle Pixel aller Bilder zusammengefasst werden stehen in Tabelle 32.

Tabelle 32: Darstellung der Auswertungsergebnisse des AlexNet über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung

Klasse	Genauigkeit	Precision	Recall	F1-Score
Karotte	0,95	0,94	0,96	0,95
Beikraut		0,96	0,94	0,95

Auch das AlexNet zeigt ähnliche Klassifikationsergebnisse wie die vorherigen Klassifikatoren. Die Ergebnisse liegen, wie die Tabelle zeigt, im mittleren Neunzigprozent Bereich. Die Ergebnisse der einzelnen Pflanzen mit der Untersuchung der lokalen Genauigkeitsanalyse stehen in der Tabelle 33.

Tabelle 33: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des AlexNet Klassifikators

Kategorie	Genauigkeit (SD)	Anzahl	Genauigkeit ≥0,90	Genauigkeit ≥0,95
Gesamt	0,93 (0,21)	32.983	88%	85%
Karotte gesamt	0,97 (0,10)	17.867	93%	90%
Beikraut gesamt	0,88 (0,28)	15.116	83%	80%
Karotte ohne 3. Blatt	0,98 (0,10)	13.247	95%	92%
Karotte mit mind. 3 Blatt	0,96 (0,10)	4.620	88%	82%
Beikraut klein	0,88 (0,29)	13.786	84%	82%
Beikraut mittel	0,89 (0,24)	581	78%	70%
Beikraut Knöterich	0,94 (0,14)	422	85%	74%
Beikraut groß	0,81 (0,27)	327	60%	41%

Die lokalen Genauigkeiten liegen ebenfalls in derselben Größenordnung wie bei den anderen Klassifikatoren. Auch die Standardabweichung lässt wieder auf einen nicht normalverteilten Graphen schließen. Es soll auch hier wieder ein Histogramm die Verteilung der Genauigkeiten der einzelnen Pflanzen aufzeigen.

Verteilung der Genauigkeiten

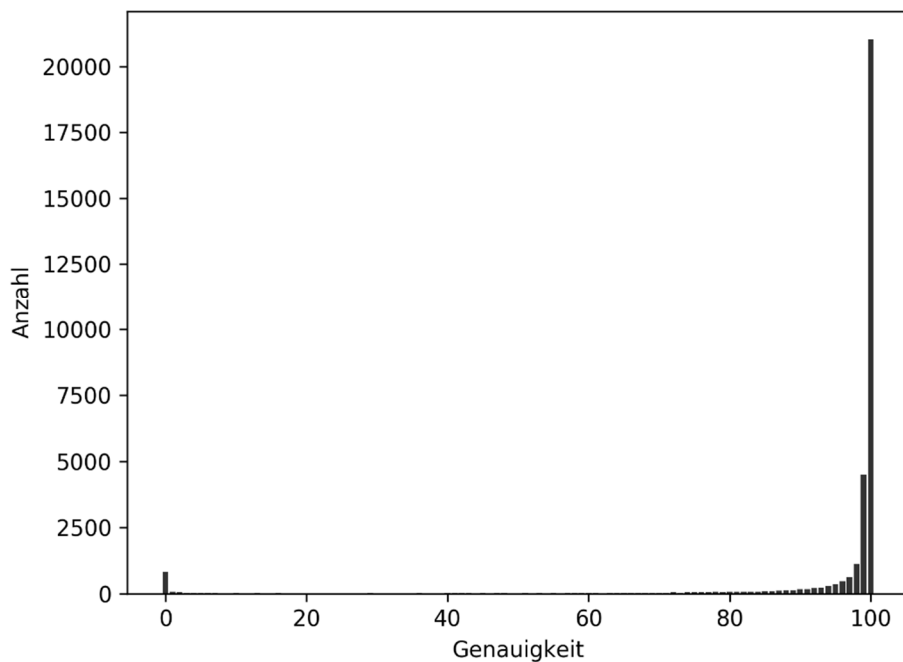


Abbildung 100: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des AlexNet-Klassifikators

Die Abbildung 100 zeigt denselben Charakter wie die bisherigen untersuchten Klassifikatoren. Von den 32.983 Pflanzen besitzen 77% eine Genauigkeit von mindestens 0,99. Eine Genauigkeit von mindestens 0,90 besitzen 88% aller Pflanzen. Unter einer Genauigkeit von 0,1 liegen 3,5% und unter einer Genauigkeit von 0,01 immer noch 2,5% aller Pflanzen. Die Betrachtung der einzelnen Klassen („Karotte“ und „Beikraut“) in Abbildung 101 zeigt das bekannte Verhalten der anderen Klassifikatoren. Hier besitzen 93% aller Karotten eine Genauigkeit von mindestens 0,90 und eine Genauigkeit von mindestens 0,99 immer noch 79% aller Karottenpflanzen. Die Beikräuter spalten sich hier wie gewohnt in die zwei Bereiche der fast komplett richtig klassifizierten (83% aller Beikräuter haben mindestens eine Genauigkeit von 0,90) und den komplett falsch klassifizierten (5% aller Beikräuter besitzen eine Genauigkeit von unter 0,01). Die einzelnen Pflanzengruppen weisen ein ähnliches Verhalten auf.

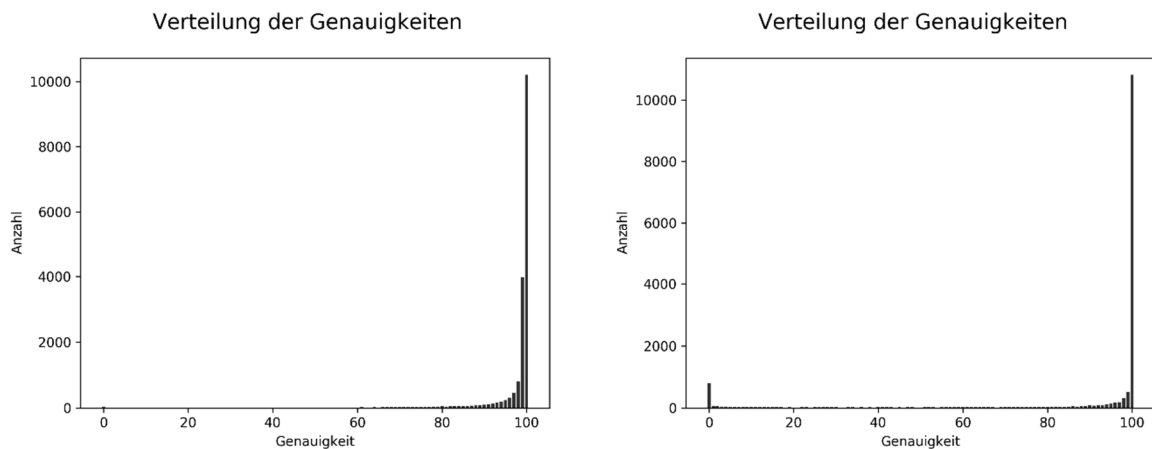
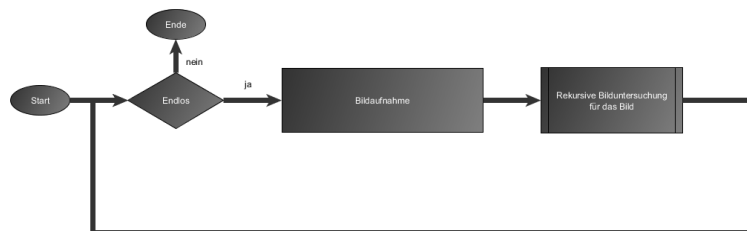


Abbildung 101: Darstellung der Genauigkeitshistogramme des AlexNet-Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“

4.5.5 Area-Netz

Ein weiterer Ansatz, der nicht die pixelweise Klassifikation verfolgt, läge darin, eine Fläche von Pixel auf einmal zu klassifizieren. Der Ansatz, der in dieser Arbeit untersucht wird, kann folgendermaßen beschrieben werden: Ein Bildausschnitt kann mehrere Pflanzenklassen („Karotte“ und „Beikraut“) abbilden. Es wäre nun möglich, das Bild so oft zu teilen, bis in jedem dieser Teilbilder nur noch eine Klasse vorhanden ist. Damit wäre zwar im ersten Moment nicht direkt die Pixelposition bekannt, aber alle Pflanzen, die sich in diesem Teilbild befinden können einer Klasse zugeordnet werden. Die Pixelauflösung der Klassifikation kann anschließend über die Maske des Segmentierers erfolgen. Eine Möglichkeit wäre, die Bilder mittels rekursiven Ansatzes in der Mitte in neue Teilbilder zu zerlegen und zu untersuchen. Die Zerlegung müsste solange fortschreiten, bis lediglich eine Klasse in dem Teilbild vorkommt. Der Klassifikator müsste damit keine Aussage über eine Position einer Klasse im Bild treffen, sondern lediglich angeben, welche Klassen im Teilbereich vorhanden sind. Dieser Klassifikator wurde in [Knoll, et al., 2018] veröffentlicht⁴⁵. Das Ablaufdiagramm in Abbildung 102 stellt die Arbeitsmethode zur Klassifizierung dar.

Hauptfunktion



rekursive Funktion

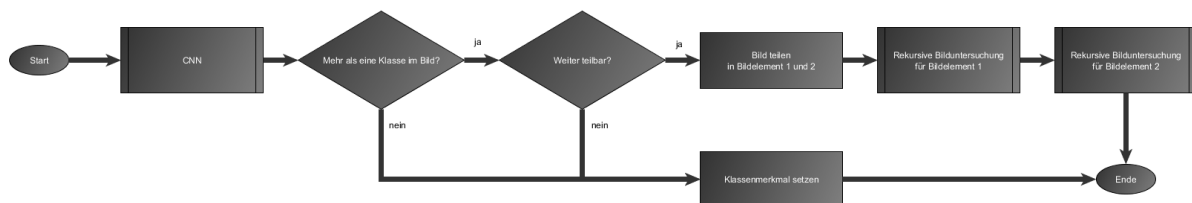


Abbildung 102: Ablaufplan für den Area-Klassifikator

Der Algorithmus beginnt mit der Aufnahme eines Bildes. Dieses Bild wird der rekursiven Funktion übergeben. Die Funktion prüft zum Beispiel mit einem CNN, wie viele Klassen sich in diesem Bild befinden. Wenn nur eine Klasse in diesem Bild vorhanden ist, wird dieses Gesamtbild mit der jeweiligen Klasse verknüpft. Sollte sich mehr als eine Klasse in dem Bild befinden, wird geprüft ob das Bild weiter teilbar ist. Sollte dies der Fall sein, wird das Bild in zwei gleich große Flächen geteilt und für jede dieser Teilbilder wird die rekursive Funktion erneut aufgerufen. Dieser Vorgang wiederholt sich solange, bis entweder nur noch genau eine Klasse in jedem Bildelement vorhanden ist oder das Bild eine minimale Größe beim Teilen unterschreiten würde.

Als Eingangsbild in dem Algorithmus wurde eine Auflösung von 320x320 Pixel gewählt. Dies hat den Grund, dass die Zahl 320 immer wieder halbiert werden kann und ihr Quotient eine ganze Zahl darstellt. Das kleinste Ergebnis wäre damit die Zahl 5. Ein weiterer Vorteil ist es, dass die Zahl 320 exakt sechs Mal in die Bildbreite der Kamera mit 1920 Pixeln passt. Die Höhe wurde ebenfalls auf 320 Pixel gesetzt, auch wenn diese nicht exakt in das Kamerabild von 1080 Pixel passt. Dies hat den Grund, dass ein ähnliches Pixelminimum wie 5 Pixel (heißt: die Zahlen 3, 4, 6, 7...) ebenfalls nicht zu einem besseren

⁴⁵ Die erste Version des Area-Klassifikators wurde mit Sascha Poczihoski nach seiner Abschlussarbeit im Projekt trainiert.

Ergebnis führen würden. Eine quadratische Grundfläche ermöglicht auch weitere Verfahren beim Training, wie später noch erklärt wird.

Damit kann das Kamerabild in 3x6 Segmente zerlegt werden. Dabei fällt ein Rand von 120 Pixeln in der Höhe an. Das Kamerabild wird jedoch in der Regel so über die Karottenreihen zentriert, dass oben und unten ein Bereich entsteht, der nicht von Karotten bewachsen ist. Daher werden die 120 Pixel als Rand von je 60 Pixel aufgeteilt und vorerst ignoriert. Dieser Bereich könnte zwar auch in sechs weitere Bildelemente zerlegt werden, da dieser Bereich jedoch abseits der Karottenreihen ist, kann dies ignoriert werden. Die Aufteilung ist in Abbildung 103 dargestellt. Der rote Bereich zeigt den zu ignorierenden Rand. Der blaue Bereich zeigt die 18 Bildelemente, die einzeln dem iterativen Klassifikationsalgorithmus übergeben werden. Die Bildelemente sind so angeordnet, dass ein Bildelement in der Mitte der Höhe des Bildes liegt. Da in der Mitte überwiegend die Karotten verlaufen, könnten so einige Elemente bereits in der ersten iterativen Stufe nur Beikräuter abbilden, wie dies in der unteren Reihe teilweise der Fall ist, was bedeuten würde, dass bereits in der ersten Klassifikationsstufe große Flächen klassifiziert würden.

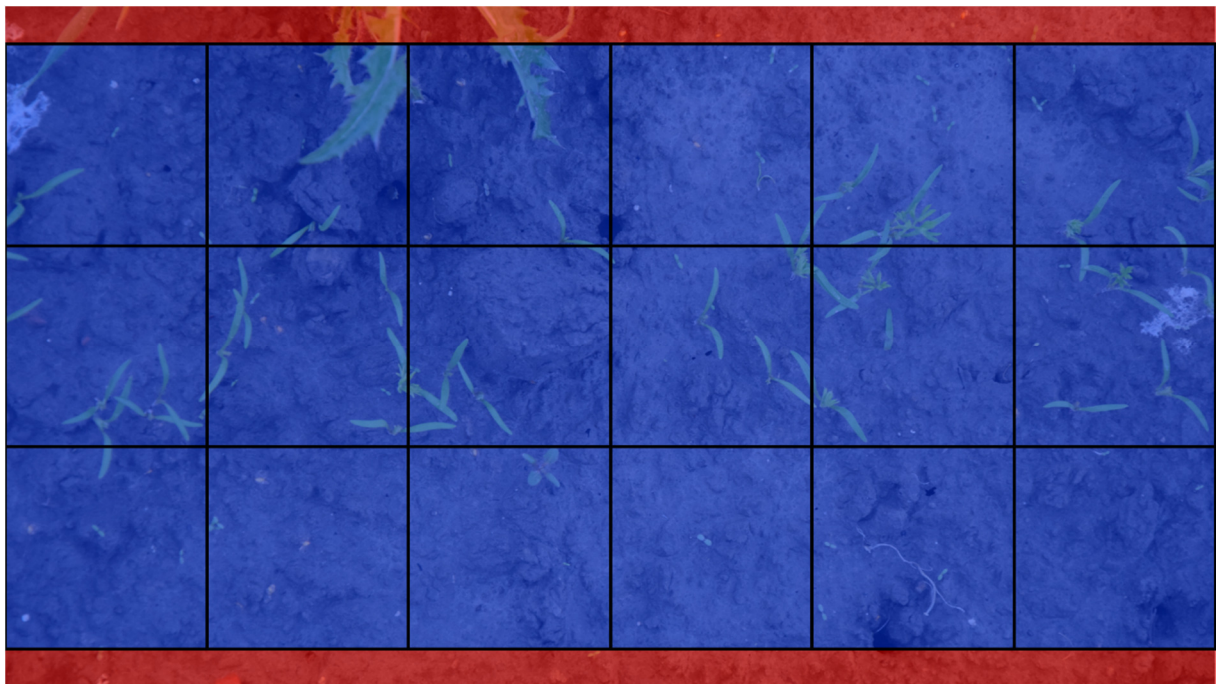


Abbildung 103: Aufteilung des Kamerabildes in die 18 Bildsegmente für den Area-Klassifikator

Jedes der 18 Bildelemente kann dann, wie in Tabelle 34 dargestellt, zerlegt werden. Dabei wird auch hier erst die Höhe geteilt, um den Wachstumsstreifen der Karotten auszunutzen.

Tabelle 34: Darstellung der Elementgrößen des Area Klassifikators

Tiefe		1	2	3	4	5	6	7	8	9	10	11	12	13
Bildgröße	Höhe	320	160	160	80	80	40	40	20	20	10	10	5	5
	Breite	320	320	160	160	80	80	40	40	20	20	10	10	5
ideale benötigte Anzahl der 1075 Testbilder (in Tausend)		19	13	19	26	39	53	83	116	195	294	523	856	1.590

In der letzten Spalte der Tabelle ist die Anzahl der Elemente angegeben, die für die 1075 Bilder benötigt werden würde, sofern der Klassifikator keinen Fehler macht. Je genauer der Klassifikator zwei benachbarte Klassen trennen soll, desto tiefer muss er gehen. Hier könnte für die

Geschwindigkeitsoptimierung notfalls an der Iterationstiefe gespart werden. Dies würde jedoch auch zu Genauigkeitsverlusten an Randbereichen von Klassenübergängen im Bild führen.

Als Klassifikator wurde erneut eine LeNet Struktur verwendet. Der Aufbau besteht aus acht Schichten und hat den folgenden Aufbau, die Hyperparameter wurden dabei wieder experimentell bestimmt:

- Eingang: 3 x (320x320)
- Convolution: 16 x (15x15)
- Pooling: 16 x (2x2)
- Convolution: 24 x (13x13)
- Pooling: 24 x (2x2)
- Convolution: 48 x (9x9)
- Pooling: 28 x (2x2)
- Dense: 1 x (64)
- Ausgang: 1 x (4)

Die Aktivierungsfunktion ist auch hier wieder jeweils die ReLu-Funktion⁴⁶. Das Ausgangsschaltznetz besteht aus vier Neuronen. Die Neuronen bilden dabei die vier Klassen: „Keine Pflanze“, „nur Karotte“, „nur Beikraut“ und „beide Arten (Karotte und Beikraut)“ ab. Die Eingangsschicht ist immer ein RGB Bild mit der Größe von 320x320 Pixeln. Auch nach dem Verkleinern des Eingangsbildes wird dem Netzwerk erneut diese Größe übergeben. Das verkleinerte Bild wird dabei in die Mitte der 320x320 Pixel gesetzt und der restliche Bereich wird auf Schwarz gesetzt. Zum Trainieren wurde erneut eine Batchsize von 64 Bildern verwendet. Jedoch werden wesentlich mehr Bildausschnitte benötigt, als bei den pixelweisen Klassifikatoren. Die Bildelemente werden dabei auch wieder aus den oben beschriebenen Lernbildern extrahiert. Trainiert wurden eine Million Epochen mit jeweils 64 Bildern (Batchgröße), dabei wurden immer neue Bilder verwendet, so dass 64 Millionen Bilder zum Training verwendet wurden. Dabei ist es möglich, dass zwei verwendete Bildelemente lediglich eine Translation⁴⁷ von wenigen Pixeln aufweisen. Zur Bereitstellung dieser Datengrundlagen wurden der Hintergrund der Lernbilder auf Schwarz gesetzt und die einzelnen Pflanzen ausgeschnitten. Die ausgeschnittenen Pflanzen wurden anschließend automatisch zu den neuen benötigten Bildelementen zusammengesetzt. Aufgrund der Bildmenge und der Rechenzeit beim Training kann hier keine Kreuzvalidierung verwendet werden. Der Batchaufbau verwendete jeweils fünf Bilder jeder Bildelementgröße aus Tabelle 34, lediglich das kleinste Bildelement von 5x5 Pixeln war nur vier Mal enthalten.

Die pixelweise Auswertung der 1075 Testbilder steht in Tabelle 35. Auch der Area-Klassifikator erreicht eine Genauigkeit von 0,95 über die globale Genauigkeitsanalyse. Die Precision- und Recall-Werte liegen ebenfalls im Bereich von 90%.

Tabelle 35: Darstellung der Auswertungsergebnisse des ersten Area-Klassifikators mit 4 Ausgängen über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung

Klasse	Genauigkeit	Precision	Recall	F1-Score
Karotte	0,95	0,93	0,99	0,95
Beikraut		0,98	0,92	0,95

Werden auch hier wieder die Genauigkeiten innerhalb einer einzelnen Pflanze betrachtet (lokale Genauigkeitsanalyse, siehe Tabelle 36), also untersucht, wieviel Prozent der Pixel einer Pflanze richtig klassifiziert wurden und anschließend über die Pflanzengenauigkeit ein Mittelwert gebildet, zeigt sich

⁴⁶ Die **Rectifier Linear Unit** ist in Kapitel 2.2.5.1 beschrieben und in Abbildung 17 dargestellt.

⁴⁷ Auch eine Translation von ein Pixel wäre möglich.

auch hier wieder, dass die Genauigkeit bei der Karotte besser ist als beim Beikraut. Außerdem zeigt sich auch hier wieder, dass die Genauigkeitsverteilung keine Normalverteilung darstellt.

Tabelle 36: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des ersten Area-Klassifikators mit 4 Ausgängen

Kategorie	Genauigkeit (SD)	Anzahl	Genauigkeit $\geq 0,90$	Genauigkeit $\geq 0,95$
Gesamt	0,92 _(0,25)	32.983	88%	86%
Karotte gesamt	0,98 _(0,13)	17.867	95%	94%
Beikraut gesamt	0,84 _(0,33)	15.116	79%	77%
Karotte ohne 3. Blatt	0,98 _(0,12)	13.247	96%	94%
Karotte mit mind. 3 Blatt	0,97 _(0,13)	4.620	94%	93%
Beikraut klein	0,84 _(0,36)	13.786	79%	77%
Beikraut mittel	0,78 _(0,36)	581	67%	64%
Beikraut Knöterich	0,97 _(0,12)	422	94%	93%
Beikraut groß	0,74 _(0,37)	327	60%	48%

Die Darstellung der Genauigkeitsverteilung, wie sie in Abbildung 104 abgebildet ist, zeigt, dass der Area-Klassifikator wie die anderen untersuchten Methoden zur Klassifikation auch ein ähnliches extremes Verhalten aufweist. Es zeigt sich, dass 83% aller Pflanzen mit einer Pixelgenauigkeit von mindestens 0,99 klassifiziert werden. Das andere Extrem zeigt, dass 5% aller Pflanzen eine Pixelgenauigkeit von unter 0,01 haben. Die Größenordnung ist damit ähnlich wie bei den anderen Klassifikatoren.

Verteilung der Genauigkeiten

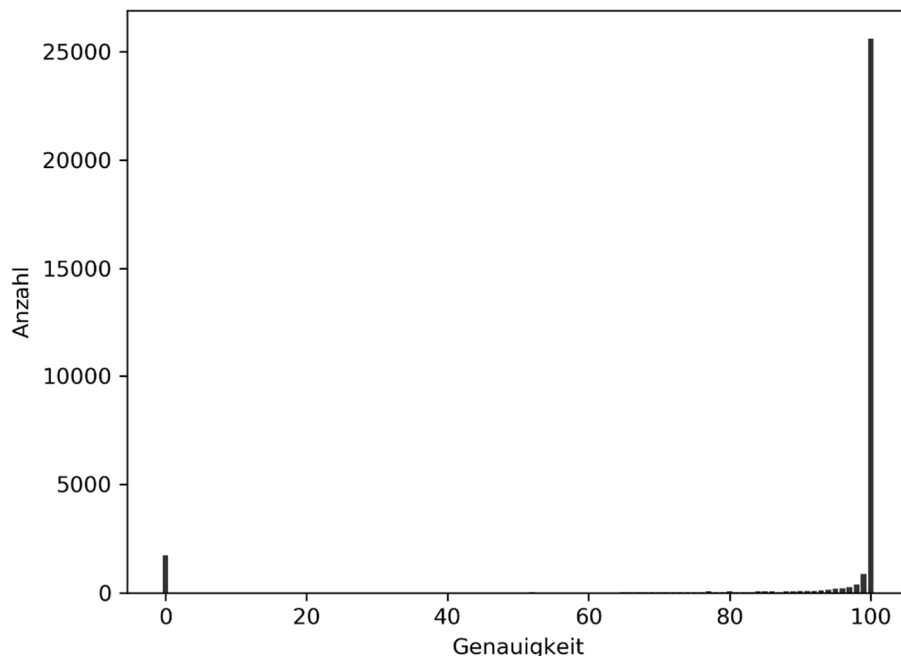


Abbildung 104: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des ersten Area-Klassifikators mit 4 Ausgängen.

Auch hier zeigt die Einzelbetrachtung der Klasse „Karotte“ und der Klasse „Beikraut“ wie in Abbildung 105, dass die Schwierigkeit des Klassifikators vor allem in der Klasse „Beikraut“ liegt. Die Karotte wird fast komplett richtig klassifiziert, wobei das Beikraut sich in die zwei Extrema aufteilt. Bei

genauerer Betrachtung fällt jedoch auf, dass auch die Karotte eine geringe Anzahl von komplett falsch klassifizierten Pflanzen aufweist.

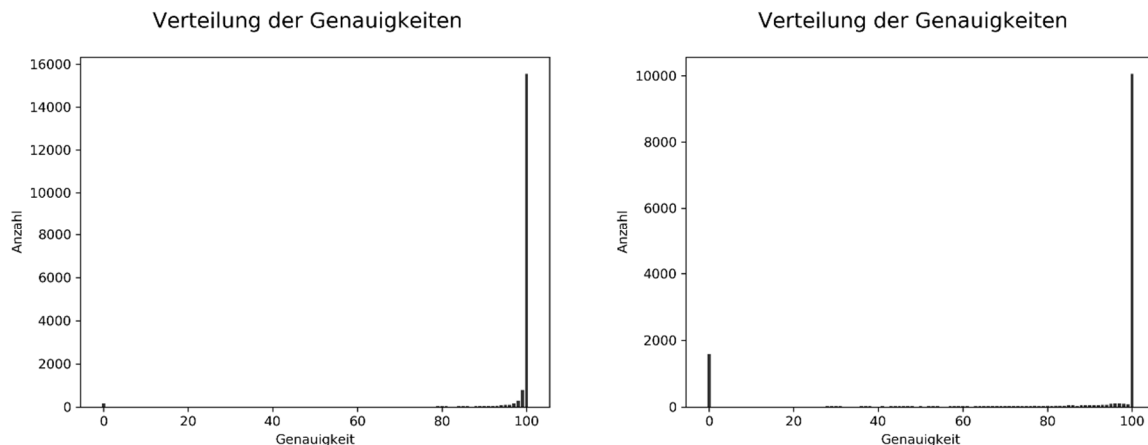


Abbildung 105: Darstellung der Genauigkeitshistogramme des ersten Area-Klassifikators mit 4 Ausgängen mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“

Das Problem kann an einem klassifizierten Bild anschaulich erklärt werden. In Abbildung 106 ist ein Klassifikationsergebnis als Überlagerung auf dem Eingangsbild dargestellt. In dieser Abbildung sind die grünen Flächen vom Klassifikator als Karotten bezeichnet worden und in den roten Flächen befindet sich Beikraut. Zu beachten ist außerdem, dass der Klassifikator nicht das gesamte Bild bearbeitet, sondern nur einen Streifen in der Mitte des Bildes, wie weiter oben erklärt wird. Ein wesentliches Problem zeigt der blaue Kreis im Bild: Der Klassifikator hat Probleme, seine volle Iterationstiefe zu verwenden. Dies bedeutet, dass er im blauen Bereich das Bild weiter hätte zerteilen müssen, dies jedoch nicht macht. Dieser Effekt führt auch dazu, dass vor allem kleine Pflanzen komplett falsch klassifiziert werden oder auch Karotten, wenn sich diese im selben Bildausschnitt mit einem größeren Beikraut befinden. Die Vermutung ist, dass die Lernmenge nicht alle Variationen abdeckt. Daher könnte eine größere Lernmenge mit weiteren Variationen dieses Problem lösen.

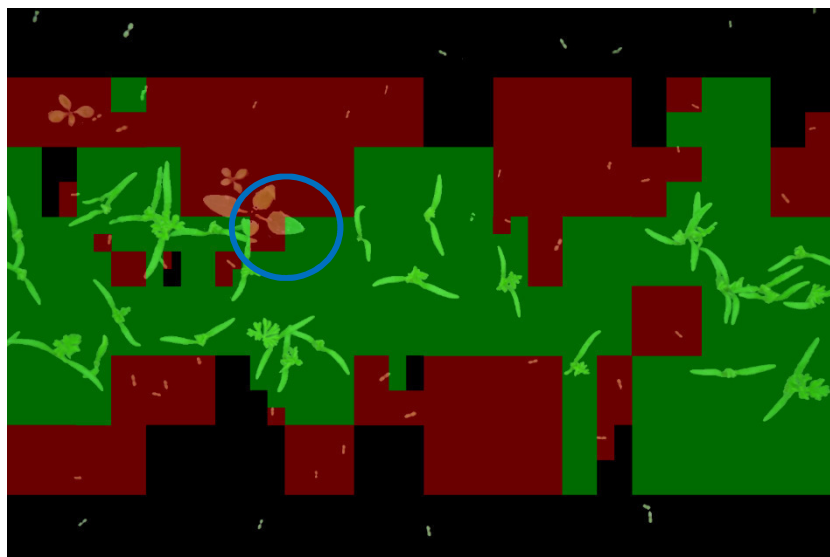


Abbildung 106: Darstellung eines Klassifikationsergebnisses des ersten Area-Klassifikators mit 4 Ausgängen mittels Überlagerungsdarstellung; grüne Flächen sind Karotten, rote Flächen beinhalten Beikraut

Im Folgenden soll der Area-Klassifikator mit einer kleineren Netzwerktopologie untersucht werden, um die Geschwindigkeit bei der Klassifikation zu erhöhen und die Trainingszeit zu verringern. Für diese Untersuchung wurde das Netzwerk auf die folgende Größe reduziert:

- Eingang: 3 x (320x320)
- Convolution: 16 x (15x15)
- Pooling: 16 x (2x2)
- Convolution: 16 x (15x15)
- Pooling: 16 x (2x2)
- Dense: 1 x (64)
- Ausgang: 1 x (4)

Das Netzwerk besteht nun aus einem Convolution-Pooling-Paar weniger und die zweite Convolution-Schicht hat eine größere Featuregröße (von 13x13 auf 15x15 Pixel) bekommen. Außerdem wurde die Dimension von 24 auf 16 herabgesetzt. Die Ergebnisse der globalen Genauigkeitsanalyse nach einem erneuten Training für alle Pixel der 1075 Bilder (wie oben beschrieben) sehen wie folgt aus:

Tabelle 37: Darstellung der Auswertungsergebnisse des modifizierten Area-Klassifikators mit 4 Ausgängen über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung

Klasse	Genauigkeit	Precision	Recall	F1-Score
Karotte	0,93	0,90	0,96	0,93
Beikraut		0,96	0,90	0,93

Die Tabelle 37 zeigt, dass die Ergebnisse in Bezug auf den Area-Klassifikator mit einer zusätzlichen Convolution-Pooling-Schicht leicht (2%) gesunken sind. Jedoch sollte nicht davon ausgegangen werden, dass dies an der Modifikation direkt liegt. Ein Grund könnten auch die unterschiedlichen zufällig gewählten Lernbilder sein. Die Genauigkeitsverteilung in der Histogrammdarstellung (Abbildung 107) der lokalen Genauigkeitsanalyse zeigt wieder ein ähnliches Verhalten.

Verteilung der Genauigkeiten

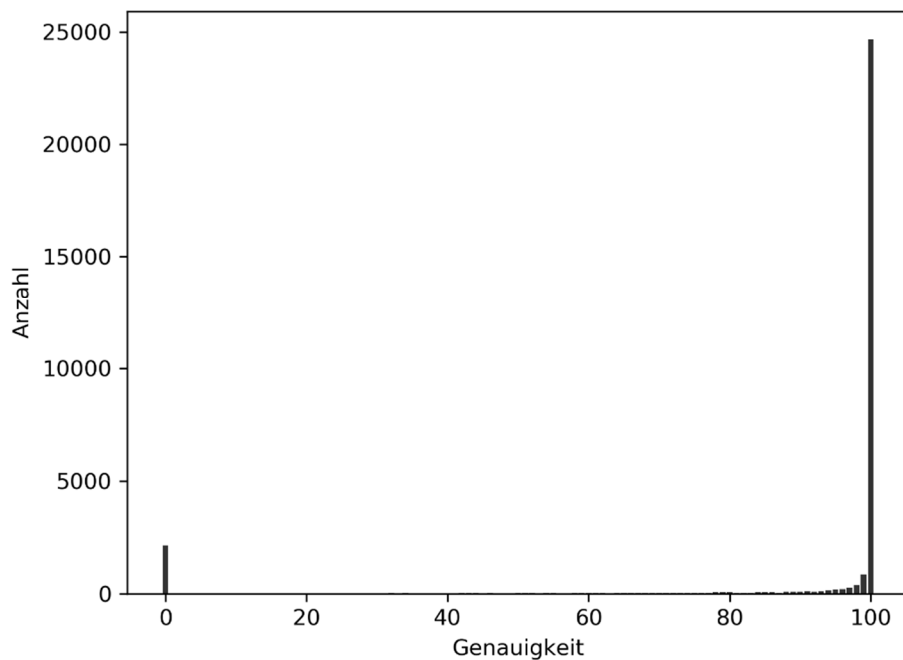


Abbildung 107: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des modifizierten Area-Klassifikators mit 4 Ausgängen

Die Abbildung zeigt, dass der Charakter der Verteilung ähnlich ist, wie beim ursprünglichen Netzwerk. Die Histogrammdarstellung der beiden Einzelklassen ist in Abbildung 108 dargestellt.

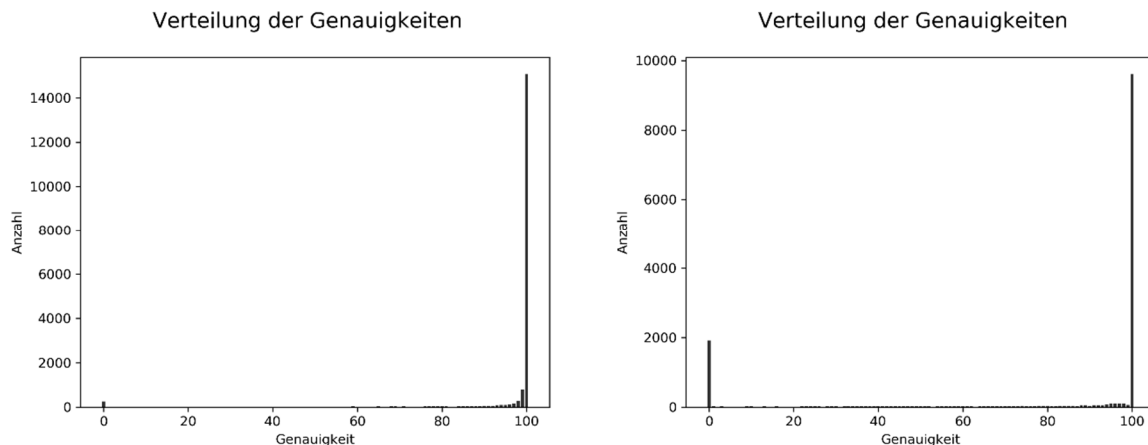


Abbildung 108: Darstellung der Genauigkeitshistogramme des modifizierten Area-Klassifikators mit 4 Ausgängen mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“

Auch hier ist der Charakter der Verteilung ähnlich dem des ursprünglichen Netzwerkes. Das Beikraut zeigt hier die beiden Extrema und die Karotte weist kaum Fehler auf. Der modifizierte Klassifikator soll daher für die weitere Untersuchung verwendet werden.

Um die Klassifikationsergebnisse zu verbessern, soll das Netzwerk mit Hilfe von Multi-Label Klassifikation neu aufgebaut und trainiert werden. Dies bedeutet, dass das Netzwerk in Zukunft nur noch zwei Ausgänge besitzen soll, also aufgebaut ist wie folgt:

- Eingang: 3 x (320x320)
- Convolution: 16 x (15x15)
- Pooling: 16 x (2x2)
- Convolution: 16 x (15x15)
- Pooling: 16 x (2x2)
- Dense: 1 x (64)
- Ausgang: 1 x (2)

Die beiden Ausgangsneuronen stehen dabei jeweils für eine Klasse. Einmal für die Karotte und einmal für das Beikraut. Die Ausgänge geben damit die Wahrscheinlichkeit an, ob sich die jeweilige Klasse im Bild befindet. Die Grenze wurde dabei auf den Wert 0,5 gesetzt, so dass sich folgende Interpretation des Ausgangs ergibt.

Tabelle 38: Ausgangsinterpretation des Area-Klassifikators mit zwei Ausgangsneuronen

Neuron für Karotte	Neuron für Beikraut	Interpretation	Weiteres Vorgehen
< 0,5	< 0,5	Es handelt sich um Hintergrund.	als Hintergrund markieren
≥ 0,5	< 0,5	Es sind nur Karottenpflanzen im Bild.	als Karotte markieren
< 0,5	≥ 0,5	Es sind nur Beikrautpflanzen im Bild.	als Beikraut markieren
≥ 0,5	≥ 0,5	Es sind Karotten- als auch Beikrautpflanzen im Bild.	Bild teilen und beide Teile neu klassifizieren

Die Schwellwerte können natürlich auch verwendet werden, um eine gewisse Empfindlichkeit des Klassifikators einzustellen, sollen jedoch vorerst auf diesen Werten bleiben. Die Idee dahinter ist, dass sich das Netzwerk nicht mehr selbst entscheiden muss, ob zwei Klassen im Bild sind oder nicht. Jedes Ausgangsneuron überprüft für sich die gefundenen Eigenschaften auf seine Klasse, unabhängig davon, was das Neuron der anderen Klasse herausfindet. Damit könnte im Zweifel die Entscheidung auf ein weiteres Teilen des Bildes leichter fallen. Nach einem erneuten Training des Netzwerkes mit dem oben beschriebenen Verfahren ergeben sich folgende Ergebnisse über die 1075 Bilder für die globale Genauigkeitsanalyse.

Tabelle 39: Darstellung der Auswertungsergebnisse des Area-Klassifikators mit 2 Ausgängen über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung

Klasse	Genauigkeit	Precision	Recall	F1-Score
Karotte	0,95	0,95	0,95	0,95
Beikraut		0,95	0,95	0,95

Die Werte in Tabelle 39 zeigen, dass die Klassifikationsergebnisse für jeden Pixel aller Bilder bei 95% Genauigkeit liegen. Auch die Precision-, Recall- und F1-Score-Werte liegen bei 95%. Die Werte liegen damit in der Umgebung der beiden anderen Variationen des Area-Klassifikators. Die Durchschnittswerte der richtig klassifizierten Pixel einer Pflanzenart (lokale Genauigkeitsanalyse), stehen in Tabelle 40.

Tabelle 40: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des Area-Klassifikators mit 2 Ausgängen

Kategorie	Genauigkeit (SD)	Anzahl	Genauigkeit ≥0,90	Genauigkeit ≥0,95
Gesamt	0,90 (0,27)	32.983	86%	84%
Karotte gesamt	0,96 (0,16)	17.867	91%	90%
Beikraut gesamt	0,83 (0,35)	15.116	78%	76%
Karotte ohne 3. Blatt	0,96 (0,15)	13.247	93%	91%
Karotte mit mind. 3 Blatt	0,93 (0,19)	4.620	88%	86%
Beikraut klein	0,83 (0,35)	13.786	78%	77%
Beikraut mittel	0,82 (0,34)	581	71%	67%
Beikraut Knöterich	0,99 (0,04)	422	97%	95%
Beikraut groß	0,83 (0,32)	327	73%	65%

Die Tabelle zeigt, dass 90% aller Pixel einer Pflanze im Durchschnitt richtig klassifiziert wurden, jedoch auch, dass die Standardabweichung von 0,27 auf eine wieder nicht-normalverteilte Funktion hinweist, weshalb diese Tabelle nur der Vollständigkeit halber hier dargestellt wurde. Zur besseren Analyse und zum Vergleich soll erneut das Histogramm die Verteilung der Pixelgenauigkeiten zeigen.

Verteilung der Genauigkeiten

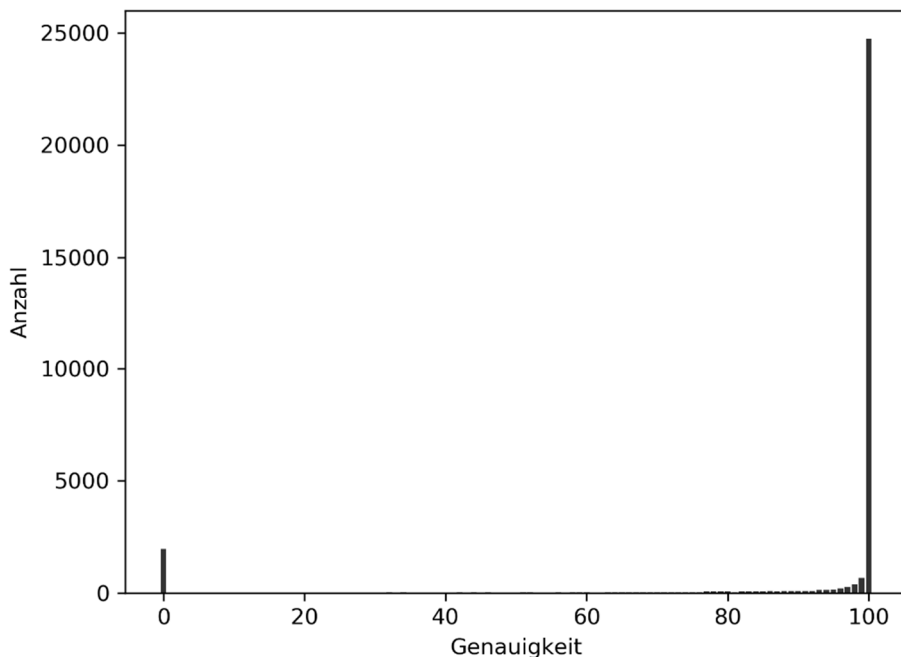


Abbildung 109: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des Area-Klassifikators mit 2 Ausgängen.

In Abbildung 109 wird gezeigt, von wie vielen Pflanzen, wie viel Prozent ihrer Pixel richtig klassifiziert wurden. Auch hier stechen die beiden Extrema heraus. Von 80% aller Pflanzen werden mindestens 99% aller Pixel richtig klassifiziert. Die Pflanzen, bei denen mindestens 90% aller Pixel richtig klassifiziert wurden, stellen 86% der gesamten Testmenge. Die Pflanzen, die weniger als 1% richtig klassifizierter Pixel besitzen belaufen sich auf 6%. Die Ergebnisse sind ähnlich wie beim ursprünglichen Netzwerk. Auch die Aufteilung der Histogramme in die beiden Klassen „Karotte“ und „Beikraut“, wie in Abbildung 110, zeigt einen ähnlichen Charakter.

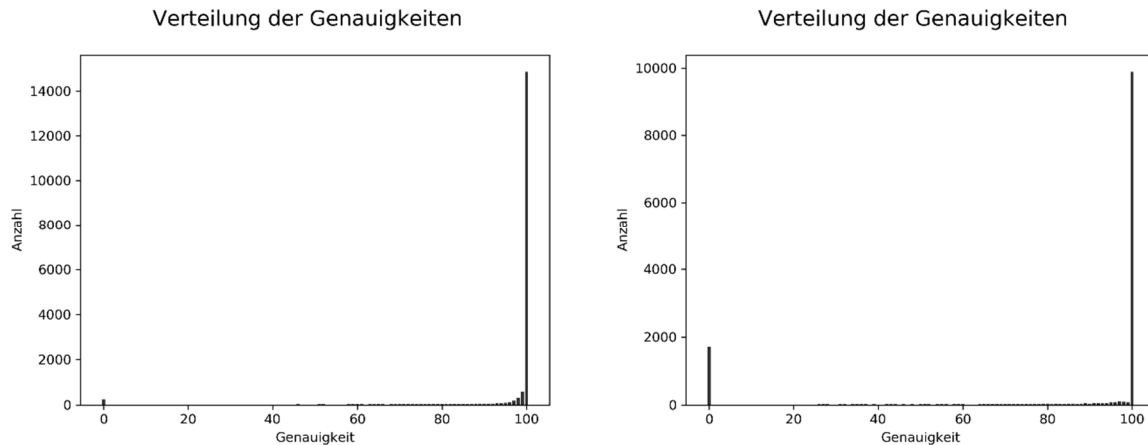


Abbildung 110: Darstellung der Genauigkeitshistogramme des Area-Klassifikators mit 2 Ausgängen mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“

Neben einem verbesserten Lernen sollte auch die Netzwerkstruktur weiter untersucht werden. Es kann sicher noch die Featuregröße (derzeit bei 15x15 Pixel) reduziert werden.

Eine weitere Modifikation läge darin, das Netz separat für die Eingangsbildgröße zu designen und zu trainieren. Damit ist gemeint, dass ein Netzwerk für die kleinste Bildstufe (hier 5x5 Pixel) entworfen und trainiert wird. Da es sich bei Eingangsbildern von 5x5 Pixel um Bilder handelt, die nicht die große Variation aufweisen, wie Bilder von zum Beispiel 320x320 Pixel, würde hier die Convolution-Schicht eventuell mit einem Feature ausreichen. Die Erweiterung des Netzwerkes auf die nächstgrößeren Eingangsbilder könnte dann auf diesen Ergebnissen aufbauen, so dass das eben trainierte Feature auch für die nächsten Eingangsbildgrößen zusammen mit weiteren Features verwendet wird. Dieser Vorgang wird dann so oft wiederholt, bis die maximale Eingangsbildgröße erreicht wurde. Dabei wird für jede Einganggröße eine eigene Dense-Schicht trainiert.

Auch könnte ein sinnvolleres Aufteilen der Bildfragmente ein Vorteil sein. Derzeit wird die Bildaufteilung dadurch erzeugt, dass das Eingangsbild genau in der Hälfte in zwei gleich große Unterbilder geteilt wird. Es könnte von Vorteil sein, die Teilung zwischen zwei Pflanzen zu vollziehen, wenn sich zum Beispiel zwei separate Pflanzen von verschiedenen Klassen im Bild befinden. Da die Wahrscheinlichkeit vermutlich hier am höchsten wäre, dass eine Pflanze zu einer Klasse und die andere zur anderen Klasse gehört. Die Unterscheidung der Trennlinie sollte jedoch schneller sein, als das herkömmliche Vorgehen, da sonst der Geschwindigkeitsvorteil verloren gehen würde.

4.5.6 Auswertung der Klassifikatoren

Im folgenden Kapitel sollen die präsentierten Ergebnisse der untersuchten Klassifikatoren gegenübergestellt und diskutiert werden. Anhand der Ergebnisse soll ein Klassifikator zur weiteren Untersuchung und zur Optimierung für die Pflanzenklassifikation ausgewählt werden. Die vorgestellten Klassifikatoren sind alle Deep-Learning Ansätze und erreichten, bis auf das einfache MLP, sehr gute Klassifikationsergebnisse. Daher sollen hier nur die drei pixelweisen Klassifikatoren (CNN, GoogLeNet und AlexNet) sowie der Flächenweise Klassifikator (Area) weiter betrachtet werden. Die Ergebnisse der globalen Genauigkeitsanalyse sind zusammengefasst in Tabelle 41 dargestellt. Die einzelnen Werte der Genauigkeit, der Precision, des Recalls und damit auch des F1-Score ähneln sich so stark, dass hier nicht von statistischen Unterschieden gesprochen werden kann. Alle pixelweisen Klassifikatoren wurden für die Auswertung mit denselben Trainingsbildern trainiert. Außerdem wurden für diese Netze 100.000 Epochen mit einer Batchgröße von 64 zum Training verwendet. Das maximale konstante Niveau der Genauigkeit erreichten diese Netze vor den 100.000 Epochen. Das Area-Netz verwendete ebenfalls eine Batchgröße von 64 Bildern, jedoch benötigte es eine Million Epochen und eine andere Bildmenge für das Training.

Tabelle 41: Gegenüberstellung der Klassifikationsergebnisse aus den globalen Genauigkeitsuntersuchungen

Klassifikator	Klasse	Genauigkeit	Precision	Recall	F1-Score
CNN	Karotte	0,96	0,95	0,97	0,96
	Beikraut		0,97	0,95	0,96
GoogLeNet	Karotte	0,95	0,94	0,97	0,95
	Beikraut		0,97	0,94	0,95
AlexNet	Karotte	0,95	0,94	0,96	0,95
	Beikraut		0,96	0,94	0,95
Area	Karotte	0,95	0,95	0,95	0,95
	Beikraut		0,95	0,95	0,95

Auch die Analyse über die lokale Pflanzengenauigkeit der vier Klassifikatoren zeigt ein ähnliches Verhalten. Die präsentierten Genauigkeitsverteilungen mittels Histogrammdarstellung zeigen, dass sich die Genauigkeiten der einzelnen Pflanzen in zwei wesentliche Extrema aufteilen. Einmal in das Extrem der fast oder komplett falsch klassifizierten Pflanzen und einmal in das Extrem der fast oder komplett richtig klassifizierten Pflanzen. Dabei ist die Ausprägung bei den Beikräutern besonders stark und bei den Karotten fast nicht existent. Daher wäre es unzureichend, den Klassifikator nur anhand der untersuchten Genauigkeiten auszuwählen.

Ein wesentlicher Unterschied besteht in der benötigten Verarbeitungszeit der Klassifikatoren. Die Zeit, die die Netze für die Klassifikation der 1075 Validierungsbilder benötigt haben, sind in der Tabelle 42 aufgeführt. Die Validierungsbilder besitzen alle eine unterschiedliche Anzahl von Pflanzenpixel, daher soll hier kein Mittelwert pro Bild angegeben werden, sondern lediglich die benötigte Gesamtzeit. Die Messung soll auch lediglich einen Eindruck der benötigten Größenordnung wiedergeben. In der Zeiterfassung ist weder das Laden noch das Speichern der Bilder enthalten. Auch das Segmentieren wurde hier nicht mitefasset. Es beinhaltet daher nur die eigentliche Klassifikation. Weil das GoogLeNet teilweise für ein Bild mehr als eine Stunde benötigt, wurde entschieden, nur jeden zehnten Pflanzenpixel auszuwerten. Daher ist die Zeitmessung des GoogLeNet auch nur auf jeden zehnten Pflanzenpixel ausgelegt.

Tabelle 42: Benötigte Zeit für die Klassifikation der 1075 Validierungsbilder

Klassifikator	Bilder	Gesamtzeit	Klassifikationen	Theoretische Zeit pro Klassifikation
CNN	1075	24.775s (~7h)	85.383.882	0,290ms
GoogLeNet	1075	322.058s* (~4d)	11.341.040	28ms
AlexNet	1075	56.891s (~16h)	85.383.882	0,666ms
Area	1075	120s	180.600	0,665ms

*für jeden 10. Pflanzenpixel

Die Tabelle zeigt, dass das Area-Netz die besten zeitlichen Eigenschaften bietet. Dies liegt vor allem daran, dass nicht jeder Pixel einzeln betrachtet wird, sondern ganze Regionen im Bild auf einmal klassifiziert werden können. Das GoogLeNet hingegen benötigt extrem viel Zeit für die Klassifikation, was auf die Größe des Netzwerkes zurückzuführen ist. Genauso wie das AlexNet ist auch das GoogLeNet nicht für diese Art der Herangehensweise vorgesehen. Die ursprüngliche Aufgabe der beiden Netze liegt darin, eher eine allgemeine Aussage darüber zu treffen, was auf einem Bild zu sehen ist und 1000 Klassen zu unterscheiden. Das AlexNet benötigt zwar weniger Zeit als das GoogLeNet, zeigt jedoch auch hier starke zeitliche Kosten. Das CNN, das auf der LeNet Struktur basiert, benötigt nicht ganz so viel Zeit, ist jedoch auch langsamer als das Area-Netz. Zu Bedenken ist hier, dass die Klassifikatoren nicht wirklich auf die Zeit optimiert worden sind, jedoch alle denselben Basiscode durchlaufen. Es wurde selbstverständlich auch dieselbe Hardware verwendet, wozu eine NVIDIA GeForce GTX 1080 Ti Grafikkarte gehört.

Im Folgenden soll das CNN, das auf Basis der LeNet Struktur basiert, weiter untersucht und optimiert werden. Das Area-Netz bietet zwar vergleichbare Genauigkeitswerte und auch eine bessere Zeit-Performance für die Klassifikation, benötigt jedoch ein längeres und komplexeres Training. Der Pixelweise-Klassifikator kann innerhalb eines Tages trainiert werden. Der Area-Klassifikator benötigt hingegen eine Woche, wegen der größeren benötigten Anzahl von Lernbildern. Ein Problem ist dabei auch, dass der Erfolg des Trainings erst am Ende ersichtlich ist und daher eine Woche Rechenzeit bei der falschen Wahl von Startparametern vergeudet würde. Das AlexNet und GoogLeNet bieten ebenfalls identische Klassifikationserfolge, verwenden hierfür jedoch auch eine komplexere Struktur. Das GoogLeNet besitzt unter anderem auch vortrainierte Schichten für die 1000 Class-Challenge. Für eine Optimierung wäre es notwendig, diese Inception-Schichten für die Pflanzenklassifikation anzupassen, dies ist jedoch mit der hier zur Verfügung stehenden Hardware rechentechnisch (zeitlich) derzeit nicht möglich. Auch bieten die Inception-Schichten nach den Untersuchungen der Genauigkeiten scheinbar keinen Vorteil für die Pflanzenerkennung.

Im nächsten Schritt soll die Genauigkeit des CNN näher untersucht werden. Alle Klassifikatoren zeigten ein ähnliches Verhalten, in dem es zwei Extrema in der Verteilung der Genauigkeiten innerhalb einer Pflanze gibt. Eine bereits genannte Vermutung ist, dass die Fehlklassifikationen zu Stande kommen, wenn das Eingangsbild des Netzwerkes einer kleinen zu klassifizierenden Pflanze ebenfalls Teile einer größeren Pflanze der anderen Klasse beinhaltet. Eine Beispielkonstellation ist in Abbildung 111 dargestellt. Hier ist die Karotte die große Pflanze und das Beikraut die kleine zu klassifizierende Pflanze. Dies wäre nach den Daten auch die häufigste Konstellation dieser Fehler. Dadurch, dass die kleinen Pflanzen den größten Anteil der Beikräuter in den Validierungsdaten ausmachen, ist diese Konstellation statistisch gesehen am wahrscheinlichsten.



Abbildung 111: Beispielkonstellation von Pflanzen, die zum Problem der Falschklassifikation führen könnten

Zum Nachweis, dass die Fehler auf diese Ursache zurückzuführen sind, soll ein Experiment helfen. Das Netzwerk wird, wie oben bereits vorgestellt, neu trainiert. Diesmal wird jedoch der Hintergrund bei den Eingangsbildern mit dem Segmentierer auf Schwarz gesetzt. Anschließend wird das Netzwerk für die Auswertung der 32.983 Pflanzen verwendet. Außerdem werden die 32.983 Pflanzen nochmal in einer Vergleichsuntersuchung klassifiziert, jetzt jedoch so, dass im Bild alle Pflanzen auf Schwarz gesetzt werden, außer der Pflanze, dessen Pixel gerade klassifiziert werden. So ist im Eingangsbild des Netzwerkes immer nur die Pflanze abgebildet, welche im mittleren Pixel sitzt. Nach der Auswertung kann erneut die lokale Genauigkeitsanalyse ausgewertet werden. Dies geschieht über ein Histogramm, das über die Verteilung der Klassifikationsergebnisse innerhalb einer Pflanze erzeugt wird.

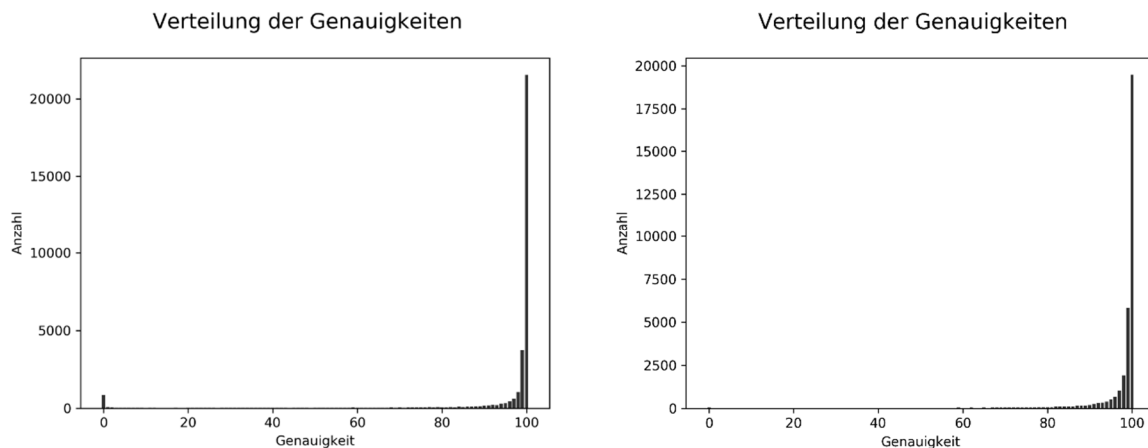


Abbildung 112: Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen; Links: Klassifikation ohne Ausblendung von Nachbarpflanzen; Rechts: Klassifikation mit Ausblenden von Nachbarpflanzen

In der Histogrammdarstellung der Abbildung 112 ist zu sehen, dass der extreme Wert der komplett falsch klassifizierten Pflanzen in der Messung, in der die Nachbarpflanzen ausgeblendet wurden, verschwunden ist, wohingegen der extreme Wert bei der Vergleichsmessung, also ohne Ausblendung, noch vorhanden ist. Dies zeigt, dass das Problem wirklich darauf beruht, dass Nachbarpflanzen, sofern diese einer anderen Klasse angehören, das Klassifikationsergebnis beeinflussen. Die Segmentierung der

Pflanzen, also das Ausblenden des Hintergrundes, bringt keinen nennenswerten Genauigkeitsvorteil gegenüber der ursprünglichen Untersuchung. Aus rechenzeittechnischer Sicht ist es nicht möglich, die Pflanzen einzeln zu separieren, bevor sie klassifiziert werden. Hier ist es nur möglich, weil jeder Pixel einer Pflanze im Label eine eindeutige Pflanzenidentifikationsnummer besitzt. Dies ist unter realen Bedingungen jedoch nicht der Fall. Daher muss ein anderer Weg gefunden werden.

Die Lösung für dieses Problem könnte sein, das Netzwerk beim Training intensiver mit Lerndaten zu konfrontieren, die genau dieses Problem aufzeigen. Dadurch würde das Netzwerk die Konstellation zweier Pflanzen besser erkennen und auswerten können. Eine andere und weitere Möglichkeit läge darin, das Eingangsbild, das derzeit bei 101x101 Pixeln liegt, zu verringern. Dadurch wäre der Bildausschnitt des Kamerabildes kleiner und die Wahrscheinlichkeit geringer, dass eine andere Pflanzenklasse mit im Bild abgebildet wird. Im Nachfolgenden muss das Netzwerk noch auf die Klassifikationsgeschwindigkeit optimiert werden. Dabei soll auch die Netzwerkstruktur weiter optimiert werden, wobei auch der Aspekt der Verringerung der Eingangsbildgröße miteinfließen soll.

4.5.7 Fazit zu den Klassifikatoren

In diesem Kapitel wurden verschiedene Ansätze von Klassifikatoren für die Pflanzenklassifikation untersucht. Dabei wurden vor allem lernfähige Strukturen verwendet, da die Pflanzenerkennung durch die zeitlichen Veränderungen einer Pflanze eine stetig wachsende Datenmenge hervorbringt. Von den fünf untersuchten Klassifikatoren wurden verschiedene Untersuchungen der Genauigkeit, der Precision, des Recalls und damit auch des F1-Score durchgeführt. Außerdem wurde eine Analyse der Signifikanz der Untersuchungsergebnisse mit Hilfe einer Genauigkeitsverteilung aufgezeigt. Alle Klassifikatoren sind Ansätze von Deep-Learning-Strukturen. Für die Klassifikation selbst wurden zwei Arten untersucht, die pixelweise Klassifikation, in der für jeden Pixel im Bild separat eine Aussage über die Klasse getroffen wird und eine flächenbasierte Klassifikation. Hierfür wurde eigens ein Area-Klassifikator entworfen, der sich eine gesamte Region ansieht und anschließend eine Aussage über die enthaltenen Klassen trifft. Es wurde entschieden, lediglich zwei Klassen zuzulassen, so dass alle Beikräuter in einer Klasse zusammengefasst wurden. Dies hat den Effekt, dass eine neue unbekannte Pflanze eine größere Ähnlichkeit mit der Klasse „Beikraut“ hat, da hier die Variation der verschiedenen Eigenschaften einer Pflanze größer wäre, als in der Klasse „Karotte“ oder einer anderen speziellen Beikrautklasse. Am Ende des Kapitels wurden die untersuchten Klassifikatoren gegenübergestellt und es wurde ein Klassifikator ausgewählt. Dieser Klassifikator wurde dann hinsichtlich seiner Fehler genauer analysiert und soll in den folgenden Kapiteln auf seine Klassifikationsgeschwindigkeit weiter optimiert werden.

Viele Faktoren beim Training sind derzeit noch nicht tiefgehend untersucht worden, so wurde die Batchgröße derzeit konstant festgelegt und könnte in einer späteren Arbeit genauer untersucht werden. Es wurde für jedes Netzwerk der Adam-Optimierer mit einer Lernrate von 0,0001 verwendet, dies sollte in der Zukunft noch tiefgreifender untersucht werden. Auch die Netzwerkstrukturen wie Feature-Größe und -Anzahl sowie der Anzahl der Schichten sollten in Zukunft nochmals genauer betrachtet werden. Beim Entwerfen eines Netzwerkes könnte zum Beispiel ein Backpropagation-Verfahren helfen, indem statt der Gewichte die Netzwerkstrukturparameter, wie zum Beispiel die Größe und Anzahl der Features oder die Anzahl der Schichten und Neuronen, automatisch optimiert werden. Auch dies könnte in einer zukünftigen Arbeit untersucht werden.

Es wurde auch angedacht, ein Nachhilfe-Lernen zu verwenden, in welchem vorgesehen war, nach einem bestimmten Lehrerfolg lediglich Lernbilder zu verwenden, die das Netzwerk bis dato falsch klassifiziert. Dies wurde zwar mit den Netzen ausprobiert, brachte hier jedoch keinen weiteren Vorteil. Jedoch könnte dies für andere Arbeiten interessant sein.

Während der Entstehung der Arbeit wurde das U-Net vorgestellt, das Elektronenmikroskop-Aufnahmen von biologischen Zellen auswertet. Das U-Net besteht aus einem Komprimierungspfad und einem Extraktionspfad, um ein gesamtes Bild pixelweise zu klassifizieren und zu segmentieren. Der räumlichen Bezug der Klassifikationsergebnisse wird durch direkte Verknüpfungen zwischen den beiden Pfaden beibehalten [Ronneberger, et al., 2015], [Shengyuan & Jiaming, 2019]. Die Architektur konnte durch die weit fortgeschrittene Arbeit leider nicht mehr auf das hier bestehende Problem getestet werden und soll in nachfolgenden Arbeiten untersucht werden.

4.6 Optimierung des Klassifikators

4.6.1 Berechnungsoptimierung

Für die Klassifikation aller Objekte in einem Bild existieren verschiedene Ansätze. Ein möglicher Ansatz wäre, damit zu beginnen, die einzelnen Objekte erst im Bild zu lokalisieren und anschließend zu klassifizieren. Diese zwei Schritte sind jedoch nicht immer möglich, so auch hier bei der Klassifikation von Pflanzen. Es ist nicht trivial, zu unterscheiden, wo genau im Bild eine Pflanze aufhört und eine andere beginnt. Die hier verwendete Methode des CNN betrachtet daher jeden Pixel für sich und klassifiziert so das gesamte Bild. Das Problem ist dabei die Echtzeit. Die Echtzeit ist hier die Geschwindigkeit, in der die Kamera die Bilder aufnehmen muss, damit bei der Translation des Roboters eine lückenlose Bildaufnahme des Feldes entsteht. Die Echtzeit soll daher vorerst bei 10 Bildern pro Sekunde liegen. Die folgende Optimierung wurde in [Knoll, et al., 2017] und in [Knoll, et al., 2019] veröffentlicht.

Eine andere Methode zur Detektion und Klassifikation bietet das Yolo-Netzwerk. Es wurde wie das U-Net während der Entstehung dieser Arbeit vorgestellt. Yolo steht für You Only Look Once. Das Netzwerk teilt das Bild in ein Raster ein. Für jedes Rasterelement versucht das Netzwerk mindestens zwei Boundingboxen von interessanten Objekte anzugeben. Weiter gibt es die Klassenwahrscheinlichkeit und die Wahrscheinlichkeit für die Richtigkeit der jeweiligen Boundingbox an. Eine nachfolgende Analyse der einzelnen Ergebnisse der verschiedenen Elemente erzeugt eine Liste von Boundingboxen der gefundenen Elemente und die dazugehörige Klasse. Das Yolo-Netzwerk schafft in einem Test eine FPS von 155 mit 20 Klassen [Redmon, et al., 2016]. Erste Untersuchungen zeigen, dass bei der Pflanzenklassifikation nicht die Genauigkeit erreicht wird wie bei den hier vorgestellten Klassifikatoren. Weil das Yolo-Netzwerk erst zu einem Zeitpunkt veröffentlicht wurde, zu dem diese Arbeit schon sehr weit fortgeschritten war, wurde beschlossen es erst in nachfolgenden Arbeiten genauer zu untersuchen.

4.6.1.1 Erste Optimierung der Abarbeitung des Convolutional-Neural-Network

Das Netz verwendet ein Eingangsbild in der Größe von 101 x 101 Pixel, das aus dem gesamten Kamerabild extrahiert wird. Auf dieses Eingangsbild werden nun die Elemente des CNN angewendet, also die Convolution-Schichten, die Pooling-Schichten und die Dense-Schichten. Anschließend gibt der Ausgang des Netzwerkes die Klassenzugehörigkeit des mittleren Pixels des Eingangsbildes an. Damit das gesamte Kamerabild klassifiziert werden kann, muss dieser Vorgang für jeden Pixel im Kamerabild wiederholt werden. Die Kamerabilder besitzen eine Full-HD Auflösung, also 1080 x 1920 Pixel, mit über 2 Millionen Klassifikationen. Das CNN müsste also zwei Millionen Mal berechnet werden, um alle Pixel zu klassifizieren. Das Ausgangsproblem ist jedoch nur die Klassifikation der Pflanzen an sich. Im Durchschnitt enthält ein solches Bild etwa 7% Pflanzenpixel⁴⁸, das entspricht im Schnitt etwas weniger als 150.000 Pixel. Mittels vorgestellter Segmentierung⁴⁹ kann herausgefunden werden, ob es sich bei einem Pixel um ein Pflanzenpixel oder um ein Hintergrundpixel handelt. Diese Segmentierung ist von der Berechnung her nicht so komplex wie ein CNN und daher schneller abzuarbeiten. Der entwickelte HSV-Segmentierer verwendet die Formel (106)⁵⁰:

⁴⁸ siehe dazu Kapitel 3.6.1

⁴⁹ siehe dazu Kapitel 3.6.6

⁵⁰ Die Herleitung der Formel befindet sich im Kapitel 3.6.6, die Formel selbst ist in diesem Kapitel unter der Formel (86) eingeführt worden.

$$S(x,y)' = \begin{cases} 1 & \text{wenn } \frac{\min(\psi_{Rot}(x,y), \psi_{Grün}(x,y), \psi_{Blau}(x,y))}{\psi_{Grün}(x,y)} \leq \frac{255 - THs}{255} \\ 0 & \text{sonst} \end{cases} \quad (106)$$

Das segmentierte Bild $S(x,y)'$ wird über die Aufteilung des RGB Bildes $\psi(x,y)$ in seine drei Bestandteile Rot, Grün und Blau aufgeteilt. Ein Pixel des segmentierten Bildes wird dann auf 1 gesetzt, wenn der minimale Wert von Rot, Grün oder Blau eines Pixel durch den grünen Pixel geteilt, kleiner oder gleich einem Schwellwert (THs) ist, wobei der Schwellwert in dem 8 Bit Raum des Bildes über die Skalierung mittels 255 erfolgt. Die Höhe des Schwellwertes wird in Kapitel 3.6.6 genauer erläutert und soll hier nicht wiederholt werden. Sollte der Schwellwert des Pixels nicht erreicht werden, wird er auf 0 gesetzt.

Damit ergibt sich die erste Optimierung: Bevor das gesamte Bild in zwei Millionen Bilder zerlegt wird, wird die Segmentierung angewendet. Das heißt, das gesamte Bild wird durchlaufen und mittels Formel (106) entschieden, ob es sich um ein Pflanzenpixel handelt oder nicht. Zum einen wird das Bild segmentiert, das heißt der Hintergrund wird auf Schwarz (Value = 0) gesetzt. Zum anderen werden, sofern es sich um ein Pflanzenpixel handelt, dessen Pixelkoordinaten in eine Liste geschrieben. Anschließend werden nur noch Bildfragmente dieser Koordinaten in das CNN gegeben, die auch in dieser Liste stehen, also nur die relevanten Pflanzenpixel. Die unnötigen Pixel des Erdbodens zum Beispiel entfallen. Damit wird das CNN etwa 15-Mal weniger aufgerufen.

4.6.1.2 Zweite Optimierung der Abarbeitung des Convolutional-Neural-Network

Wie bereits erwähnt, wird durch die Segmentierung der Hintergrund auf den Wert 0 gesetzt. Auch die Bildelemente, die das CNN betrachten, besitzen nun segmentierte Bilder. Normalerweise würde auf einer Convolution-Schicht die Featureberechnung auf jeden Pixel des Bildelementes angewendet werden. Damit ergeben sich jedoch drei verschiedene Fälle für die Berechnung, die in Abbildung 113 dargestellt sind, jeweils für ein Beispiel der Featuregröße von 5x5.

Der rote Fall zeigt eine Featureberechnung, in der das Feature komplett auf dem Hintergrund liegt. Hier wird jedes Element mit 0 multipliziert, was zur Folge hat, dass die gesamte Berechnung 0 ergibt, beziehungsweise den Biaswert. Das Ergebnis dieser Berechnung ist daher schon vor der Rechnung bekannt. Selbst wenn der Hintergrund (Erdboden) nicht den Wert Null hätte, würde diese Berechnung keine Information über die Pflanze beinhalten, daher kann das Ergebnis einfach ein konstanter fiktiver Wert sein.

Im grünen Fall liegt der mittlere Featurewert nicht auf einer Pflanze. Das heißt, dass ein Teil ebenfalls eine Multiplikation mit Null ist. Dieser Fall verzerrt die Pflanze im Randbereich und wird dadurch in der Featuremap größer.

Im blauen Fall liegt mindestens der mittlere Pixel des Features auf einem Pflanzenpixel.

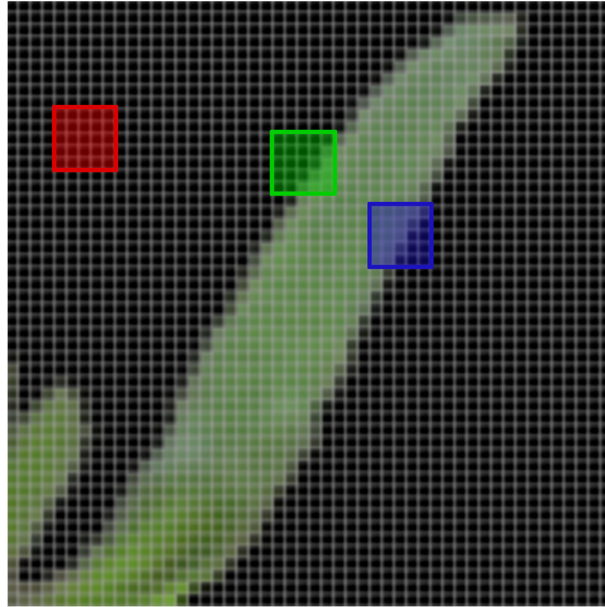


Abbildung 113: Fallunterscheidung bei der Anwendung eines Features

Nach Betrachtung dieser drei Fälle müssen nur der grüne und blaue Fall berechnet werden. Also die Fälle, in denen das Feature mindestens einen Eingangswert ungleich null hat. Von allen anderen Featureberechnungen sind die Ergebnisse vor der Berechnung bekannt. Diese Ergebnisse sind Null, beziehungsweise sind lediglich der jeweilige Biaswert. Die Koordinaten, in denen der blaue Fall auftritt sind bekannt. Das sind die Koordinaten, die bei der Segmentierung gespeichert werden. Der grüne Fall benötigt wiederum eine Berechnung, ob dieser Fall eingetreten ist. Dies verlangt eine Überprüfung jedes in der Segmentierung gefundenen Pflanzenpixels. Diese Kontrolle hat einen zu großen Aufwand, da der grüne Fall nur im Randbereich einer Pflanze auftritt und daher soll dieser Fall ignoriert werden.

Daraus folgt die zweite Optimierung: Es soll die Berechnung dahingehend optimiert werden, dass die Featureberechnung (und alle anderen Elemente des CNNs) nur auf die gefundenen Pflanzenpixel angewendet wird.

4.6.1.3 Dritte Optimierung der Abarbeitung des Convolutional-Neural-Network

In Abbildung 114 ist schematisch dargestellt, wie sich zwei benachbarte Pflanzenpixel in ihrem Berechnungsfenster unterscheiden, links ein Pflanzenpixel und rechts der Nachbar-Pflanzenpixel. Der grüne Bereich stellt stilisierte Pflanzenpixel dar, die klassifiziert werden müssen. Der rote Bereich stellt das Fenster dar, das das CNN als Eingangsbild bekommt, um den orangefarbenen Pixel zu klassifizieren.

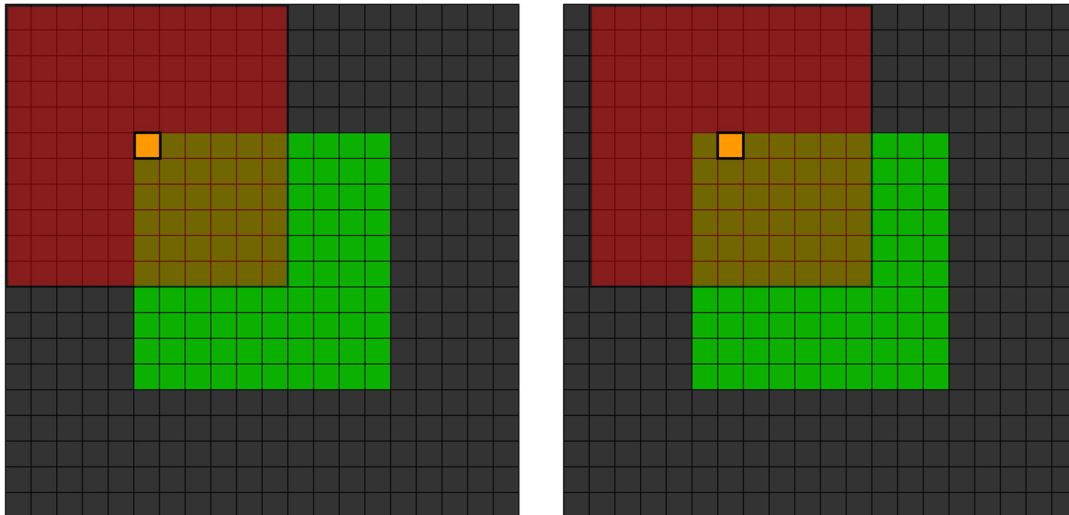


Abbildung 114: Visualisierung der Ausschnittsbetrachtung zweier benachbarter Berechnungen

Bei genauer Betrachtung kann festgestellt werden, dass die beiden Fenster (roter Bereich) sich stark überlappen. Das bedeutet wiederum, dass in beiden Fenstern größtenteils dieselben Werte stehen. Wenn nun das CNN das Feature in der Convolution-Schicht berechnet, würden auch aus der Featureberechnung dieselben Werte entstehen. In Abbildung 115 ist schematisch ein Feature mit der Größe von 5x5 Pixel in blau skizziert. Dieses Feature steht an denselben Koordinaten innerhalb des gesamten Bildes, jedoch ist es verschoben innerhalb des vom CNN zu betrachtenden Fensters (rot). Da der Ort jedoch im gesamten Bild derselbe ist, würde die Berechnung für den mittleren orangefarbenen Pixel denselben Wert sowohl im linken als auch im rechten Bild ergeben. Diese Erkenntnis lässt sich auf alle Elemente eines CNN anwenden. Dadurch kann, sofern der Randbereich ignoriert wird, die einzelne Featureberechnung zuerst auf das gesamte Bild angewendet werden. Theoretisch spielt die Zerlegung des Eingangsbildes erst mit der Dense-Schicht eine entscheidende Rolle. Die Dense-Schicht kann jedoch genauso wie eine Convolution-Schicht berechnet werden. Die Featuregröße ergibt sich dabei aus dem Zusammenspiel von Eingangsbildgröße und den vorhandenen Pooling-Schichten im Netzwerk.

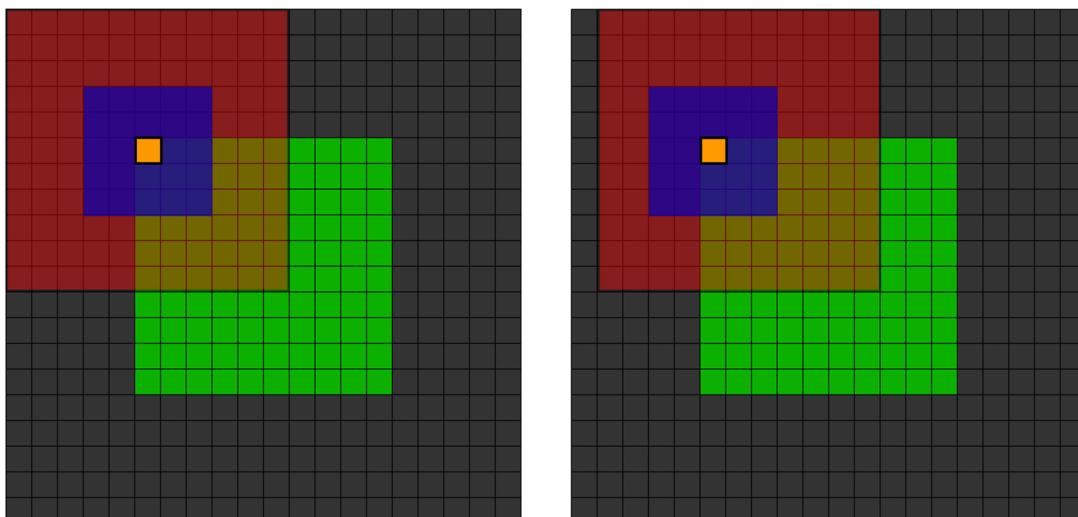


Abbildung 115: Visualisierung einer identischen Featureberechnung zweier benachbarter Berechnungen

So ergibt sich die dritte Optimierung: Es werden alle CNN Elemente (Convolution, Pooling und Dense) zuerst auf alle Pflanzenpixel des Kamerabildes angewendet. Eine Fensterbetrachtung von zum Beispiel 101x101 Pixel erfolgt nicht mehr direkt, sondern wird durch die Berechnung der Dense-Schicht erzeugt.

Diese Optimierungen wurden in einer eigens entwickelten CPU und als GPU Lösung mittels CUDA umgesetzt. Diese beiden Hardware Optimierungen werden nachfolgend beschrieben und vorgestellt.

4.6.1.4 Einfluss der Optimierung auf die Ergebnisse

Durch diese Optimierungen erfolgen jedoch auch Veränderungen am Ergebnis der Klassifikation. Den größten Einfluss haben hierbei die Pooling-Schichten. Bei jedem Pooling wird Information aus dem Netzwerk entfernt. Dadurch, dass das Kamerabild nicht mit Hilfe der Bildausschnitte (Fenster) betrachtet wird, verliert das Pooling jedoch seinen örtlichen Bezug. Die Abbildung 116 veranschaulicht das Problem.



Abbildung 116: Darstellung des Pooling Problems

In der Abbildung 116 ist jeweils ein Fenster als roter Rahmen dargestellt. Vor der Optimierung würde jedes Fenster einzeln durch das CNN laufen um seinen mittleren Pixel zu klassifizieren. Das rechte Fenster ist um einen Pixel weiter nach rechts verschoben. Die vier Farben Purpur, Blau, Gelb und Grün zeigen die einzelnen Anwendungen des Poolings an. Wie zu sehen ist, ergeben sich verschiedene Ergebnisse, da das Pooling immer am Rand beginnt und um die Poolinggröße (hier zwei Felder) weiter wandert. Dadurch ergeben sich aus zwei benachbarten Fenstern zwei verschiedene Pooling Ergebnisse. Daher würden sich, wenn dies in der Optimierung berücksichtigt werden soll, mit jedem Pooling vier neue Zweige für die weitere Berechnung ergeben. Da das Pooling einmal auf das gesamte Kamerabild angewendet wird, einmal mit Startpunkt oben links und ein weiteres Mal mit dem Startpunkt oben links, plus einem Pixel weiter. Über die Höhe ergeben sich zwei weitere Freiheitsgrade. In der weiteren Berechnung müssten beide Poolingergebnisse separat weiterverarbeitet werden. Bei einem weiteren zweiten Pooling würden sich wieder vier neue Pfade ergeben, so dass insgesamt 16 separate Ergebnisse berechnet werden müssten. Das Klassifikationsbild setzt sich anschließend aus den 16 verschiedenen Verzweigungen pixelweise rotierend durch die 16 Ergebnisse zusammen. Wird nur ein Pfad nach dem Pooling betrachtet, wäre das Ausgangsbild nach zweimaligem Pooling kleiner. Die Länge und Breite des Ausgangsbildes wäre dann jeweils nur $\frac{1}{4}$ so groß wie das ursprüngliche Kamerabild. Bei der Betrachtung aller Pfade würde etwa⁵¹ die Größe des Kamerabildes herauskommen.

Ein weiterer Unterschied stellen die fehlenden Randbereiche der Pflanzen dar⁵². Also die Pixel, in denen der Mittelpunkt des Features kein Pflanzenpixel ist, aber Pflanzenpixel im Feature vorkommen. Da die Features nicht auf alle Elemente angewendet werden, in denen sie irgendeine Überlappung mit der

⁵¹ je nachdem wie die Convolution-Schicht mit dem Randbereich verfährt

⁵² grüner Fall in Abbildung 113

Pflanze besitzen, existieren hier Berechnungsunterschiede. Auch dies kann einen Unterschied der Klassifikation im Gegensatz zur herkömmlichen Berechnung erzeugen. Das Problem könnte jedoch eliminiert werden, wenn bereits beim Training die Optimierung angewendet wird, so dass die Schichten lernen, dass die Informationen, die eine Featureberechnung in diesem Fall erzeugen würden, nicht weiter betrachtet werden können.

Die fehlende Berechnung der Featurewerte außerhalb der Pflanze⁵³ kann dahingehend andere Werte hervorrufen, dass das Ergebnis mit Null ausgewiesen wird, allerdings in Wirklichkeit der Biaswert ist. Dieser Fehler könnte ebenfalls durch die Berücksichtigung beim Training eliminiert werden.

Eine schwache Auswirkung stellt der fehlende Randbereich des nicht mehr existierenden Fensters dar. Untersuchungen zeigen, dass sich die einzelnen Werte im Randbereich von den Werten unterscheiden, die bei der normalen Berechnung entstehen. Dadurch, dass diese Anomalie bei der Dense-Schicht wieder herausgefiltert wird, hat dies keinen Einfluss auf das Klassifikationsergebnis. Dieser Effekt wirkt sich jedoch verstärkt bei kleinen Fenstergrößen aus, indem sich der Randbereich direkt auf den zu klassifizierenden Bereich der zu untersuchenden Pflanzenpixel auswirkt. Getestet werden konnte dies, indem die anderen Optimierungen ausgeschaltet wurden. Auch wenn dies keinen direkten Einfluss hatte, sollte es beim Lernen berücksichtigt werden.

4.6.2 Hardware Optimierung / Umsetzung im FPGA als CPU

Zur Klassifizierung der Pflanzen wurde eine CPU innerhalb eines FPGA selbst entwickelt und untersucht⁵⁴. Die CPU verwendet die im vorangegangenen Kapitel beschriebenen Optimierungen der Berechnungen, um eine effiziente Klassifikation zu ermöglichen. Ein wesentlicher Punkt liegt darin, dass durch die Unterteilung des Kamerabildes in kleinere Fenster immer wieder dieselben Berechnungen angestellt werden, da zwei benachbarte Fenster lediglich nur um einen Pixel verschoben sind. Normalerweise würden die Berechnungen für das vorherige Fenster verworfen werden, was aber ineffizient ist, da für das nächste Fenster erneut teilweise dieselben Berechnungen ausgeführt werden müssen. Daher wendet dieser Prozessor zuerst die Features auf das gesamte Kamerabild an und die Unterteilung in kleinere Fenster erfolgt erst mit der Dense-Schicht. Damit wird ein Großteil der Berechnungen eingespart. Genauer bedeutet dies für die Convolution-Schicht, dass der erste Featurewert des ersten Features mit dem gesamten Kamerabild multipliziert und in der Featuremap gespeichert wird. Im nächsten Schritt wird der zweite Featurewert des ersten Features mit dem Kamerabild multipliziert und zu dem jeweiligen Wert in der Featuremap addiert. Dieser Vorgang setzt sich daher im Wesentlichen aus einer MAC55 Einheit zusammen. Analog soll die Berechnung für die Pooling-Schichten und die Dense-Schichten erfolgen. Umgesetzt wurde die CPU mit Hilfe eines FPGA⁵⁶, der mit seinen Logikzellen eine optimale Arbeitsumgebung für das Entwickeln der CPU stellt. Der Vorteil der CPU liegt in der massiven und optimierten Parallelisierung für die Abarbeitung der CNN-Strukturen, sowie der geringe Energieverbrauch⁵⁷. Der entwickelte Prozessor besitzt den Aufbau wie in Abbildung 117, die einzelnen Module und ihre Arbeitsweise sollen im nachfolgenden erläutert werden.

⁵³ roter Fall in Abbildung 113

⁵⁴ Die entwickelte CPU des Autors wurde in [Knoll, et al., 2017] veröffentlicht und in der Abschlussarbeit von Michael Grelecke [Grelcke, 2017] umgesetzt.

⁵⁵ Multiply-Accumulate

⁵⁶ Field Programmable Gate Array

⁵⁷ Laut Aussage des FPGA Herstellers beträgt die Leistungsaufnahme circa 7W.

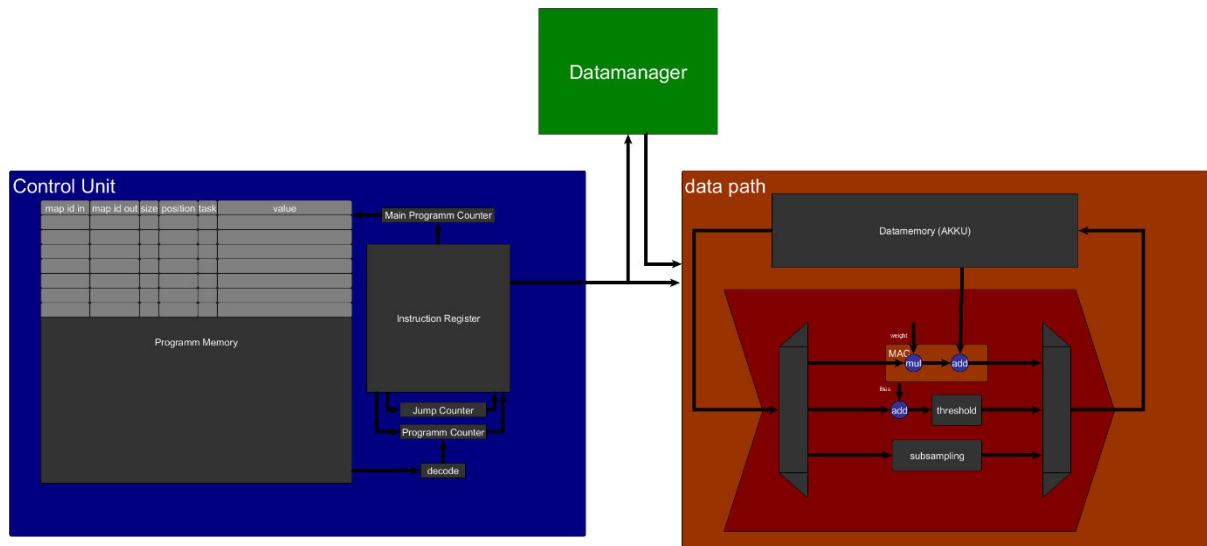


Abbildung 117: Aufbau der entwickelten CPU für die schnelle Berechnung der CNN Strukturen

Wie in Abbildung 117 dargestellt, besteht der vorzustellende Prozessor im Wesentlichen aus drei Bestandteilen. Aus der Control Unit, dem Datenmanager und dem Datenpfad. Die Aufgaben der einzelnen Bestandteile sind nachfolgend beschrieben.

4.6.2.1 Die Control Unit

Die Control Unit entscheidet, welchen Arbeitsschritt der Prozessor zu welchen Zeitpunkt abarbeiten soll. Im Programmspeicher ist jeder einzelne Arbeitsschritt, also das Programm oder besser gesagt, das CNN hinterlegt. Jeder Code (dargestellt als eine Zeile) im Programmspeicher besteht dabei aus 64 Bit. Diese 64 Bit teilen sich dabei wie folgt auf:

- 8 Bit Map ID in
- 8 Bit Map ID out
- 4 Bit Feature Size
- 8 Bit Feature Position
- 4 Bit Task
- 32 Bit Value

Die Map-Ids geben an, hinter welcher ID die Eingangsdaten gespeichert sind und hinter welcher ID das Ergebnis gespeichert werden soll. Die Feature Size beschreibt, wie groß der Arbeitsbereich ist, wobei immer von einem Quadrat ausgegangen wird, so dass die Feature Size lediglich die Größe einer Seite speichern muss. Die Feature Size könnte zum Beispiel die Größe eines Features, die Poolinggröße oder die Anzahl der Dense-Neuronen in einer Schicht beinhalten. Feature Position beinhaltet die lokale Position innerhalb der Operationsmethode, zum Beispiel die Position des jeweiligen Gewichts in einem Feature. Der Task gibt an, um welche Aufgabe (zum Beispiel: Feature-Gewicht-Berechnung, Feature Schwellwert, Pooling oder die Schritte für die Dense-Schicht und einige mehr)⁵⁸ es sich gerade handelt. Die 32 Bit Variable beinhaltet zum Beispiel das aktuelle Gewicht mit dem multipliziert werden muss

⁵⁸ Es könnte nun darüber diskutiert werden, ob diese 64 Bit nicht eine Verschwendung von Speicherplatz sind. Da zum Beispiel das Pooling keine 32 Bit Variable benötigt, könnte zumindest diese hier eingespart werden. Es zeigte sich jedoch, dass es wesentlich einfacher ist, immer von einem 64 Bit Befehl auszugehen und somit die Ressourcen in der Dekodierung zu sparen. Die Einsparung in diesem Bereich wäre zum gesamten Speicherverbrauch minimal.

oder den Schwellwert für die Schwellwerteinheit. Je nach Aufgabe werden unterschiedliche Speicherstellen in der Zeile verwendet oder leer gelassen.

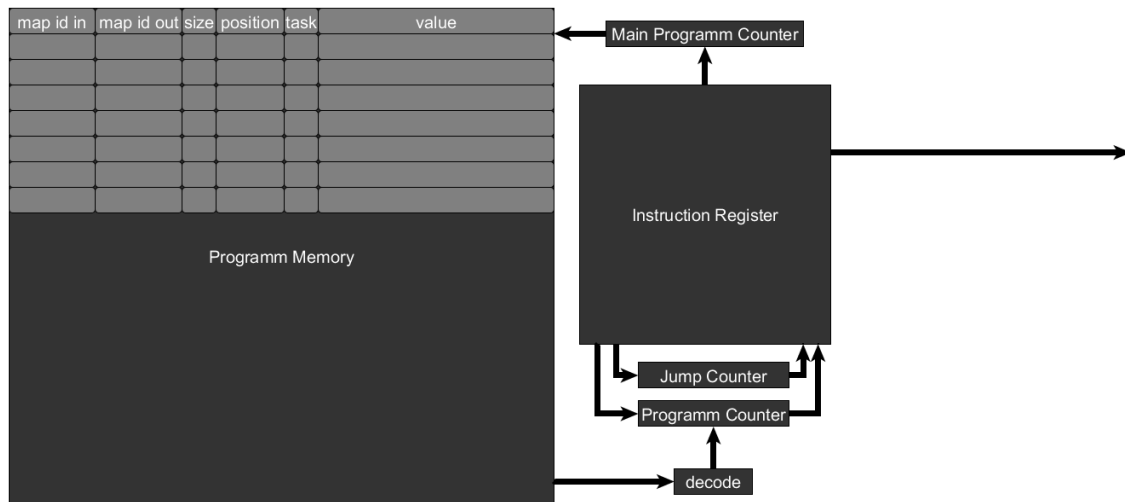


Abbildung 118: Detaillierter Aufbau der Control Unit

In Abbildung 118 ist die Control Unit des vorgestellten Prozessors nochmal genauer dargestellt. Das Herzstück des Prozessors ist das Instruction-Register. In diesem stehen die einzelnen Arbeitsschritte für jede Aufgabe untereinander. Mit jedem Schritt wird eine Programmzeile aus dem Programmspeicher geladen. Anschließend wird diese Programmzeile dekodiert, das bedeutet, dass der Task aus der Programmzeile extrahiert wird und in eine Adresse für das Instruction-Register übersetzt wird. Mit dieser Startadresse wird nun der Programmcounter geladen, so dass das Mikroprogramm im Instruction-Register den jeweiligen Befehl im Datenpfad steuern und abarbeiten kann. Nach jedem Schritt im Instruction-Register wird der Programm-Counter inkrementiert, damit das Mikroprogramm durchlaufen werden kann. Außerdem werden die anderen Bestandteile des Codes extrahiert. Der letzte Befehl des Mikroprogramms ist das Laden eines neuen Befehls, was damit beginnt, dass der Main-Programm-Counter inkrementiert wird, damit die Programmzeile aus dem Programmspeicher geladen werden kann. Zusätzlich besitzt die Control Unit einen Jump-Counter. Dieser ermöglicht es, Mikrobefehle zu wiederholen. Eine Berechnung mit einem Gewicht wird nicht nur einmal ausgeführt, sondern muss für jeden Pixel im Kamerabild erfolgen. Damit nun nicht für jeden Pixel eine neue Programmzeile im Programmspeicher belegt werden muss, wird der Befehl über den Jump-Counter mehrmals ausgeführt.

4.6.2.2 Der Datenmanager

Der Datenmanager beinhaltet im Wesentlichen die Memory Management Unit (MMU). Diese merkt sich, an welcher Speicheradresse sich welche Information befindet. Unter anderem sind dies die Startadressen der einzelnen Featuremaps. Außerdem soll der Datenmanager die Vorverarbeitung ermöglichen. Das heißt, er organisiert die Positionen der Pflanzenpixel, die er durch die Segmentierung des Kamerabildes herausfindet. Die Segmentierung soll dabei direkt beim seriellen (pixelweisen) Einlesen des Kamerabildes geschehen. Wie bereits oben beschrieben, ermöglichen die bekannten Positionen der Pflanzenpixel, die einzelnen Berechnungen nur auf diese Pflanzenpixel anzuwenden. Der Datenmanager ermöglicht eine effizientere Abarbeitung der Klassifikation, indem er unnötige Klassifikation von vornherein ausschließt. Da dies parallel passiert, wird diese Vorverarbeitung den Prozessor nicht stören.

4.6.2.3 Die Datenpfadeinheit

Im Datenpfad werden die eigentlichen Berechnungen durchgeführt. Ein Auszug aus der Datenpfadeinheit ist in Abbildung 119 dargestellt. Im AKKU sind alle Daten enthalten, die verarbeitet werden müssen. Wo genau die Daten im AKKU liegen, weiß der Datenmanager.

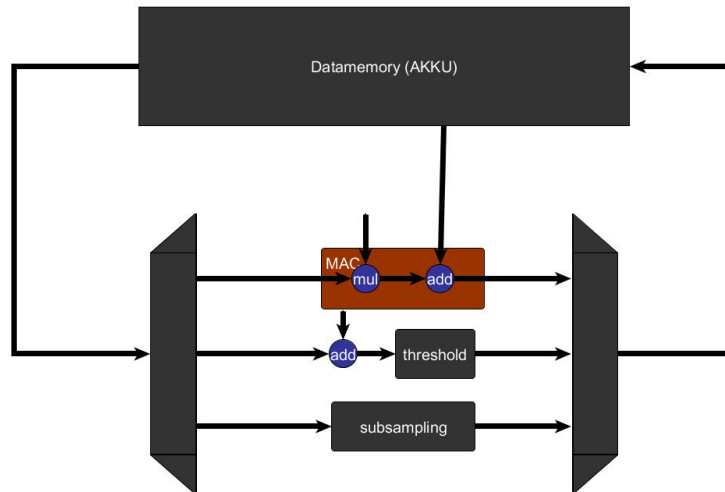


Abbildung 119: Detaillierte Darstellung der Datenpfadeinheit

Die Arithmetic Logic Unit (ALU) ist für die Berechnung zuständig. In Abbildung 119 sind auszugsweise die MAC-Einheit, der Schwellwert und das Subsampling dargestellt.

Die MAC-Einheit wird zum Beispiel für die Convolution- und Dense-Schichten verwendet. Diese multipliziert die gesamten Bilder mit ihren einzelnen Featuregewichten und addiert das vorherige Ergebnis hinzu. Wurde die MAC-Einheit für alle Gewichte des Features auf das gesamte Bild angewendet, stehen im AKKU sämtliche Summen der Featureberechnung. Um das Feature abzuschließen, muss jeder Wert nochmals geladen und auf die Schwellwerteinheit gegeben werden. Im AKKU steht nun die Featuremap, die als Eingang für das nächste Schicht verwendet wird. Das Subsampling ist das Modul, das die Pooling-Schicht berechnet. Die Dense-Schicht verwendet dieselben Module wie die Featureberechnung verwendet. Dadurch, dass das CNN massiv parallelisierbar ist, kann durch das Duplizieren der ALU ohne größeren Aufwand die Geschwindigkeit beim Berechnen potenziert werden.

4.6.2.4 Die Adressaufteilung des Speicherbereichs

Der FPGA besitzt jedoch nur einen richtigen Speicherblock. Daher müssen sämtliche Speicherbereiche sinnvoll zusammengefasst werden. Hier soll folgende Aufteilung verwendet werden: In die oberen Speicherplätze, also die niedrigen Adressen, soll der Programmspeicher geladen werden, da dieser von Anfang an bekannt ist und eine konstante Größe über die Laufzeit besitzt. Nach dem Programmspeicher folgt der Bild-Speicherblock, in dem die Zwischenwerte gespeichert werden. Dieser Speicher breitet sich ab dem Programmspeicher mit zunehmender Adresse nach unten aus. Der Datenmanager speichert seine MAP IDs, die Aussagen wo welche Daten im Speicher liegen, an der höchsten Adresse des Speichers und wächst nach oben. Die Speicherbelegung ist in Abbildung 120 dargestellt.

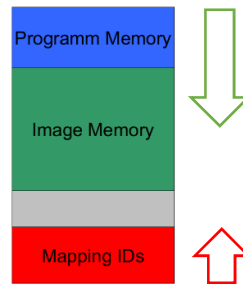


Abbildung 120: Darstellung der Adressaufteilung des Speichers

Ein Problem bei dieser Art der Speicheraufteilung ist es, dass es zu einem Fehler kommen kann, wenn der Speicherplatz insgesamt nicht ausreicht und der Datenmanager beginnt, Daten der Map IDs in den Bildspeicherbereich zu schreiben oder umgekehrt.

4.6.2.5 Test der CPU

Für den Test der CPU wurde vorerst ein einfaches CNN gewählt. Dieses Netzwerk wurde von LeCun in [LeCun, et al., 1995] vorgestellt und besitzt den in Abbildung 121 dargestellten Aufbau. Das Netzwerk wird als Lenet-1 bezeichnet und besteht aus 6 Schichten. Die Klassifikationsaufgabe des Netzwerks besteht in der OCR-Erkennung⁵⁹. Das Eingangsbild besitzt die Größe von 28x28 Pixel mit jeweils 8 Bit Farbtiefe. Es handelt sich daher um ein Graustufenbild. Darauf folgt eine Convolution-Schicht mit vier Features von der Größe 5x5 Pixel. Nach der Convolution-Schicht folgt ein Pooling und ein erneutes Anwenden einer Convolution-Schicht. Es handelt sich dabei um 12 Features mit der jeweiligen Größe von erneut 5x5 Pixel. Nach einem erneuten Pooling erfolgt über eine Dense-Schicht die Ausgangsschicht. Dieses Netzwerk wurde trainiert und erreichte eine Genauigkeit von 98,58% auf den frei verfügbaren MNIST Datensatz, der aus 70.000 handgeschriebenen Ziffernfolgen von 0 bis 9 besteht [LeCun, et al., 2019]. Wobei 60.000 Ziffernfolgen zum Training verwendet wurden und 10.000 unabhängige Ziffernfolgen zum Testen.

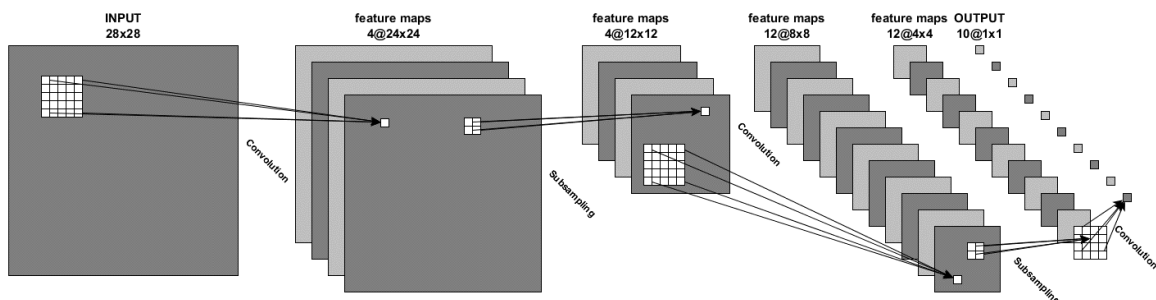


Abbildung 121: Aufbau des LeNet 1 zur Klassifikation von Schriftzeichen [LeCun, et al., 1995]

Das Netzwerk mit seinen Gewichten wurde im Programmspeicher der entwickelten CPU geladen und für die Aufgabe der Schriftzeichenerkennung angewendet. Als Hardwaregrundlage wurde das Altera Evaluations Board Terasic DE1-SoC Development Kit mit dem FPGA Altera Cyclone V SoC 5CSEMA5F31C6 gewählt [Altera, 2015].

⁵⁹ Optical Character Recognition

Die CPU verursacht im FPGA folgenden Hardwareverbrauch:

- Verwendete Logik Einheiten: 910 / 32.070 (3%)
- Anzahl der Register: 802
- Anzahl der Speicherblöcke (Bits): 2.097.152 / 4.065.280 (52%)
- Anzahl der DSP Blöcke: 5 / 87 (6%)

Der gesamte Prozessor benötigt 910 Logik Einheiten, wenn er nur eine ALU besitzt. Das entspricht etwa 3% der zur Verfügung stehenden Hardware auf dem FPGA. Es werden 802 Register benötigt und circa 2 Millionen Bits für den Speicher. Außerdem werden 5 DSP Blöcke verwendet, überwiegend zum Multiplizieren.

Der Speicherverbrauch des FPGA für das LENET-1 sieht dabei wie folgt aus:

- Programm Speicher: 208.896 Bit
- Eingangs-Bilder: 25.088 Bit
- Convolution: 73.728 Bit
- Leaky_Rectified: 0 Bit
- Pooling: 18.432 Bit
- Convolution: 24.576 Bit
- Leaky-Rctified: 0 Bit
- Pooling: 6.144 Bit
- Ausgangswerte: 320 Bit
- Map-ID Adressen: 35 Bit

Gesamt: 357.184 Bit / 4.065.280 (8,79%)

Damit lässt sich eine Geschwindigkeit für das LeNet-1 und seine einzelnen Schichten bei einer Taktfrequenz von 50 MHz wie folgt ermitteln:

- Convolution: 9,232 ms
- Leaky_Rectified: 0,369 ms
- Pooling: 0,369 ms
- Convolution: 12,480 ms
- Leaky-Rctified: 0,124 ms
- Pooling: 0,124 ms
- Ausgang: 0,617 ms

- Gesamt: 23,315 ms

Gesamt: ~42 Bilder / s

Dies bedeutet, dass der Prozessor für die Klassifikation von 28x28 Pixel großen Zahlen insgesamt 23,315ms benötigt. Dem entsprechen circa 42 Bilder pro Sekunde. Diese Berechnungen liefen dabei nur mit 50 MHz und besitzen noch keinerlei Optimierung, das bedeutet nur eine ALU und kein pipelining.

Die vorgestellte CPU wurde mit 10.000 Bildern getestet und erreichte dabei die gleiche Genauigkeit von 98,58% wie die Lösung am herkömmlichen Rechner. Dadurch, dass die Eingangsgröße lediglich 28x28 Pixel beträgt, konnten die weiteren vorgestellten Optimierungen, wie sie für die Pflanzenklassifikation ausgearbeitet worden sind, für die OCR Erkennung nicht angewendet werden.

Ein Test zur Pflanzenklassifikation ergab, dass der Speicherbereich im FPGA zu gering ist, um die gesamten Featuremaps zu speichern. Auch ein Versuch, die Daten aus dem FPGA nach jedem Schritt

auf einen externen Speicher zu schreiben, brachte keinen Erfolg. Hier zeigte sich auch, dass ein möglicher Datentransfer zwischen FPGA und externem Speicher jeden Geschwindigkeitsvorteil eliminieren würde. Jedoch konnte durch das schrittweise Überprüfen der Ergebnisse der verschiedenen Schichten des CNN überprüft werden, dass die Umsetzung auch für die Pflanzenklassifikation mit den weiter oben vorgestellten Optimierungen funktioniert, da die einzelnen Zwischenwerte identisch mit den Zwischenwerten am herkömmlichen Rechner für die gleiche Optimierung waren. Es war zeitlich nicht möglich, auf diese Weise eine größere Menge von Bildern für eine statistische Aussage generieren.

Ein FPGA-Board Hersteller sicherte zwar ein Evaluationsboard mit einem passenden größeren FPGA zu, leider ist dieser bis Ende dieser Arbeit jedoch nicht geliefert worden, so dass dies nicht weiter getestet werden konnte. Die Umsetzung mittels FPGA sollte jedoch weiter in zukünftigen Arbeiten verfolgt werden.

4.6.3 Hardware Optimierung / Umsetzung in der GPU

Die oben beschriebene Optimierung ist innerhalb einer GPU mittels CUDA-Programmierung umgesetzt worden⁶⁰. Durch die Optimierung werden die Hintergrundpixel, also die Erdpixel, auf Schwarz (Wert 0) gesetzt. Daher ist es besser, das CNN ebenfalls mit schwarzen Hintergrundpixel zu trainieren. Diese wichtige Anpassung am CNN selbst wurde bereits in dem Kapitel „Auswertung der Klassifikatoren“⁶¹ vorgenommen. Daher sollen diese trainierten Gewichte für das Netzwerk verwendet werden. Außerdem hat das Ausgangsbild der Optimierung durch die Pooling-Schichten nur noch ein Viertel der Größe des Eingangsbilds. Damit für den Vergleich zwischen der ursprünglichen Abarbeitung und der optimierten Abarbeitung gleiche Voraussetzungen gelten, soll bei der ursprünglichen Abarbeitung hier auch nur jeder vierte Pixel der Länge und jeder vierte Pixel der Höhe des Eingangsbilds klassifiziert werden. Das verwendete Netzwerk setzt sich zusammen aus einer Eingangsschicht mit der Größe von 101x101 Pixel, gefolgt von zwei Schichtpaaren aus Convolution und Pooling. Das erste Paar hat dabei die Dimension 16 und das zweite Paar die Dimension 32. Das Netzwerkende bilden drei Schichten mit je einer Anzahl von 64 Dense-Neuronen, die abschließend in zwei Ausgangsneuronen enden. Der komplette Aufbau einschließlich der Featuregrößen ist damit:

- Eingangsschicht: 3 x (101x101)
- Convolution: 16 x (5x5)
- Pooling: 16 x (2x2)
- Convolution: 32 x (7x7)
- Pooling: 32 x (2x2)
- Dense: 1 x (64)
- Dense: 1 x (64)
- Dense: 1 x (64)
- Ausgang 1 x (2)

Die verwendeten Aktivierungsfunktionen bildet jeweils die ReLu-Funktion in dem Netzwerk. Die erste Untersuchung soll die Genauigkeit der Optimierung untersuchen. Dafür wird dasselbe Netzwerk mit denselben Gewichten verwendet, um die Validierungsbilder zu klassifizieren. Einmal mit der herkömmlichen Berechnung und einmal mit der optimierten. Eine Gegenüberstellung der Klassifikationsergebnisse steht in Abbildung 122. Das linke Bild zeigt das Ergebnis der ursprünglichen Methode, das rechte Bild das Klassifikationsergebnis der optimierten Methode. Die grünen Pixel sind für die Klassifikatoren Karottenpixel und die roten Pixel Beikrautpixel. Auf die Darstellung eines Ground-Truth Bildes soll hier verzichtet werden. Bei genauer Betrachtung fällt auf, dass die optimierte

⁶⁰ Die erste Idee und Architektur der Software vom Autor wurde in der Abschlussarbeit von Vitali Czymmek [Czymmek, 2017] umgesetzt und konnte in [Knoll, et al., 2019] veröffentlicht werden.

⁶¹ Kapitel: 4.5.6

Methode innerhalb der Pflanze einige Pixel falsch klassifiziert. Bei der ursprünglichen Methode geschieht dies nur vereinzelt.

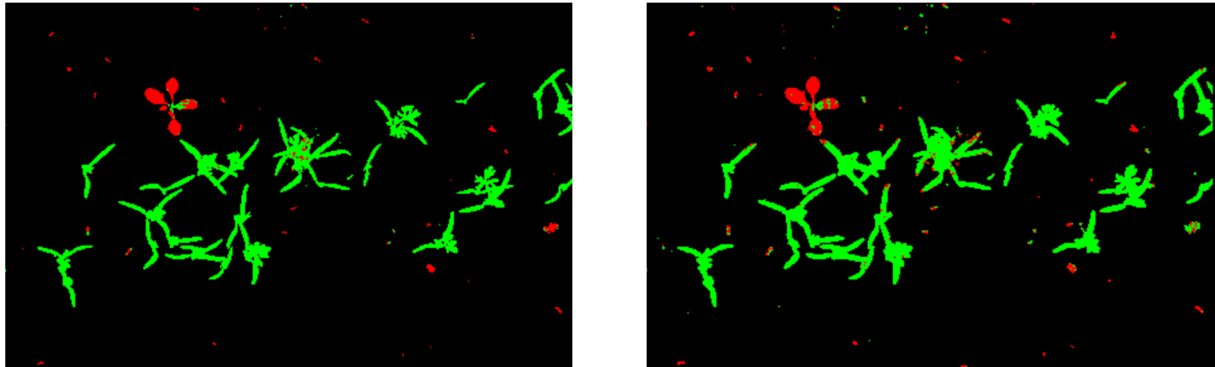


Abbildung 122: Ergebnisse der ursprünglichen Berechnung (links) und der optimierten Berechnung (rechts) desselben Bildes

4.6.3.1 Analyse der Genauigkeit der optimierten Methode

Die Ergebnisse der Analyse der globalen Genauigkeit der beiden Methoden ist in Tabelle 43 dargestellt. Die Zahlen zeigen, dass es von der Auswertung aller Pixel nur leichte Unterschiede in der Genauigkeit, Precision, im Recall und F1-Score zwischen der ursprünglichen und der optimierten Methode gibt. Die Zahlen wurden jeweils aus über fünf Millionen Klassifikationen berechnet. Wie gewohnt, wurde auch hier eine Skalierung zwischen den beiden Klassen verwendet.

Tabelle 43: Darstellung der Auswertungsergebnisse der globalen Genauigkeit für die ursprüngliche und optimierte Methode

Methode	Klasse	Genauigkeit	Precision	Recall	F1-Score
ursprüngliche	Karotte	0,92	0,90	0,94	0,92
	Beikraut		0,94	0,90	0,92
optimierte	Karotte	0,91	0,89	0,94	0,91
	Beikraut		0,94	0,89	0,91

Wird jedoch die Analyse über die Genauigkeiten innerhalb der 32.000 einzelnen Pflanzen betrachtet, wie es das Histogramm in Abbildung 123 zeigt, werden die Unterschiede sichtbar. Das linke Histogramm zeigt die Auswertung der ursprünglichen Methode und das rechte Histogramm die optimierte Methode.

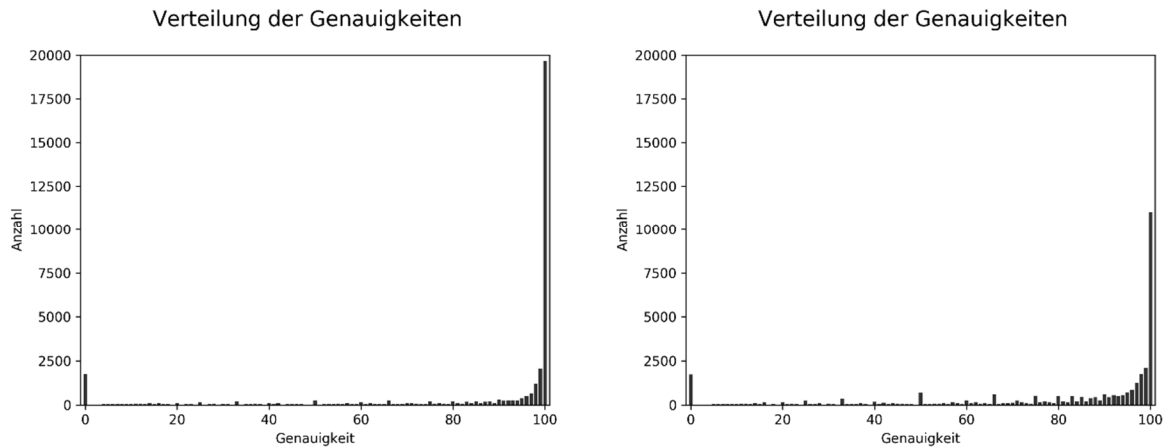


Abbildung 123: *Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen; Links: ursprüngliche Methode; Rechts: optimierte Methode*

Es zeigt sich, dass die Pflanzen, die in der Nähe der 100%-Klassifikationsgenauigkeit liegen, sich weiter verteilen und damit schlechtere Ergebnisse liefern. Die Vereinfachungen, die bei der optimierten Methode verwendet werden, verursachen diese Genauigkeitsreduktion. Das Problem könnte daran liegen, dass das verwendete Netz für eine andere Abarbeitung trainiert wurde und damit mit den Vereinfachungen falsche Schlüsse zieht. Daher soll das Netz erneut trainiert werden, jedoch direkt mit der Optimierung.

4.6.3.2 Trainieren des Netzwerks über die optimierte Methode der Abarbeitung

Das verwendete Tensorflow-Framework, das zum Training der Netze verwendet wird, ermöglicht keine einfache Methode zur Implementierung der optimierten Methode. Daher soll ein anderer Ansatz gewählt werden und mit Hilfe des Transfer-Trainings die optimierte Methode trainiert werden. Es sollen die bestehenden Gewichte, die mit der ursprünglichen Methode trainiert wurden, genauer gesagt: die Gewichte der ersten Convolution-Schicht, als Basis verwendet werden. Mit der optimierten Methode werden mit diesen Gewichten alle Lernbilder, die Schicht der Segmentierung, der ersten Convolution und des ersten Poolings durchlaufen. Die hier entstehenden Featuremaps werden für jedes der Lernbilder abgespeichert. Anschließend werden mit diesen Featuremaps das restliche Netzwerk, also die zweite Convolution-Schicht, das zweite Pooling sowie die Dense-Schichten trainiert. Nach dem Training werden die Gewichte der ersten Convolution aus dem ursprünglichen Training und die Gewichte der zweiten Convolution aus dem gerade beschriebenen Training zusammengesetzt. Die zusammengesetzten Gewichte werden nun mit der optimierten Methode verwendet, um die Lernbilder erneut zu verarbeiten, diesmal durch die Schichten der Segmentierung, der ersten Convolution-Schicht und der ersten Pooling-Schicht, sowie der zweiten Convolution- und der zweiten Pooling-Schicht. Auch diese Featuremaps werden ausgegeben und verwendet zum Training der Dense-Schichten und der Ausgangsschicht. Abschließend werden die Gewichte der Convolution-Schicht (aus ursprünglichen Trainings), die Gewichte der zweiten Convolution-Schicht (aus dem oben beschriebenen Training) und die Gewichte der Dense- und Ausgangsschicht (aus dem zuletzt beschriebenen Trainingsdurchlauf) zusammengesetzt für ein Netzwerk, das die optimierte Methode verwenden kann. Die Batchgröße wurde dabei wieder auf 64 festgesetzt und es wurde solange trainiert, bis keine Änderung der Genauigkeit mehr erfolgte (100.000 Epochen). Die Beispielklassifikationen sind in Abbildung 124 erneut gegenübergestellt. Auf der linken Seite ist die Klassifikation der optimierten Methode mit den ursprünglichen Gewichten dargestellt und auf der rechten Seite eine Beispielklassifikation der optimierten Methode mit neu trainierten Gewichten. Auch hier sind die grünen Pixel die vermuteten Karotten und die roten Pixel sind Pixel von Beikräutern.

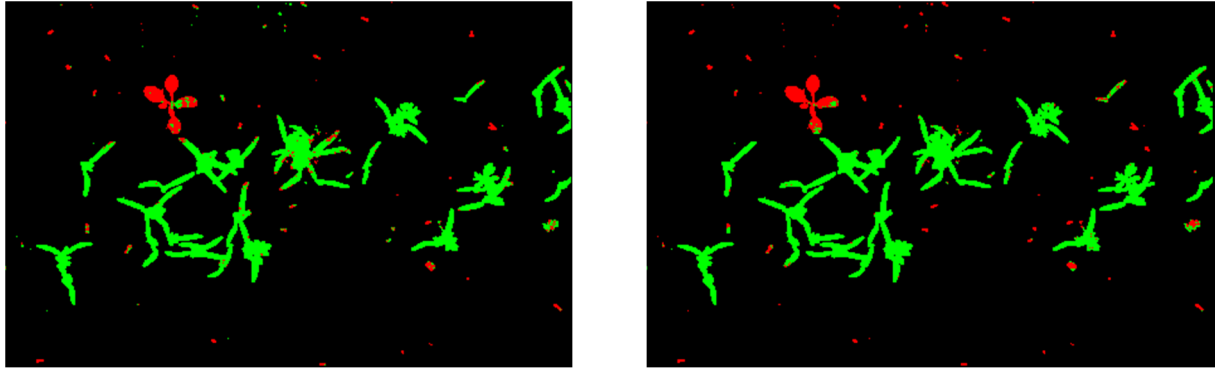


Abbildung 124: Ergebnisse der optimierten Berechnung mit den herkömmlichen Gewichten (links) und der optimierten Berechnung mit angepassten Gewichten (rechts)

Die Bilder zeigen, dass die Klassifikationsergebnisse mit den neuen Gewichten weniger Fehlklassifikationen innerhalb der Pflanzen aufzeigen. Auch hier soll eine Gegenüberstellung der globalen Genauigkeit erfolgen. Die Ergebnisse sind in Tabelle 44 aufgelistet. Es wurde eine Skalierung zwischen den beiden Klassen verwendet.

Tabelle 44: Darstellung der Auswertungsergebnisse der globalen Genauigkeit für die ursprüngliche und optimierte Methode und die optimierte Methode mit angepassten Gewichten

Methode	Klasse	Genauigkeit	Precision	Recall	F1-Score
ursprüngliche	Karotte	0,92	0,90	0,94	0,92
	Beikraut		0,94	0,90	0,92
optimierte	Karotte	0,91	0,89	0,94	0,91
	Beikraut		0,94	0,89	0,91
optimierte (trainiert)	Karotte	0,92	0,91	0,94	0,92
	Beikraut		0,93	0,90	0,91

Die Werte zeigen nur leichte Unterschiede. Interessanter ist die Auswertung der lokalen Genauigkeiten. Die Pixelgenauigkeit innerhalb der einzelnen Pflanzen ist in Abbildung 125 abgebildet. Das linke Histogramm zeigt die Verteilungen der Genauigkeiten der optimierten Methode mit den ursprünglichen Gewichten und das rechte Histogramm die Verteilung der optimierten Methode mit den angepassten Gewichten. Die Histogramme zeigen, dass die Werte der Pixelgenauigkeiten der einzelnen Pflanze wieder auf das Extrem der zu 100% richtigen Klassifikation streben.

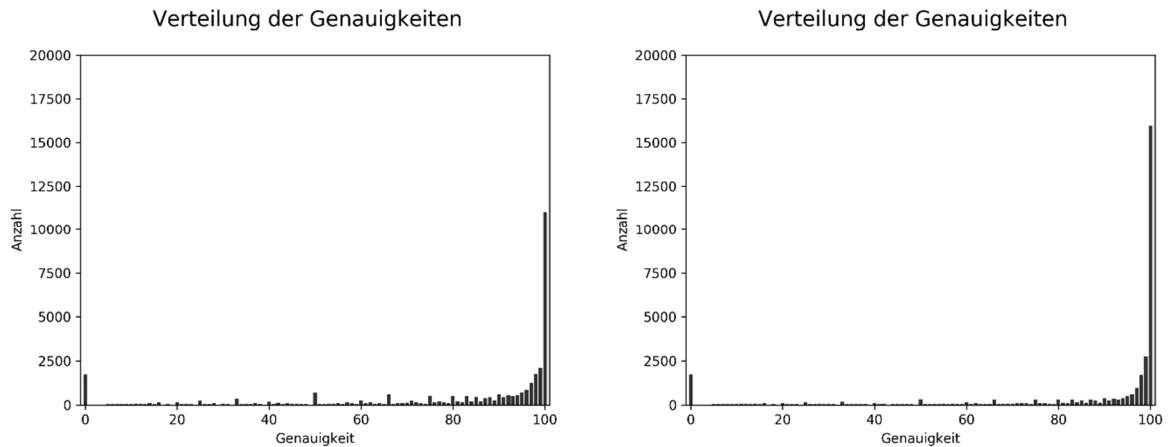


Abbildung 125: Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen; Links: optimierte Methode mit ursprünglichen Gewichten; Rechts: optimierte Methode mit angepassten Gewichten

4.6.3.3 Geschwindigkeitsanalyse der optimierten Methode

Als abschließende Untersuchung soll die Geschwindigkeit der optimierten Methode untersucht werden. Zur Analyse wurde für jedes der 1075 Validierungsbilder die Zeit für die Segmentierung und die Klassifikation gemessen. Die Segmentierung ist bei der Messung enthalten, da diese für beide Methoden ein wichtiger Bestandteil ist. Neben dem Setzen der Hintergrundpixel auf Schwarz wird der Segmentierer verwendet, um die Koordinaten der Pflanzenpixel zu ermitteln, damit nur noch diese Pixel dem Klassifizierer übergeben werden müssen. Genauer beginnt die Zeitmessung nach dem Laden des Bildes und endet nach der Erstellung des Ausgangsbildes. Die Zeiten sind in dem Zeit-Klassifikations-Diagramm in Abbildung 126 eingetragen. Das Diagramm zeigt auf der Abszisse die benötigte Zeit in Millisekunden und auf der Ordinate die Anzahl der Klassifikationen in einem Bild. Die roten Punkte sind die 1075 Bilder der herkömmlichen Methode, die blauen Punkte die 1075 Bilder der optimierten Methode. Für die Messung wurde die gleiche Hardware verwendet, welche unter anderem eine Nvidia GTX 1080 Titan Grafikkarte beinhaltet.

Zeit Klassifikations Diagramm

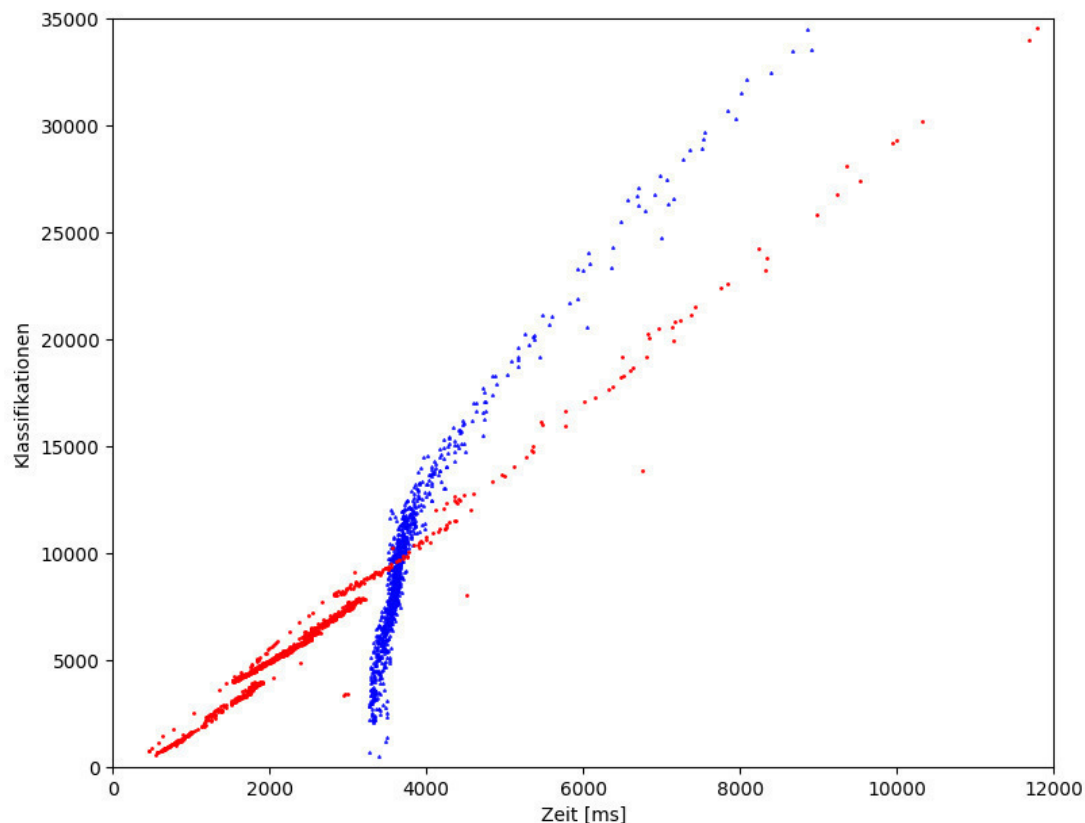


Abbildung 126: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) im Zeit-Klassifikations-Diagramm

Das Diagramm zeigt, dass die herkömmliche Methode sich in etwa linear verhält. Die optimierte Methode hingegen scheint eine Grundberechnungszeit zu benötigen und geht dann in einen linearen Verlauf über. Daraus folgt, dass die optimierte Methode erst ab circa 10.000 Klassifikationen (Pflanzenpixel) schneller ist, als die herkömmliche Methode. Das Bild zeigt weiter, dass die meisten Bilder weniger als 10.000 Pflanzenpixel besitzen. Dadurch, dass nur jeder vierte Pixel in der Höhe und der Weite ausgewertet wird, betrifft die Anzahl der Pflanzenpixel nicht das ursprüngliche Kamerabild, sondern bildet die Anzahl der klassifizierten Pixel am Ausgangsnetz. Zur besseren Analyse des Zeitverhaltens sollen die Zeiten der größten Schichten im Netz separat dargestellt werden. In der Abbildung 127 sind die Convolution-Schichten und die Dense-Schichten dargestellt. Die erste Convolution-Schicht bilden die blauen Punkte, die zweite Convolution-Schicht ist grün dargestellt, die erste Dense-Schicht rot und die Dense-Schichten zwei und drei sind violett. Die Segmentierung, die Pooling-Schichten und das Ausgangsnetzwerk wären links der ersten Convolution-Schicht und wurden aus Gründen der Übersichtlichkeit nicht mit eingezeichnet. Die zweite und dritte Dense-Schicht besitzen eine identische Anzahl von Berechnungen, daher sind diese auch auf derselben Zeitachse.

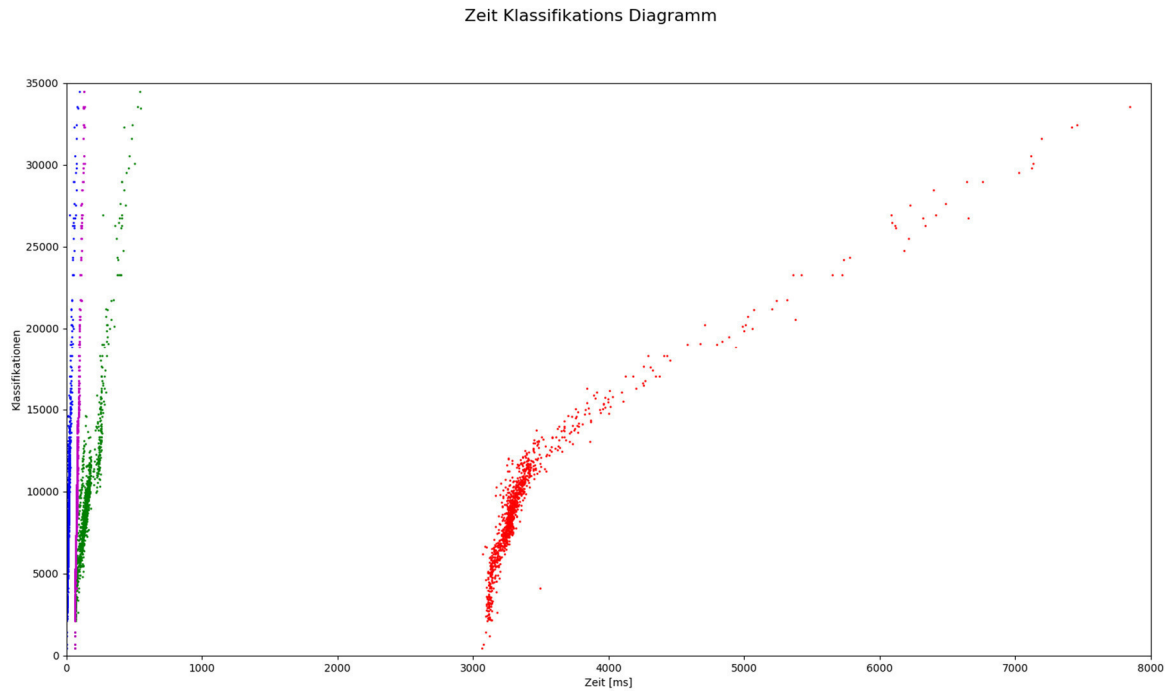


Abbildung 127: Darstellung des Zeitverhaltens der einzelnen Schichten im Netzwerk (erste Convolution-Schicht blau, zweite Convolution-Schicht grün, erste Dense-Schicht rot, zweite und dritte Dense-Schicht violett)

Es ist ersichtlich, dass die erste Dense-Schicht (rote Punkte) fast die gesamte Verarbeitungszeit beansprucht. Dies hat zwei Ursachen: Zum einen bildet diese erste Dense-Schicht die meisten Berechnungen in der Struktur, zum anderen ist die Umsetzung im Code noch nicht optimal. Die Parallelisierung ist hier nicht vollständig ausgereizt. Es werden zwar die Berechnungen der einzelnen Pflanzenpixel komplett parallelisiert, das Aufrufen der Parallelberechnung geschieht jedoch noch seriell. Das Netzwerk besitzt eine Dimension von 32 in der letzten Convolution-Schicht und eine Dimension von 64 in der ersten Dense-Schicht. Dies bedeutet, dass 32 mal 64 Dense-Berechnungen aufgerufen werden, die dann parallel über alle Pflanzenpixel abgearbeitet werden. Dies sollte in einer zukünftigen Version geändert werden, sofern es die Hardwareauslastung ermöglicht.

Im nächsten Schritt soll die Netzwerkstruktur an der optimierten Methode angepasst werden.

4.6.4 Optimierung der Netzwerkstruktur

Im vorherigen Kapitel wurde die optimierte Methode zum Abarbeiten mit der bestehenden Netzwerkstruktur untersucht. Die Genauigkeitsergebnisse waren dabei fast identisch, jedoch konnte die Geschwindigkeit noch nicht 10 Bilder pro Sekunde erreichen. Daher soll hier nun die Netzwerkstruktur an die optimierte Methode angepasst werden. Die ausgehende Netzwerkstruktur hat folgenden Aufbau:

- Eingangsschicht: 3 x (101x101)
- Convolution: 16 x (5x5)
- Pooling: 16 x (2x2)
- Convolution: 32 x (7x7)
- Pooling: 32 x (2x2)
- Dense: 1 x (64)
- Dense: 1 x (64)
- Dense: 1 x (64)
- Ausgang 1 x (2)

Außerdem ist in Abbildung 127 dargestellt, welche zeitlichen Kosten welche Schicht verursacht. Der größte Part bildet hier die erste Dense-Schicht, daher soll diese im Hauptfokus der Optimierung stehen. Damit diese Schicht vom Aufbau verringert werden kann gibt es verschiedene Herangehensweisen. Die erste wäre, die Dimension, also die Anzahl der Neuronen in dieser Dense-Schicht zu verringern. Die andere läge darin, das Eingangsbild zu verkleinern. Dieses liegt derzeit bei einer Größe von 101x101 Pixel. Das Eingangsbild wird zwar in der optimierten Methode der Abarbeitung nicht mehr direkt verwendet, bildet aber die Größe der Dense-Schicht, wie in dem Kapitel⁶² erläutert. Die derzeitige Dense-Größe ist eine Fläche von 26x26 Pixel, die durch das zweimalige Pooling der 101x101 Pixel des Eingangsbildes zustande kommt. Wenn das Eingangsbild auf 51x51 Pixel reduziert würde, wäre die Dense-Größe nur noch 13x13 Pixel groß, das bedeutet nur noch ein Viertel der ursprünglichen Größen. Außerdem soll die Dimension auf 8 Neuronen verringert werden. Als nächstes wird die zweite Convolution-Schicht von 7x7 Pixel der Featuregröße auf 5x5 und auch die Dimension von 32 Features auf 8 Features reduziert. Ebenfalls soll die erste Convolution-Schicht von der Featureanzahl halbiert werden, so dass auch diese eine Dimension von 8 besitzt. Der letzte Punkt sind die Dense-Schichten zwei und drei, die mehr Zeit als die erste Convolution-Schicht benötigen und daher entfernt wurden⁶³. Damit ergibt sich die neue Netzwerkstruktur zu:

- Eingangsschicht: 3 x (51x51)
- Convolution: 8 x (5x5)
- Pooling: 8 x (2x2)
- Convolution: 8 x (5x5)
- Pooling: 8 x (2x2)
- Dense: 1 x (8)
- Ausgang 1 x (2)

Die allgemeine Aktivierungsfunktion bleibt die ReLu-Funktion im Netzwerk. Die Netzwerkstruktur wurde einmal mit der ursprünglichen Methode neu trainiert. Auch hier wurde wieder eine Batchgröße von 64 gewählt mit 100.000 Epochen. Das Netzwerk wurde für die Optimierung mit denselben Parametern trainiert. Dabei wurden bei beiden Trainingsdurchläufen die Hintergründe durch den Segmentierer auf Schwarz gesetzt.

Zuerst soll betrachtet werden, wie sich die Netzwerkstruktur mit der optimierten Methode zur Abarbeitung mit den herkömmlichen Gewichten verhält. Dafür werden die 1075 Bilder jeweils mit der herkömmlichen und der optimierten Methode mit denselben Gewichten verarbeitet. Die Ergebnisse sind in Abbildung 128 dargestellt. Es ist zu erkennen, dass die optimierte Methode (rechtes Bild) viele falschklassifizierte Pixel erzeugt, während die herkömmliche Methode (linkes Bild) nur wenige Fehler aufweist.

⁶² Kapitel 4.6.1.3

⁶³ Die Veränderungen des Netzwerkes wurden durch experimentelles Herantasten bestimmt.

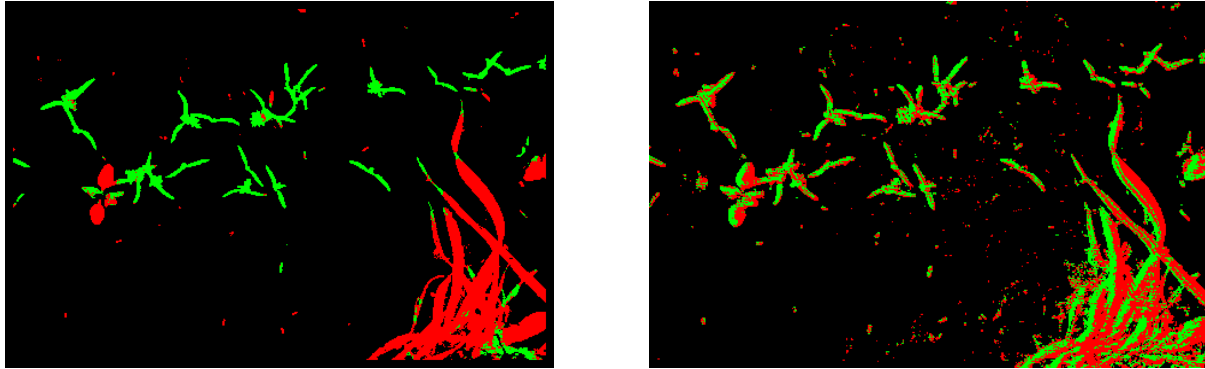


Abbildung 128: Klassifikationsergebnis mit der optimierten Netzwerkstruktur und der herkömmlichen (links) und der optimierten (rechts) Methode mit denselben Gewichten.

In den Bildern sind die grünen Pixel wieder vom Klassifikator angenommene Karotten-Pixel und rote Pixel vermutete Beikraut-Pixel. Die Auswertung der Pixelgenauigkeiten der globalen Genauigkeits-Analyse steht für die Ergebnisse der 1075 Validierungsbilder in Tabelle 45. Auch hier wurde wieder eine Skalierung zwischen den beiden Klassen verwendet.

Tabelle 45: Darstellung der Auswertungsergebnisse der globalen Genauigkeit für die ursprüngliche und optimierte Methode mit denselben Gewichten

Methode	Klasse	Genauigkeit	Precision	Recall	F1-Score
ursprüngliche	Karotte	0,92	0,90	0,95	0,92
	Beikraut		0,95	0,90	0,92
optimierte	Karotte	0,61	0,61	0,58	0,60
	Beikraut		0,60	0,63	0,62

Es wurden wieder jeweils über fünf Millionen Pflanzenpixel ausgewertet. Wie zu sehen ist, brechen die Werte für die optimierte Methode stark ein. Die Genauigkeit fällt auf 0,61, wohingegen die ursprüngliche Methode weiterhin bei 0,92 liegt. Auch die Werte der Precision, des Recall und des F1-Score fallen stark ab. Die Netzwerkstruktur hat zwar in Bezug auf die größere Netzwerkstruktur das Potential, dieselbe Genauigkeit zu erreichen, jedoch kommt die optimierte Methode damit nicht zurecht. Die lokale Genauigkeit der Pixelgenauigkeit innerhalb der einzelnen Pflanzen offenbart logischerweise auch ein Zusammenbrechen der Genauigkeiten. Die Verteilung ist wieder mittels Histogramm in Abbildung 129 dargestellt. Links ist wieder die herkömmliche und rechts die optimierte Methode abgebildet.

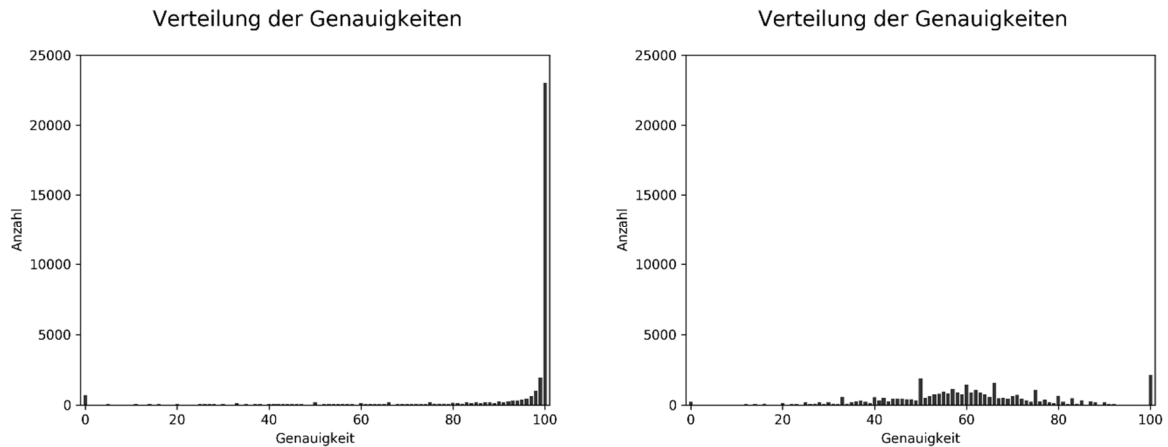


Abbildung 129: Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen mit dem optimierten Netzwerk; Links: ursprüngliche Methode; Rechts: optimierte Methode mit gleichen Gewichten

Das Ergebnis der optimierten Methode zeigt, dass die Pixelgenauigkeiten innerhalb der verschiedenen Pflanzen zusammenbrechen. Die Genauigkeiten der herkömmlichen Methode zeigen, dass die verringerte Netzwerkstruktur verwendet werden kann. Ein positiver Nebeneffekt der Verringerung des Eingangsbildes auf 51x51 ist, dass die total falschen Klassifikationen zurückgegangen sind, wie im Kapitel „Auswertung der Klassifikatoren“⁶⁴ vorhergesagt wurde.

Kleinere Netzwerkstrukturen reagieren scheinbar empfindlicher auf die optimierte Abarbeitungsfolge als größere Netzwerke. Das könnte daran liegen, dass es die größeren Strukturen durch massiveren parallelen Informationsfluss besser schaffen, fehlerhafte Zwischenergebnisse innerhalb des Netzwerkes auszugleichen. Daher soll hier nun das Netzwerk mit der optimierten Methode untersucht werden, jedoch unter Verwendung der angepassten Gewichte. Das Ergebnis ist in Abbildung 130 dargestellt.

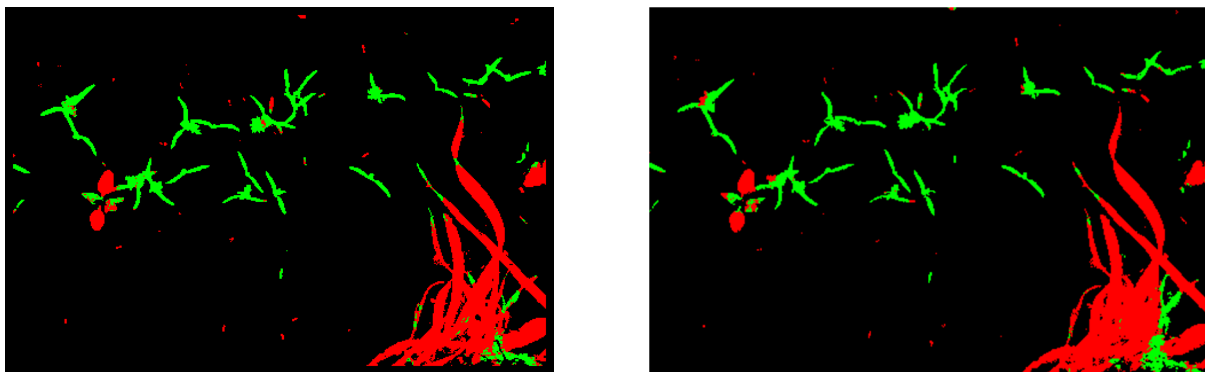


Abbildung 130: Klassifikationsergebnisse mit der herkömmlichen Netzwerkstruktur und der herkömmlichen (links) und der optimierten Methode mit angepassten Gewichten (rechts)

Im linken Bild ist die herkömmliche Methode verwendet worden, im rechten Bild die optimierte Methode mit den speziell trainierten Gewichten. Auch hier gilt, dass die grünen Pixel die Karottenpixel und die roten Pixel die Beikrautpixel für die Klassifikatoren sind. Die optimierte Methode zeigt ein ähnliches Verhalten der Güte im Ergebnis. Wird das Ergebnis in Zahlen ausgedrückt, sieht es wie in Tabelle 46 aus. In der Tabelle stehen die Werte der globalen Genauigkeitsanalyse. Hier werden alle Pixel aller Bilder zusammen für die statistischen Werte verwendet.

⁶⁴ Kapitel 4.5.6

Tabelle 46: Darstellung der Auswertungsergebnisse der globalen Genauigkeit für die ursprüngliche und optimierte Methode mit und ohne angepasste Gewichte

Methode	Klasse	Genauigkeit	Precision	Recall	F1-Score
ursprüngliche	Karotte	0,92	0,90	0,95	0,92
	Beikraut		0,95	0,90	0,92
optimierte	Karotte	0,61	0,61	0,58	0,60
	Beikraut		0,60	0,63	0,62
optimierte (trainiert)	Karotte	0,90	0,88	0,93	0,90
	Beikraut		0,92	0,87	0,89

Die Ergebnisse zeigen, dass sich die optimierte Methode mit den speziell trainierten Gewichten ähnlich verhält, wie die der ursprünglichen Methode. Abschließend soll noch eine Auswertung über die einzelnen Pflanzen erfolgen. Die Ergebnisse der lokalen Genauigkeitsanalyse sind wie gewohnt in den Histogrammen der Abbildung 131 hinterlegt. Die Genauigkeitsverteilung der herkömmlichen Methode ist im linken Histogramm und die der optimierten Methode mit den speziellen angepassten Gewichten im rechten Histogramm dargestellt. Wie zu sehen ist, haben sich die einzelnen Genauigkeiten der optimierten Methode verbessert, auch wenn diese noch nicht das Niveau der herkömmlichen Methode erreicht haben. Dies könnte mit einem intensiveren Training eventuell noch verbessert werden. Auch könnte das Netzwerk wieder etwas vergrößert werden, damit es einzelne fehlerhafte Werte im Informationsfluss des Netzes besser kompensieren kann. In Zahlen bedeutet dies, dass 72% aller Pflanzen mindestens eine Pixelgenauigkeit von 0,90 besitzen. Im Vergleich ergibt die herkömmliche Methode, dass 85% aller Pflanzen mindestens eine Pixelgenauigkeit von 0,90 aufweisen.

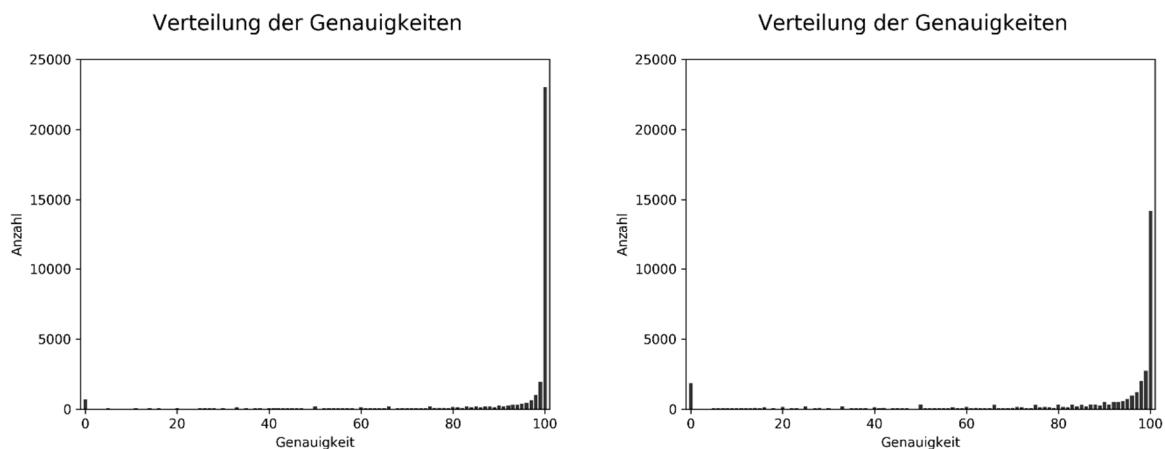


Abbildung 131: Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen; Links: ursprüngliche Methode; Rechts: optimierte Methode mit angepasstem Training

Die Geschwindigkeitsverteilung steht im Zeit-Klassifikations-Diagramm der Abbildung 132. Es wurden von beiden Methoden wieder die 1075 Validierungsbilder ausgewertet. Dabei wurde die Zeit für die Segmentierung und Klassifikation gemessen. Die blauen Punkte sind die Bilder der optimierten Methode und die roten Punkte sind die Bilder, die mit der herkömmlichen Methode ausgewertet wurden. Auf der Abszisse ist die Zeit, die für ein Bild benötigt wird in Millisekunden dargestellt und die Ordinate stellt die Anzahl der Klassifikationen in einem Bild dar.

Zeit Klassifikations Diagramm

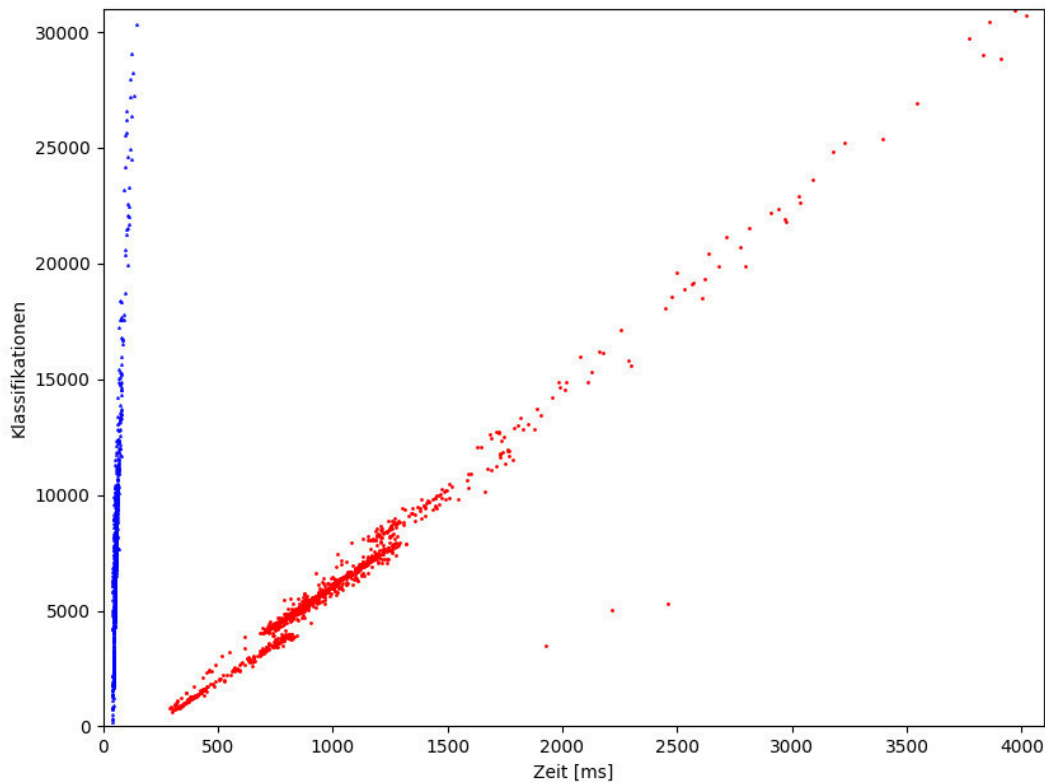


Abbildung 132: Zeit-Klassifikations-Diagramm der optimierten Netzwerkstruktur mit der herkömmlichen Methode (rot) und der optimierten Methode (blau) über die 1075 Validierungsbilder

Die herkömmliche Methode in Rot verhält sich wieder linear. Die blauen Punkte der optimierten Methode zeigen ebenfalls augenscheinlich ein lineares Verhalten. Es könnte jedoch davon ausgegangen werden, dass immer noch eine Basiszeit benötigt wird, diese jedoch durch die optimierte Netzwerkstruktur in der allgemeinen Zeit untergeht. Während die herkömmliche Methode im besten zeitlichen Fall mindestens circa 250 Millisekunden benötigt, benötigt die optimierte Methode im schlechtesten zeitlichen Fall gerade mal 100 Millisekunden. Damit ist eine FPS von 10 Bildern pro Sekunde möglich, die die Echtzeitfähigkeit des Systems erfüllt. Für 90% der Bilder ist sogar eine FPS zwischen 18 und 23 Bildern pro Sekunde möglich, da hier die Anzahl der zu klassifizierenden Pixel gering genug ist.

4.6.5 Abschließende Auswertung zur Optimierung

Abschließend soll der Einfluss verschiedener Störungen auf die Klassifikationsmethoden untersucht werden. Wie bei der Segmentierung⁶⁵ sollen hier die Problemfälle der

- Unschärfe durch Translation der Kamera,
- Unschärfe durch einen falschen Fokus des Objektivs,
- Unter- und Überbelichtung und
- Unter- und Übersättigung der Farbe des Bildes

betrachtet werden. Diese vier Fälle kommen, wie bereits bei der Segmentierung, in der Praxis vor. Eine Untersuchung auf Zoomverhalten des Objektivs soll hier nicht vorgenommen werden. Denn erstens sind in den Lernbildern verschiedene Wachstumsstadien, also verschiedene Größen der Pflanzen hinterlegt und zweitens besitzt das Objektiv einen konstanten Abstand zum Feld und keinen Zoom, so dass dieses Problem nicht vorkommt. Für diese Untersuchung wurden die 1075 Validierungsbilder künstlich in verschiedenen Stufen verändert. So kann eine Analyse nur auf die Störgröße erfolgen. Die untersuchten Methoden sind einmal die herkömmliche Bearbeitung und die optimierte Bearbeitung, wie sie in den vorherigen Kapiteln vorgestellt wurde. Dabei wurde für beide Methoden das folgende Netz verwendet:

• Eingangsschicht:	3	x	(51x51)
• Convolution:	8	x	(5x5)
• Pooling:	8	x	(2x2)
• Convolution:	8	x	(5x5)
• Pooling:	8	x	(2x2)
• Dense:	1	x	(8)
• Ausgang	1	x	(2)

Die allgemeine Aktivierungsfunktion im Netzwerk bleibt die ReLu-Funktion. Für die optimierte Methode der Abarbeitung wurden die angepassten Gewichte verwendet. In der Auswertung der Bilder ist die Segmentierung aktiv, so dass auch die Störung auf die Segmentierung wirkt. Die statistischen Größen wurden jedoch nur für die Pixel berechnet, für die auch ein Label vorliegt. Dies bedeutet, dass in dem betrachteten Fenster des Klassifikators eventuell eine unsaubere Segmentierung des Hintergrundes vorliegt, jedoch nicht die Erdbodenpixel in die statistische Auswertung einfließen. Weiter ist vom Eingangsbild nur jeder vierte Pixel der Höhe und jeder vierte Pixel der Breite des Bildes klassifiziert worden.

Zur Auswertung wird die globale Genauigkeitsanalyse verwendet, in der jeder Pixel unabhängig von jedem Bild oder jeder Pflanze für die statistische Auswertung verwendet wird. Hierbei wird eine Skalierung zwischen den Karotten und den Beikrautpixel verwendet. Außerdem soll die lokale Genauigkeitsanalyse auszugsweise das Verhalten der Störgröße auf die Genauigkeit innerhalb der einzelnen Pflanzen zeigen. Hier wird für jede Pflanze separat berechnet, wie viele Pixel von ihr richtig klassifiziert wurden. Das Ergebnis wird mittels Histogrammdarstellung über alle Pflanzen aufgezeigt.

⁶⁵ Kapitel 3.6.9

4.6.5.1 Untersuchung der Klassifikationsmethoden auf Translationsunschärfe

Zur Simulation der Translationsunschärfe wurden die Pixel über die Zeile gemittelt. Die Anzahl der Pixel über die gemittelt wurde, lag bei einem Pixel bis 30 Pixel. Dabei entspricht die Mittelung über einen Pixel dem Originalbild. In der Abbildung 133 sind vier Stufen der Translationsunschärfe dargestellt. Die Unschärfe nimmt von links nach rechts zu. Diese Unschärfe besteht unter realen Bedingungen in etwa in denselben Ausmaß wie hier angeben.

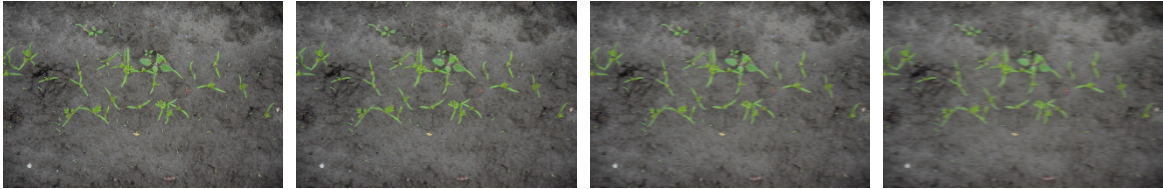


Abbildung 133: Darstellung der verschiedenen simulierten Translationsunscharfen am Beispiel eines Bildes; Von Links: Original, Integration über 10 Pixel, 20 Pixel und 30 Pixel

Die globale Genauigkeitsanalyse, in der alle Pixel unabhängig von den einzelnen Bildern und Pflanzen zusammen für die Auswertung verwendet werden, zeigt das Verhalten der jeweiligen Translationsunschärfe der beiden Methoden. Die Ergebnisse sind in Abbildung 134 dargestellt. In Rot ist die ursprüngliche Methode und in Blau die optimierte Methode hinterlegt. Beide nutzen die Genauigkeitsskala auf der linken Seite. Die Abszisse zeigt die Translationsunschärfe, die durch die Anzahl der Pixel erzeugt wurde. Auf der rechten Seite des Bildes sind mit dem dazugehörigen schwarzen Graph die Anzahl der Klassifikationen hinterlegt.

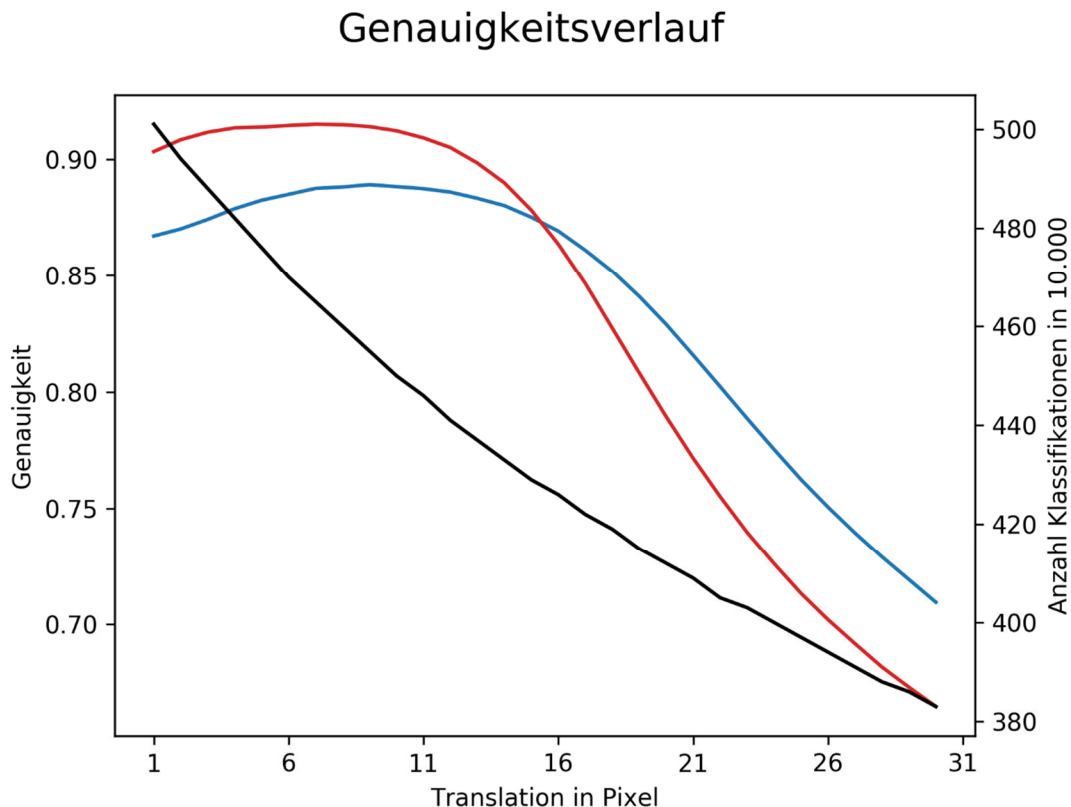


Abbildung 134: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) mit der Auswertung verschiedener Translationsgeschwindigkeiten (linke Skala); Schwarz ist die Anzahl der Klassifikationen (rechte Skala)

Dadurch, dass die Translationsunschärfe einen Einfluss auf die Segmentierung besitzt, nimmt die Anzahl der Klassifikationen ab. Dies könnte von statistischer Relevanz sein, was hier jedoch nicht der Fall ist, da im schlechtesten Fall immer noch über 3,8 Millionen Pixel zur Auswertung verwendet wurden und damit die Stichprobe groß genug ist. Die Translationsunschärfe bei einem Pixel stellt das Originalbild dar. Dass hier ein Unterschied zur Genauigkeitsuntersuchung aus dem vorherigen Kapitel existiert, liegt an der Verwendung des Segmentierers. Es zeigt sich, dass die Genauigkeit mit der Zunahme der Unschärfe zuerst leicht ansteigt, anschließend jedoch ab einer Translationsunschärfe von circa neun Pixel wieder abfällt. Dabei verläuft die Abnahme der Genauigkeit der optimierten Methode nicht so steil wie die der herkömmlichen Methode. Daraus lässt sich folgern, dass die optimierte Methode resistenter gegen Translationsunschärfe ist. Dies könnte unter anderem daran liegen, dass die Features nur auf den Pflanzenpixeln arbeiten, wodurch der Randbereich (Pflanze / Hintergrund) nicht so starken Einflüssen unterliegt. Interessant ist auch der Verlauf der Precision und des Recalls. Diese sind hier für die Kategorie der Karotte in Abbildung 135 dargestellt.

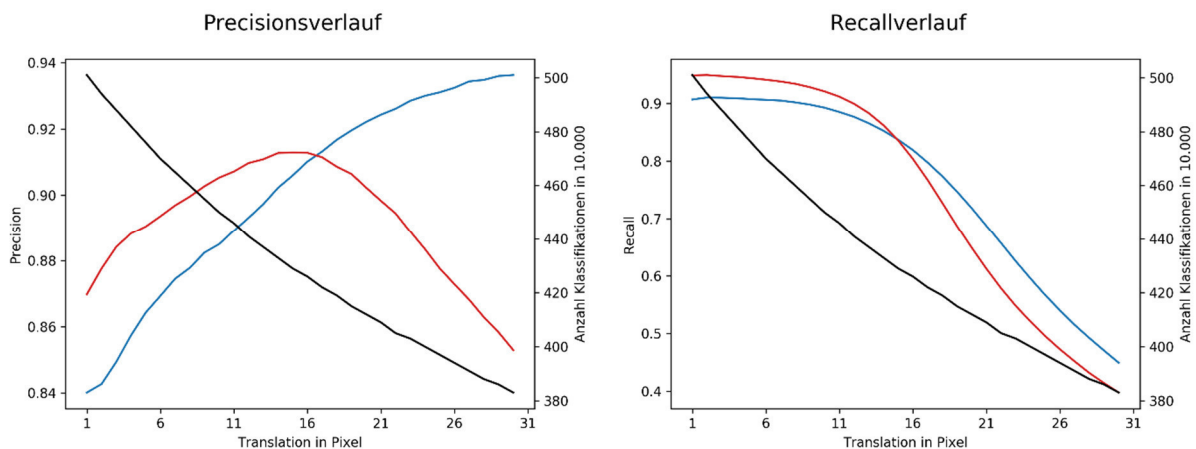


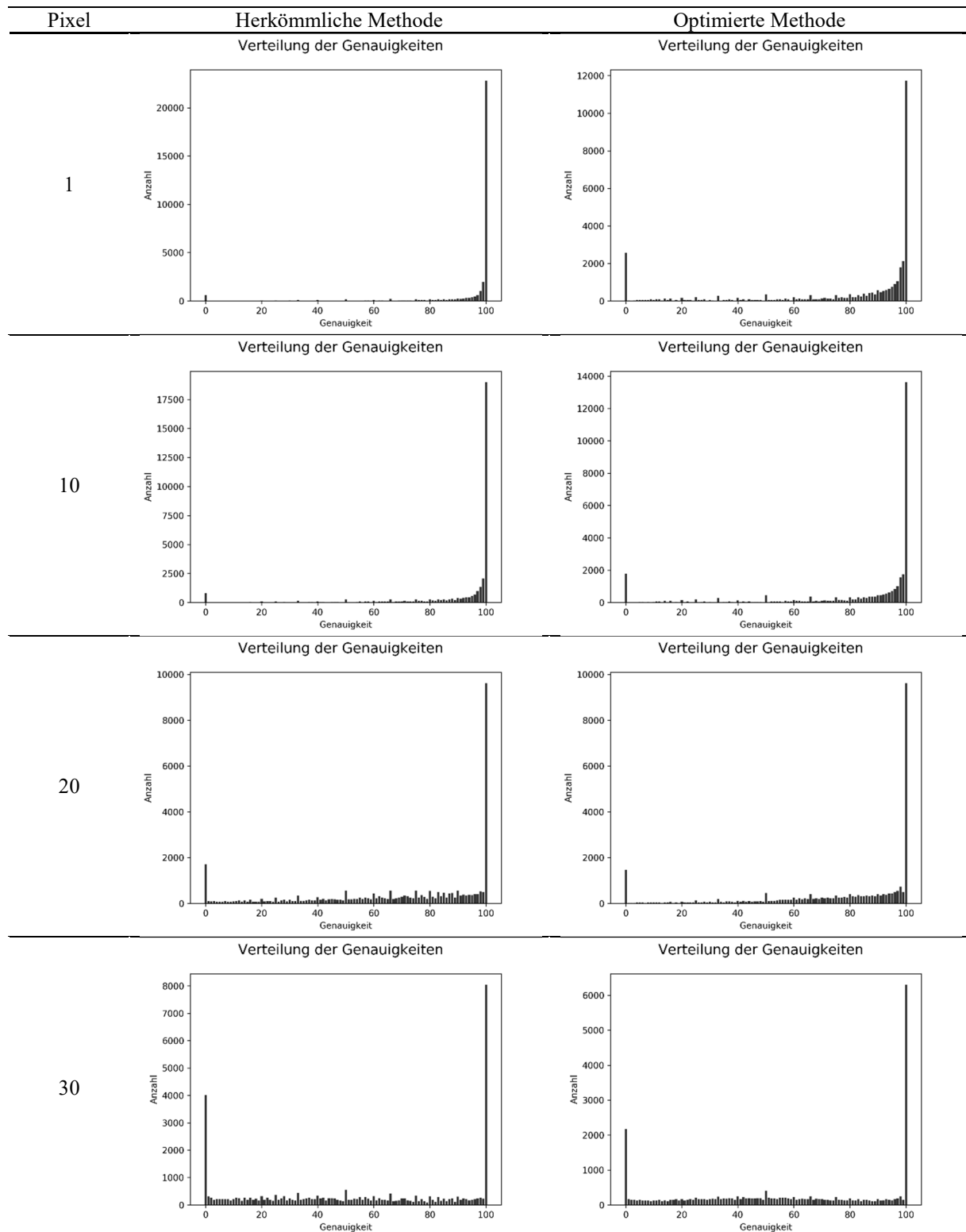
Abbildung 135: Darstellung der Precision (links) und des Recalls (rechts) für die Translationsunschärfe

Auch hier wurde wieder die globale Genauigkeitsanalyse verwendet. Der rote Graph zeigt wieder die herkömmliche Methode und der blaue Graph die optimierte Methode, beide nutzen die jeweilige linke Skala. In Schwarz ist erneut die Anzahl der Klassifikationen dargestellt mit der jeweiligen rechten Skala. Die Precision zeigt, dass diese für die herkömmliche Methode zuerst ansteigt, jedoch ab einer Unschärfe von circa 15 Pixel wieder abfällt. Die Precision der optimierten Methode steigt jedoch weiter an. Die allgemeine Zunahme der Precision der beiden Methoden wird erneut durch den Randbereich (Pflanze / Hintergrund) erzeugt. Der Segmentierer schneidet bei zunehmender Translationsunschärfe den Randbereich von den Pflanzen ab. Durch das Abschneiden verschwindet diese Fehlerquelle und die Precision nimmt dadurch für die Methoden zu. Bei weiter zunehmender Translationsunschärfe bedeutet dies für die herkömmliche Methode, dass zunehmend Beikraut als Karotte klassifiziert wird. Die optimierte Methode zeigt ein gegenteiliges Verhalten. Der Verlauf des Recalls zeigt, dass bei zunehmender Translationsunschärfe Karottenpixel als Beikrautpixel klassifiziert werden.

Abschließend soll noch gezeigt werden, wie sich die lokale Genauigkeit über die Translationsunschärfe verändert. Hier soll durch Histogrammdarstellung gezeigt werden, wie sich die Genauigkeiten in den einzelnen Pflanzen verhalten. Dafür ist in der nachfolgenden Tabelle in der linken Spalte die Anzahl der für die jeweilige Translationsunschärfe verwendeten Pixel angezeigt, in der Mitte ist das jeweilige Histogramm der herkömmlichen Methode und in der rechten Spalte das Histogramm der optimierten Methode dargestellt. Wichtig bei der Betrachtung ist, zu beachten, dass jedes Histogramm eine eigene Skalierung der Achsen verwendet. Bei zunehmender Translationsunschärfe nimmt die Anzahl der 100% richtig klassifizierten Pflanzen für die herkömmliche Methode ab und verlagert sich zu den komplett falsch klassifizierten Pflanzen. Die optimierte Methode zeigt zuerst einen leichten Anstieg der 100% richtig klassifizierten Pflanzen, nimmt jedoch bei zunehmender Translationsunschärfe ab. Bei einer

Translationsunschärfe von circa 20 Pixel besitzen beide Methoden eine gleiche Güte und ab circa 30 Pixel zeigt die optimierte Methode ein besseres Verhalten bei den komplett falsch klassifizierten Pflanzen.

Tabelle 47: Darstellung des Verhaltens der Genauigkeitsverteilung über die Translationsunschärfe



4.6.5.2 Untersuchung der Klassifikationsmethoden auf allgemeine Unschärfe

Mit den 1075 Bildern wurde eine Simulation allgemeiner Unschärfe durchgeführt, wie es zum Beispiel ein falsch eingestelltes Objektiv verursacht. Hierfür wurde ein Gaußfilter mit einer Kernelgröße von 1, 3, 5, 7 bis 25 verwendet. Die Abbildung 136 zeigt vier Kernelgrößen für die Simulation der allgemeinen Unschärfe. Die Unschärfe nimmt von links nach rechts zu.

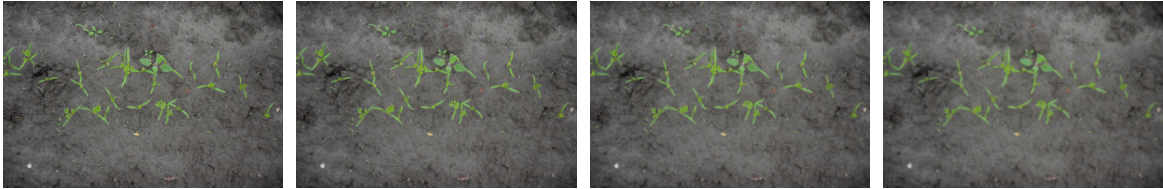


Abbildung 136: Darstellung der verschiedenen simulierten allgemeinen Unschärfen am Beispiel eines Bildes; Von links: Original, Integration über 9, 17 und 25 Pixel für die Kernelgröße

Auch hier soll zuerst die globale Genauigkeitsanalyse verwendet werden, um den Einfluss der allgemeinen Unschärfe auf die beiden Methoden darzustellen. Die nachfolgende Abbildung zeigt das Ergebnis. In rot ist die herkömmliche Methode und in blau die optimierte Methode dargestellt. Beide verwenden wieder die linke Skala. Die Anzahl der Klassifikationen mit der rechten Skala ist wieder in Schwarz dargestellt. Die Abszisse zeigt die Kernelgröße (Kernelsize), die mit dem Gaußfilter verwendet wurde.

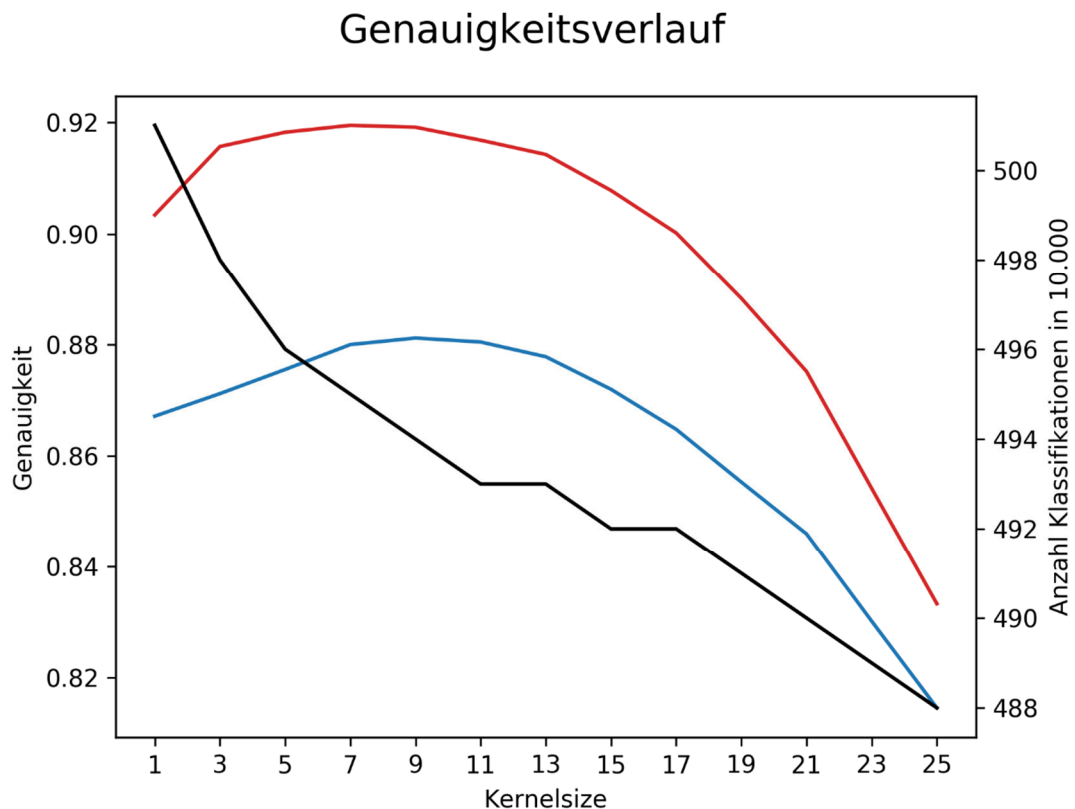


Abbildung 137: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) mit der Auswertung verschiedener allgemeiner Unschärfen (linke Skala); Schwarz ist die Anzahl der Klassifikationen (rechte Skala)

Auch hier zeigt sich, dass die Abnahme der Klassifikationen, die durch den Segmentierer verursacht wird, keinen nennenswerten statistischen Einfluss auf die Untersuchung hat. Im schlechtesten Fall wurden circa 4,8 Millionen Klassifikationen verwendet. Auch zeigen beide Methoden ein ähnliches Verhalten: Eine leichte Unschärfe erhöht die Genauigkeit des Klassifikators, eine zu große Unschärfe lässt jedoch die Genauigkeit abfallen. Es zeigt sich auch, dass es keine nennenswerten Unterschiede beider Methoden in Bezug auf das Verhalten der Unschärfe gibt. Ebenfalls zeigt der Verlauf der Precision und des Recalls (Abbildung 138), dass aus Sicht der Pflanzen beide Methoden nur leichte Unterschiede aufweisen. Auch hier gilt wieder, der rote Graph stellt die herkömmliche und der blaue Graph die optimierte Methode (linke Skala) dar. Die rechte Skala verwendet den schwarzen Graphen und zeigt die jeweilige Anzahl der Klassifikationen. Anders als bei der Translationsunschärfe steigt jedoch die Precision, wohingegen der Recall abfällt.

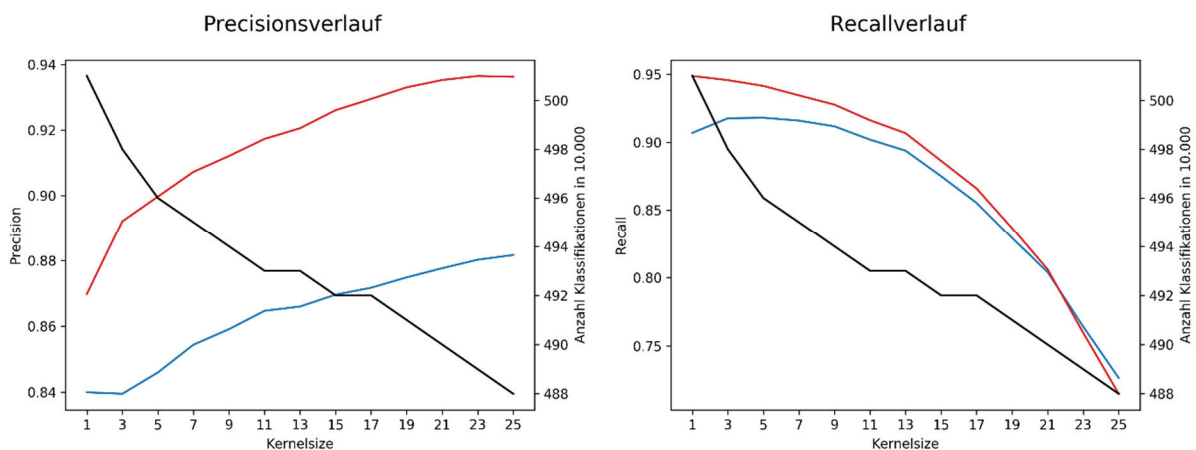
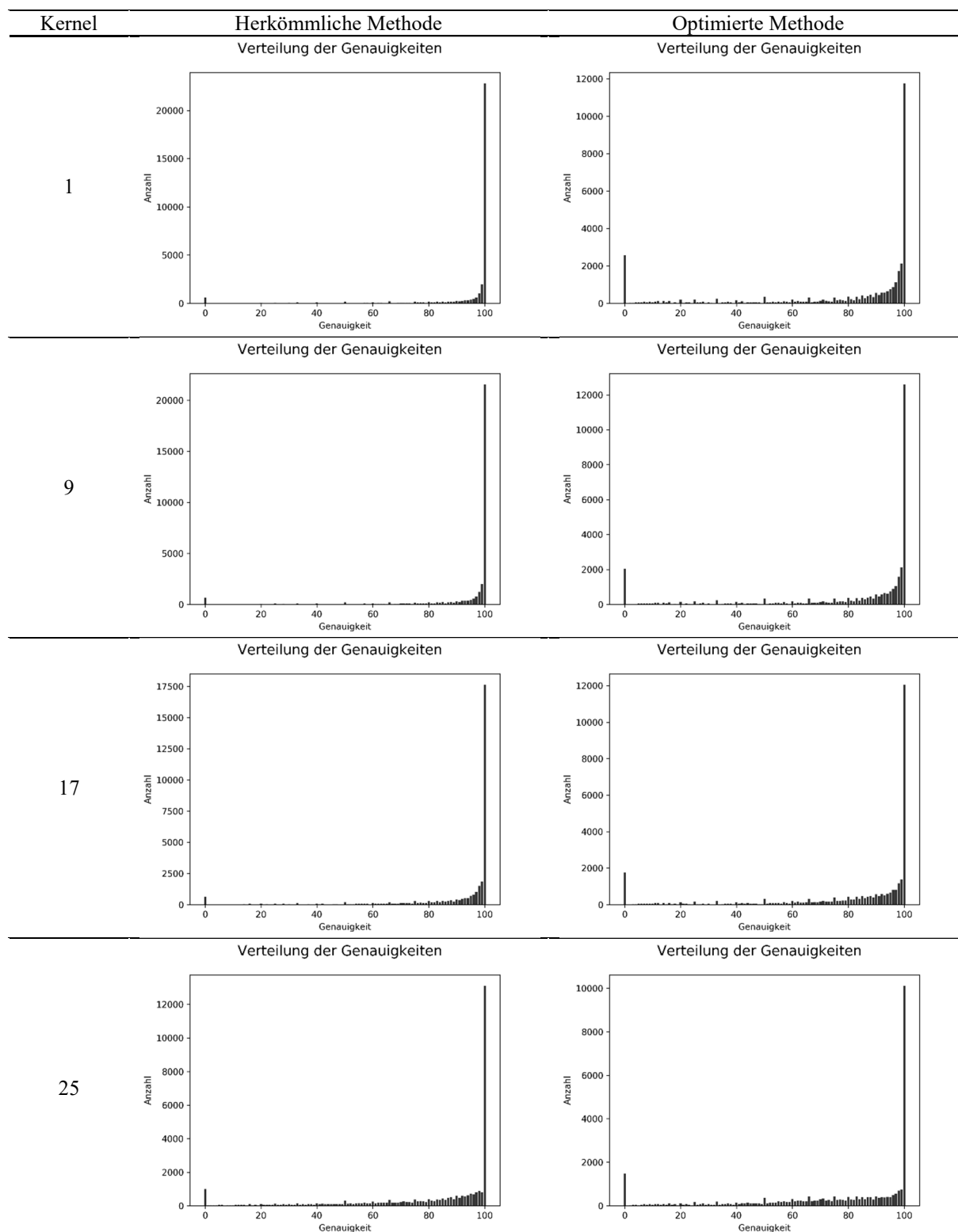


Abbildung 138: Darstellung der Precision (links) und des Recalls (rechts) für die allgemeine Unschärfe

Dies bedeutet, dass durch zunehmende allgemeine Unschärfe weniger Beikrautpixel zur Klasse „Karotte“ klassifiziert werden, jedoch auch Karottenpixel zunehmend zur Klasse „Beikraut“, so dass bei großer Unschärfe der Klassifikator eher zur Klasse „Beikraut“ tendiert.

Die lokale Genauigkeit für die Untersuchung der Genauigkeiten innerhalb einer Pflanze soll hier wie bekannt mit der Histogrammdarstellung erfolgen. Dafür ist in der nachfolgenden Tabelle eine Gegenüberstellung der herkömmlichen und der optimierten Methode für die Kernelgrößen 1, 9, 17 und 25 dargestellt. Auch hier ist es wieder wichtig, dass die einzelnen Histogramme zur detaillierten Darstellung unterschiedliche Skalen verwenden. Es zeigt sich, dass durch die Zunahme der allgemeinen Unschärfe die Genauigkeit innerhalb der Pflanzen abnimmt. Dabei ist das Verhalten beider Methoden ähnlich.

Tabelle 48: Darstellung des Verhaltens der Genauigkeitsverteilung über die Zunahme der allgemeinen Unschärfe



4.6.5.3 Untersuchung der Klassifikationsmethoden auf Unter- und Überbelichtung

Das Verhalten auf Über- und Unterbelichtung der Bilder wurde ebenfalls simuliert. Dabei wurden die Belichtungsstufen von 10% bis 300% in 10%-Schritten gewählt. Die Belichtungsstufe 0% bedeutet, dass kein Licht im Bild ist, also alle Pixel in den Bildern schwarz sind, 100% entspricht dem Originalbild und 200% und 300% entsprechen einer Überbelichtung. Beispielhaft sind vier Fälle in Abbildung 139 dargestellt. Die Belichtung nimmt von links nach rechts zu.



Abbildung 139: Darstellung der verschiedenen simulierten Helligkeitsstufen am Beispiel eines Bildes;
Von links: 30%, Original (100%), 200% und 300% Belichtung

Der Einfluss der Belichtung soll zuerst mit der globalen Genauigkeitsanalyse untersucht werden. Hier werden alle Pixel für die statistische Auswertung unabhängig von den einzelnen Bildern zusammengefasst. Das Ergebnis steht in der unten dargestellten Abbildung. Es gilt wieder: Der rote Graph zeigt die herkömmliche Methode und der blaue Graph die optimierte Methode (linke Skala). Der schwarze Graph zeigt wieder die Anzahl der Klassifikationen der einzelnen Helligkeitsstufen mit der rechten Skala.

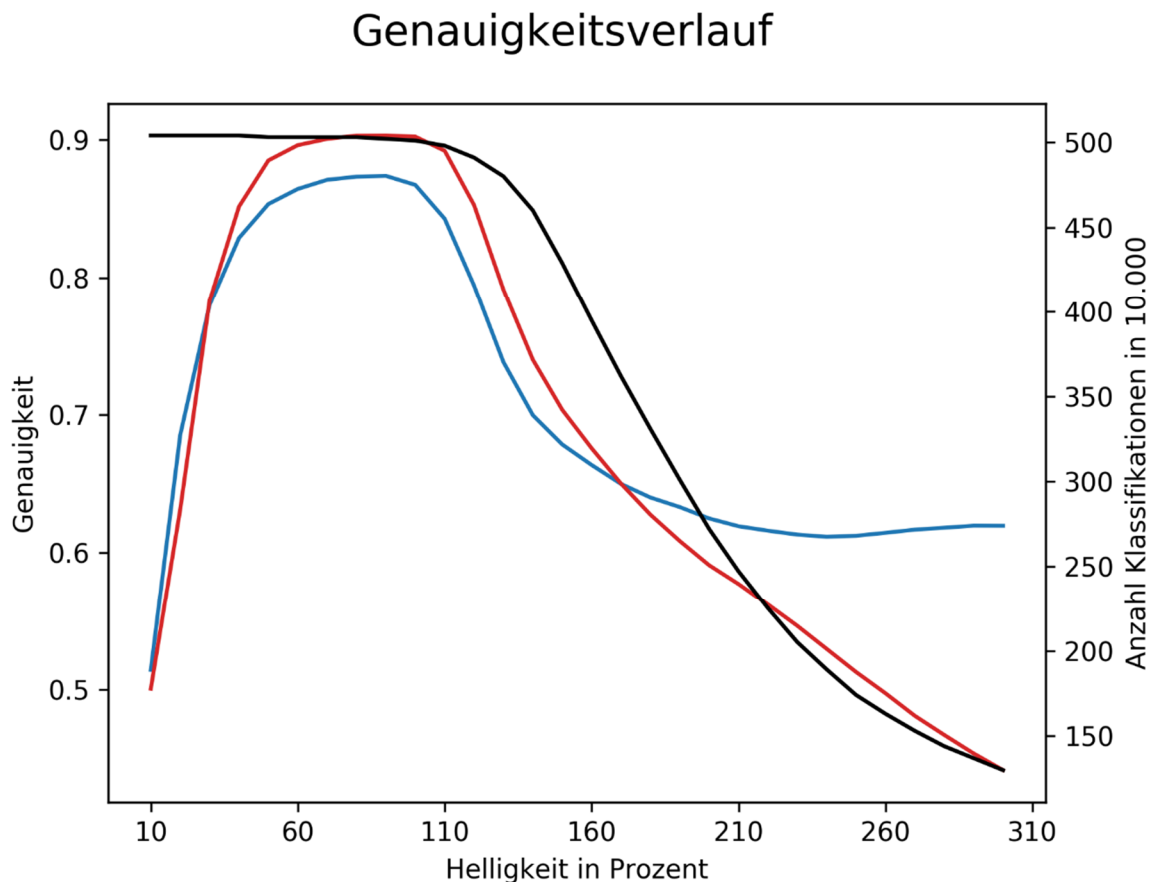


Abbildung 140: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) mit Auswertung der verschiedenen Helligkeitsstufen (linke Skala); Schwarz ist die Anzahl der Klassifikationen (rechte Skala)

Wie zu sehen ist, nehmen die Klassifikationen mit zunehmender Helligkeit ab, da der Segmentierer in den weißen Bereichen keine Informationen für die Unterscheidung zwischen Hintergrund und Pflanze findet. Jedoch liegen auch hier im schlechtesten Fall noch 1,5 Millionen Pixel für die Auswertung vor, so dass ein statistischer Einfluss nicht groß ist. Beide Methoden weisen auch hier im Anfangsbereich ein ähnliches Verhalten auf. Die Genauigkeit steigt erst leicht an. Da das Bild hier zu dunkel ist, kann der Klassifikator nicht die nötigen Informationen extrahieren, um sinnvoll die beiden Klassen einzuteilen. Bei circa 100% Helligkeit, also dem Original Bild, erreichen beide Klassifikatoren ihr Maximum und fallen dann mit zunehmender Überbelichtung wieder ab. Die optimierte Methode stabilisiert sich dabei bei circa 65% der Genauigkeit. Die unten dargestellten Verläufe der Precision und des Recalls für die Kategorie der Karotte zeigen ebenfalls, dass beide Methoden ein fast identisches Verhalten haben. Auch hier gilt, rot zeigt die herkömmliche und blau die optimierte Methode, beide verwenden die linke Skala. Die rechte Skala, die die Anzahl der Klassifikationen zeigt, gehört zu dem schwarzen Graph. Die Werte für die Precision und den Recall zeigen, dass sich der Klassifikator mit zunehmender Belichtung des Bildes eher für die Klasse „Karotte“ entscheidet.

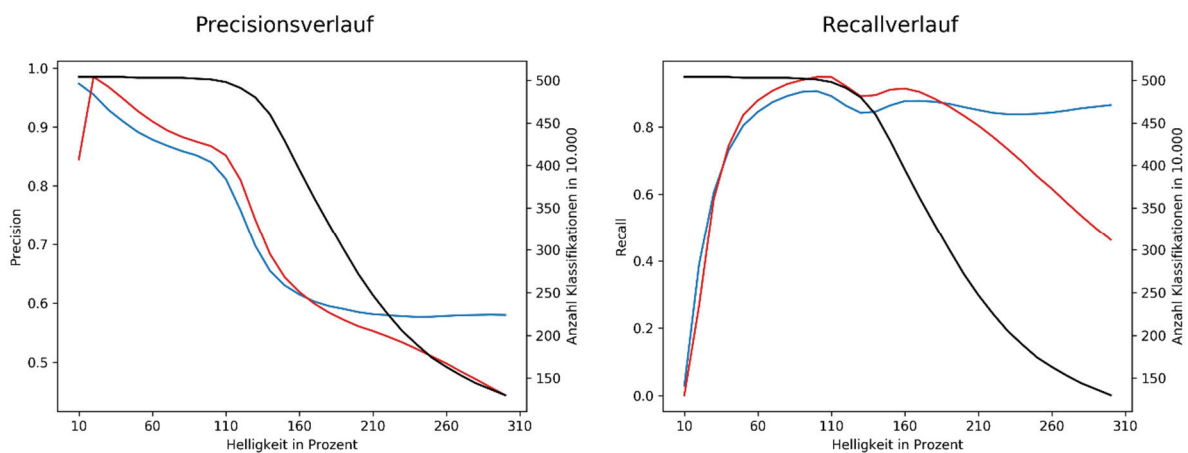
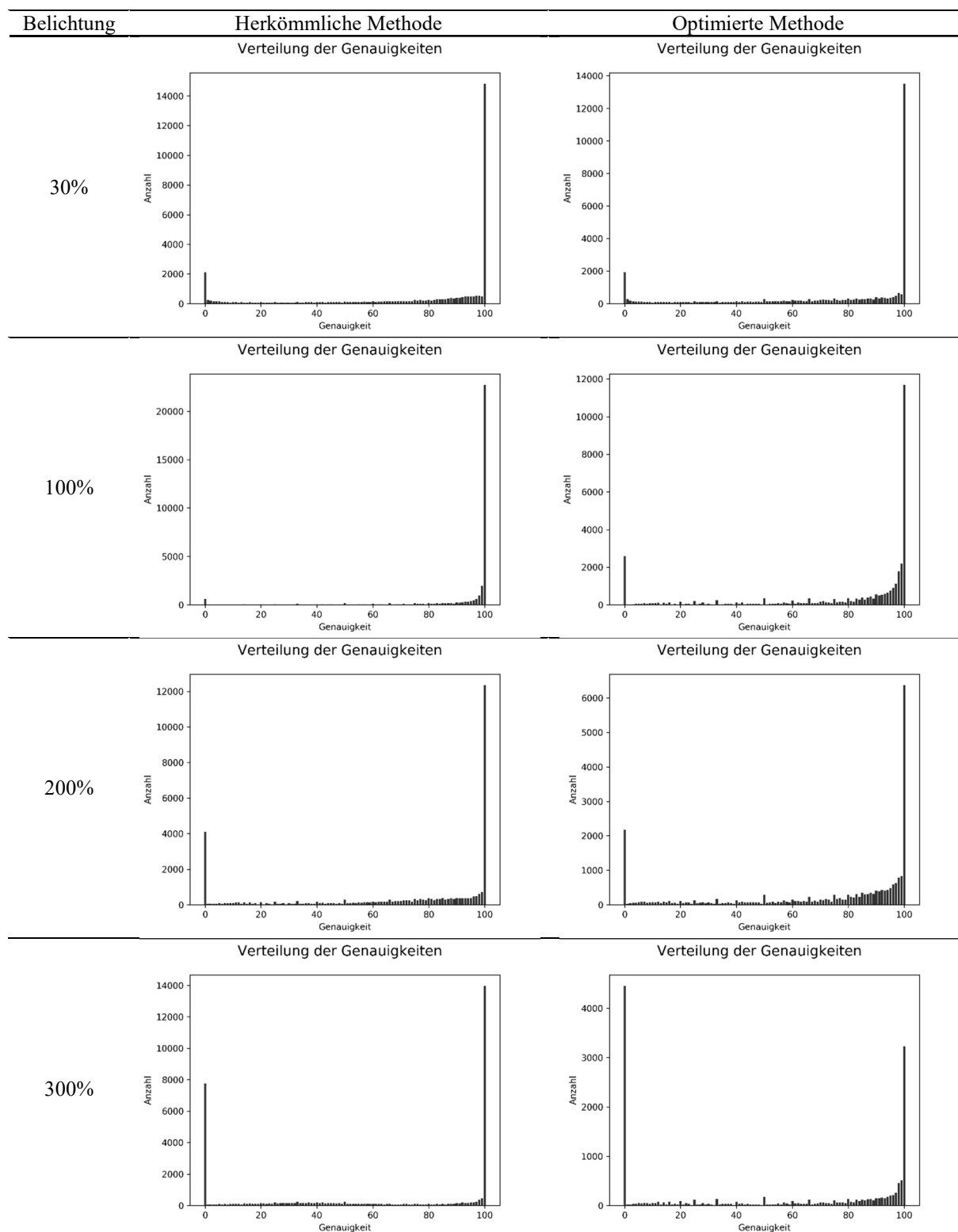


Abbildung 141: Darstellung der Precision (links) und des Recalls (rechts) für die verschiedenen Helligkeitsstufen

In Tabelle 49 sind vier Stufen verschiedener Belichtungen mit der lokalen Genauigkeitsanalyse mittels Histogrammdarstellung aufgezeigt. Die lokale Genauigkeitsanalyse wertet die Genauigkeit innerhalb jeder einzelnen Pflanze aus. Auch hier sollte darauf geachtet werden, dass jedes Histogramm eine andere Skalierung verwendet, damit eine detaillierte Darstellung möglich ist. Die Ergebnisse der Genauigkeiten der einzelnen Pflanzen hängen hier stark mit dem Segmentierer zusammen. Der Segmentierer findet in unterbelichteten und überbelichteten Bildern weniger Pflanzenpixel, da die Information verloren geht. Daher bestehen einzelne Pflanzen in diesen Fällen aus weniger Pixel, was auch das Verhalten der Genauigkeitsverteilung erklärt. In der herkömmlichen Methode steigt die einzelne Genauigkeit der Pflanzen zuerst an und fällt dann nach der Originalbelichtung wieder ab. Mit der zunehmenden Belichtung zwischen 200% und 300% verteilen sich die einzelnen Genauigkeiten der Pflanzen auf die beiden Extrema von komplett falsch und 100% richtig klassifiziert. Dies liegt wie beschrieben an der abnehmenden Zahl von Pflanzenpixel. Wenn eine Pflanze nur noch aus wenigen Pixel besteht, ist es wahrscheinlicher, dass diese komplett richtig oder komplett falsch klassifiziert werden. Die optimierte Methode hat scheinbar ihr bestes Ergebnis in den dunklen Bildern, aber das liegt auch hier an der geringen Anzahl der Pflanzenpixel. Die Genauigkeit nimmt mit zunehmender Belichtung ab, aber auch hier verteilt es sich auf die beiden Extrema.

Tabelle 49: Darstellung des Verhaltens der Genauigkeitsverteilung über die Zunahme der Belichtungsstufen



4.6.5.4 Untersuchung der Klassifikationsmethoden auf Unter- und Übersättigung der Farbe

Die Untersuchung der Über- und Untersättigung der Farbe in den Bildern wurde ähnlich wie die Belichtungsuntersuchung erzeugt. Dabei wurde die Sättigung der Farbe von 0% bis 300% in 10%- Schritten gewählt. Eine Sättigung von 0% bedeutet, dass das Bild ein Graustufenbild ist und keine Farbinformation mehr beinhaltet. Eine Sättigung von 100% entspricht wieder dem Originalbild und 200% und 300% einer Übersättigung im Bild.

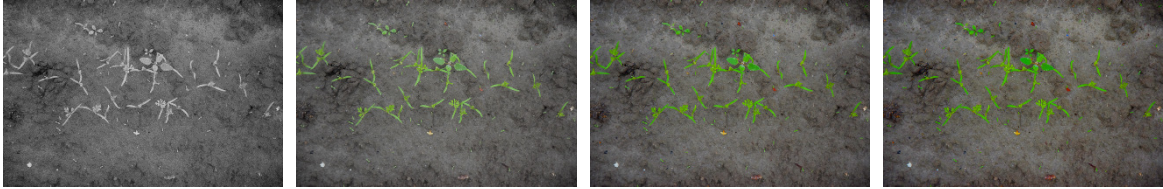


Abbildung 142: Darstellung der verschiedenen simulierten Farbsättigungen am Beispiel eines Bildes; Von links: 0%, Original (100%), 200% und 300% Farbsättigung

Die erste Untersuchung ist wieder die globale Genauigkeitsanalyse, in der jeder Pixel unabhängig von den einzelnen Bildern oder Pflanzen verwendet wird, um die statistischen Größen zu berechnen. Das Ergebnis ist in Abbildung 143 dargestellt. In Rot ist die herkömmliche und in Blau die optimierte Methode dargestellt, die die linke Skala verwendet. Der schwarze Graph verwendet die rechte Skala und zeigt die Anzahl der Klassifikationen.

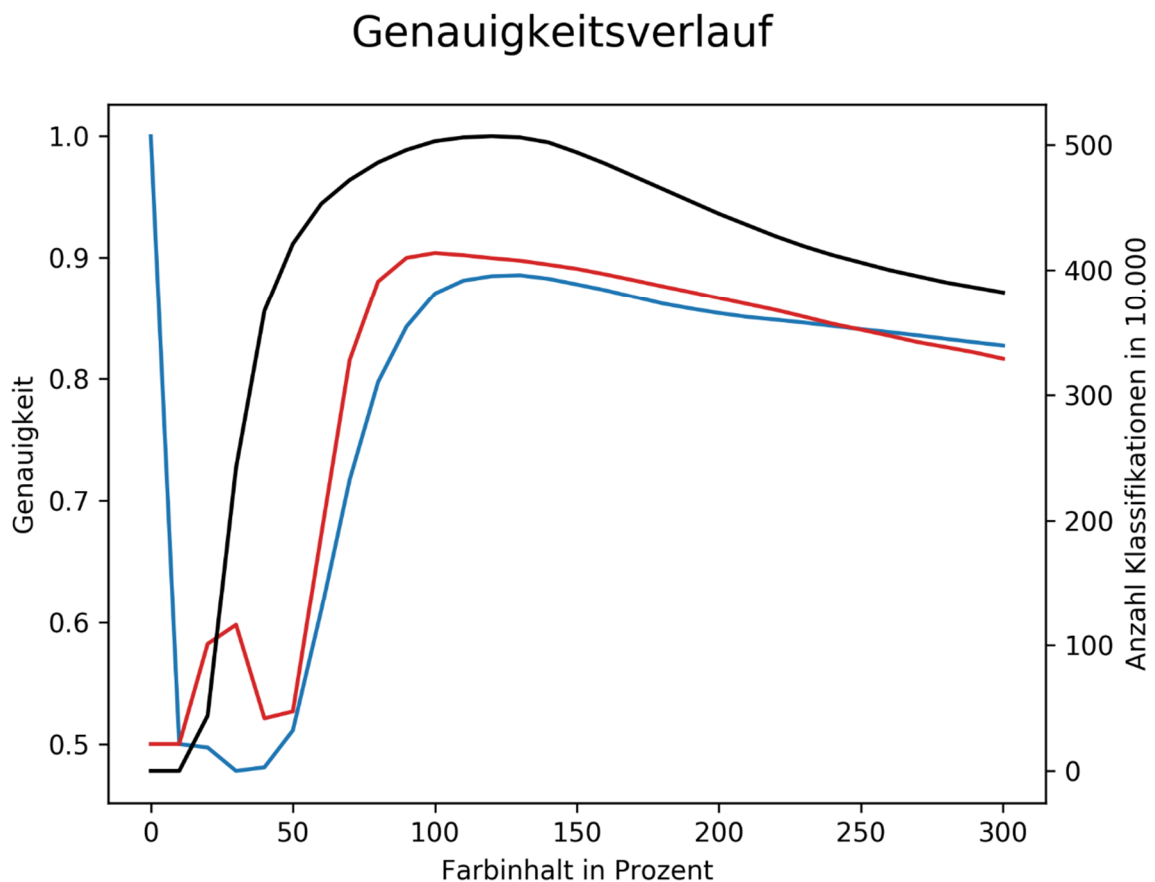


Abbildung 143: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) mit der Auswertung verschiedener Farbsättigungen (linke Skala); Schwarz ist die Anzahl der Klassifikationen (rechte Skala)

Der Segmentierer schafft es nicht, in den Graustufenbildern (Farbsättigung 0%) die Pflanzen zu separieren, daher bricht der Graph hier ein. Damit eine statistische aussagekräftige Anzahl an Klassifikationen vorliegt, soll die nachfolgende Untersuchung erst ab einer Farbsättigung von 50% beginnen. Hier liegen circa 4 Millionen Klassifikationsergebnisse vor. Die Genauigkeit hat für beide Methoden denselben Charakter. Eine Farbuntersättigung bewirkt einen Abfall der Genauigkeit. Auch eine Farbübersättigung bewirkt einen Abfall, allerdings ist der Klassifikator bei der Übersättigung nicht so empfindlich wie bei Untersättigung. Die Graphen für die Precision und den Recall für die Kategorie der Karotte sind in Abbildung 144 dargestellt. Der rote Graph zeigt die herkömmliche und der blaue Graph die optimierte Methode, die die linke Skala verwendet. Schwarz ist wieder die Anzahl der Klassifikationen mit der rechten Skala.

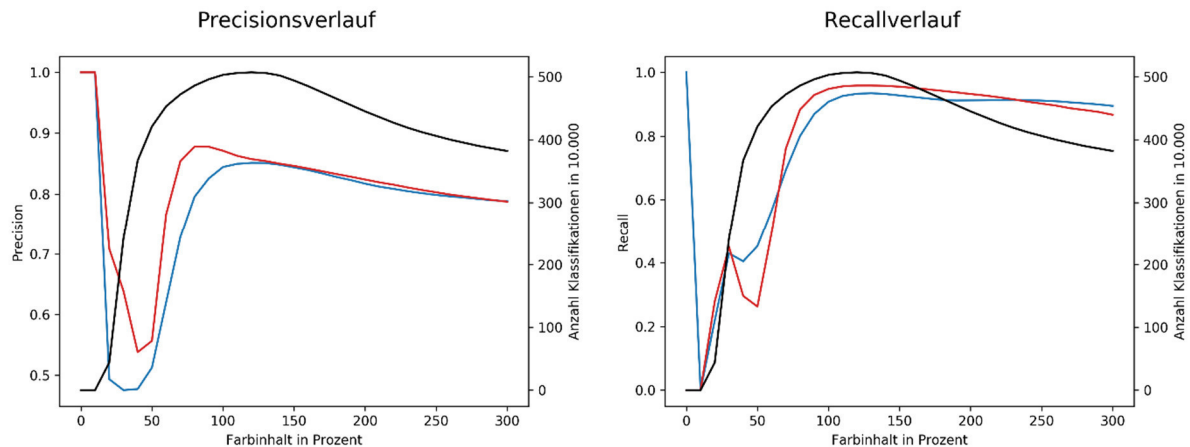
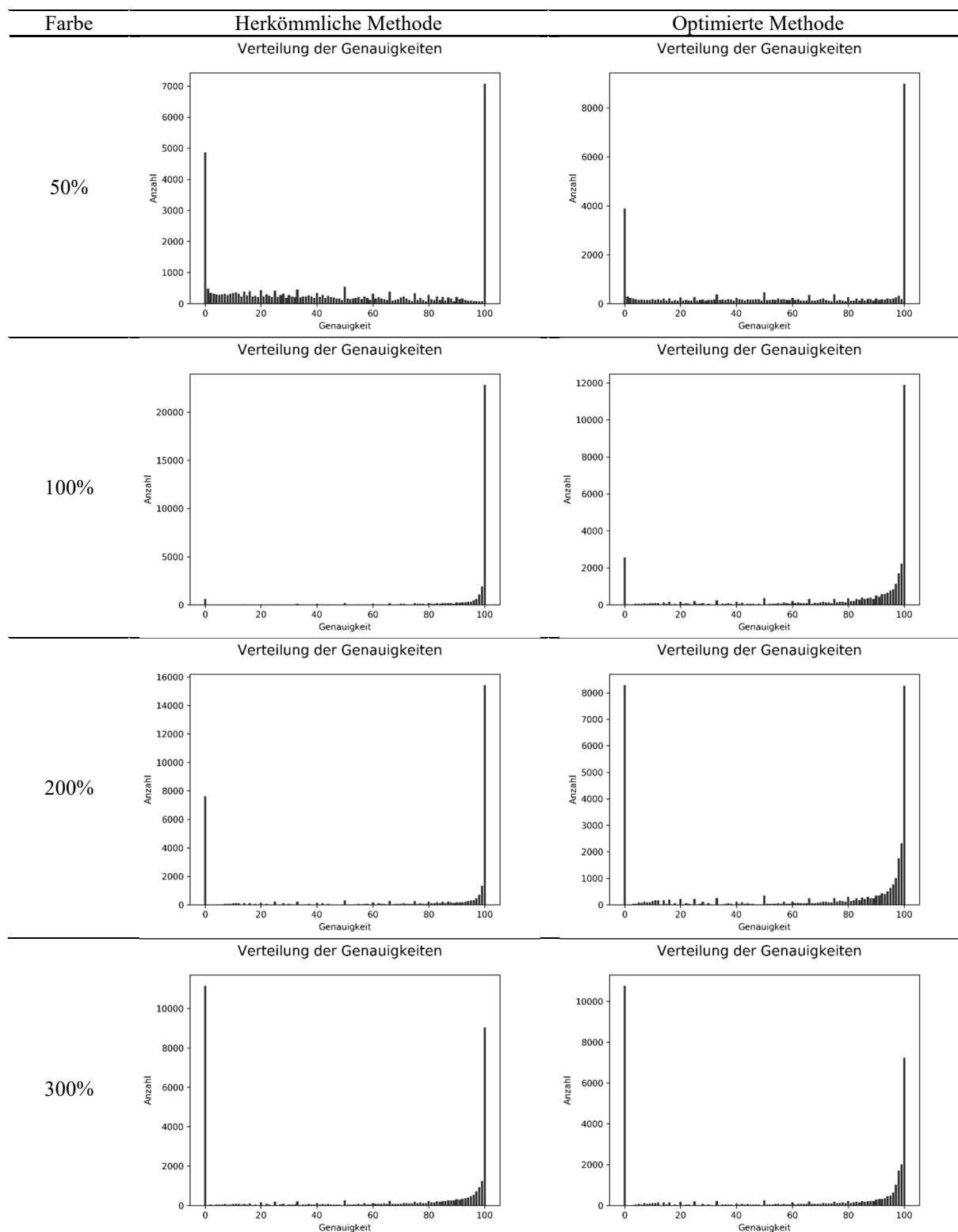


Abbildung 144: Darstellung der Precision (links) und des Recalls (rechts) für die verschiedenen Farbsättigungen

Die beiden Methoden verhalten sich bei der Precision und dem Recall gleich. Eine Farbuntersättigung zeigt ein Einbrechen sowohl bei der Precision, als auch beim Recall. Bei einer Farbsättigung von Beispielsweise 50% ist das Klassifikationsergebnis eher zufällig. Dies zeigt auch die Genauigkeit an diesem Punkt. Die Werte steigen bis zur Original-Farbsättigung an und fallen dann mit zunehmender Farbsättigung leicht ab.

Abschließend soll die lokale Genauigkeitsanalyse das Verhalten der Genauigkeiten innerhalb der einzelnen Pflanzen zeigen. Dafür sind in Tabelle 50 vier Fälle der Farbsättigung der herkömmlichen Methode und der optimierten Methode jeweils gegenübergestellt. Auch hier zeigt sich wieder das extreme Verhalten zwischen komplett falsch und komplett richtig klassifizierten Pflanzen. Bei einer Farbuntersättigung ist der Anteil der komplett falsch klassifizierten Pflanzen in beiden Methoden noch relativ groß. Die Verteilung verschiebt sich dann zu „komplett richtig klassifiziert“, in der Nähe der Original-Farbsättigung und wandert dann wieder zurück zu „komplett falsch klassifiziert“ bei zunehmender Farbsättigung. Am Ende überwiegt der Anteil der komplett falsch klassifizierten Pflanzen den Anteil der komplett richtig klassifizierten Pflanzen.

Tabelle 50: Darstellung des Verhaltens der Genauigkeitsverteilung über die Zunahme der Farbsättigung



4.6.6 Fazit zur Optimierung

In dem Kapitel, das die Optimierung des Klassifikators zeigt, wurde ein Verfahren zur effizienteren Verarbeitung des CNN vorgestellt. Anschließend wurde die optimierte Verarbeitungsmethode direkt im Aufbau einer eigenen CPU verwendet. Die CPU wurde mittels eines FPGA umgesetzt und verwendet gezielte Hardware-Algorithmen um das CNN effizient zu verarbeiten. Als Alternative wurde die optimierte Methode mit einer GPU umgesetzt und anschließend tiefer auf die Genauigkeit und Geschwindigkeit untersucht. Als Vergleichsmethode wurde die herkömmliche Abarbeitung herangezogen. Damit eine Verringerung der Genauigkeit, die durch die Optimierung zu Stande kommt, kompensiert werden kann, wurde ein eigens für die optimierte Methode entworfenes Trainingsverfahren vorgestellt, mit dem die Gewichte direkt mit den Optimierungen erstellt werden können. Für eine weitere Verbesserung der Geschwindigkeit wurde die Netzwerkstruktur anhand der Optimierungen verkleinert und angepasst. Damit können Bilder mit vielen Pflanzenpixel mit 10 FPS verarbeitet werden und 90% der Bilder können sogar mit einer FPS zwischen 18 und 23 klassifiziert werden. Mit der herkömmlichen Methode lag die FPS bei circa einem Bild in drei Sekunden. Abschließend wurde die verbesserte optimierte Struktur tiefer auf Einflüsse der Unschärfe und Bewegungsunschärfe, der Farbsättigung sowie der Belichtungsstärke der einzelnen Bilder untersucht. In zukünftigen Arbeiten sollte untersucht werden, ob eine leichte Vergrößerung der Netzwerkstruktur einen Genauigkeitsvorteil bringt, da hier weitere parallel arbeitende Strukturen fehlerhaft entstehende Informationen innerhalb des Netzes besser kompensieren können. Dafür sei zum Beispiel eine tiefere Convolution-Schicht vorgeschlagen, die nach den Untersuchungen keine großen zeitlichen Kosten bei der Abarbeitung verursacht. Auch sollte in einer zukünftigen Arbeit ein leistungsstärkerer FPGA und die entworfene CPU Struktur weiter untersucht werden. Außerdem sollte das Netzwerk in einer zukünftigen Untersuchung mit unscharfen Bildern, über- und unterbelichteten Bildern, sowie mit über- und untersättigten Farben trainiert werden, damit diese Störgrößen in der Klassifikation minimiert werden können.

4.7 Fazit zum Kapitel der Klassifikatoren

Im ersten Teil des Kapitels wurden für die Klassifikation der Pflanzen fünf verschiedene selbstlernende Klassifikatoren untersucht. Dafür wurden zuerst die in diesem Kapitel verwendeten statistischen Methoden und die Erstellung des Ground-Truth allgemein vorgestellt. Die anschließend untersuchten und vorgestellten Klassifikatoren basieren auf den Deep-Learning Ansätzen, die den Stand der Technik zum Zeitpunkt dieser Dissertation widerspiegeln. Untersuchungen des U-Net und des Yolo-Netzwerkes sollen in nachfolgende Arbeiten aufgenommen werden.

Es wurde sich für die Technologie des CNN entschieden, da die Pflanzenerkennung keine standardisierten Klassen verwenden kann. Jede zu klassifizierende Pflanze unterscheidet sich von jeder anderen Pflanze derselben Klasse. Dazu kommen verschiedene Variationen, unter anderem über das Wachstumsstadium und durch die verschiedenen Wetterbedingungen. Auch können mit der Zeit weitere noch unbekannte Beikräuter dazukommen, so dass sich der Klassifikator immer wieder neuen Bedingungen ausgesetzt sieht. Die fünf verschiedenen Ansätze wurden auf die Genauigkeit auf Pixelebene und der Pflanzenebene separat betrachtet und gegenübergestellt. Anschließend wurde anhand der Untersuchungsergebnisse eine Technologie ausgewählt und tiefer analysiert. Dabei wurde die Genauigkeit des Klassifikators genauer analysiert und herausgefunden, dass bei einigen bestimmten Fällen die Klassifikation fehlerhaft ist. Diese Fälle konnten genau beschrieben werden und später im Kapitel mit konkreten Ansätzen behoben werden. Der Klassifikator weist zwar eine Genauigkeit im oberen 90%-Bereich auf, jedoch ist seine Verarbeitungsgeschwindigkeit zu langsam für eine reale Anwendung. Daher wurde im zweiten Teil des Kapitels die Technologie für eine schnelle Abarbeitung der Klassifikation erstellt. Zur Erstellung der neuen Berechnungsmethode wurde die allgemeine Verarbeitung der Convolutional-Neural-Networks analysiert und eine neue Herangehensweise entwickelt. Außerdem wurde eine eigene Architektur einer CPU für die gezielte Verarbeitung der Convolutional-Neural-Networks entwickelt und umgesetzt. Die Architektur macht sich dabei die neue

Verarbeitungsmethode zu Nutze. Als Alternative wurde die neue Verarbeitungsmethode ebenfalls auf Basis einer GPU umgesetzt. Die neue Verarbeitungsmethode und die herkömmliche wurden abschließend auf Genauigkeit und Zeitverhalten verglichen. Während die herkömmliche Methode circa drei Sekunden pro Bild braucht, schafft die neue Methode eine FPS von bis zu 20 Bildern pro Sekunde. Dies ist ein Speed-Up Faktor von 60. Die Einbußen an Genauigkeit sind dabei nur wenige Prozent. Am Ende des zweiten Teils wurden die herkömmliche und die neue Methode auf Störgrößen, wie Unschärfe in den Bildern, die durch fehlerhaft eingestellte Objektive oder durch Translationsbewegung entstehen können, Über- und Unterbelichtung der Bilder, sowie Über- und Untersättigung der Farben in den Bildern untersucht. Dabei wurde auch das Zusammenspiel des Segmentierers und des Klassifikators betrachtet. In einer folgenden Arbeit sollte das Klassifikationsergebnis weiter verarbeitet werden. Der im Kapitel vorgestellte und entwickelte Area-Klassifikator, der eine Fläche klassifiziert, sollte in einer zukünftigen Arbeit weiter untersucht und weiter entwickelt werden. Dieser Klassifikator könnte ebenfalls mit der neuen Methode der Verarbeitung berechnet werden und einen Geschwindigkeitsvorteil daraus ziehen, der noch größer ist, als der hier tiefer untersuchte Pixelweise-Klassifikator. Auch sollte eine andere Arbeit in Zukunft die einzelnen Pixel verwenden, um eine noch genauere Aussage über die Klassen zu treffen. Dafür könnten zum Beispiel die Klassifikationsergebnisse einer Fläche von 5x5 Pixel genauer betrachtet werden und mit statistischen Methoden eine Aussage über den mittleren Pixel getroffen werden. Mit diesem Feature könnten fehlerhaft klassifizierte Pixel korrigiert werden.

Durch die umfangreiche Bilddatenerhebung auf dem Feld und anschließender Label-Erstellung standen viele Trainings-, Test- und Validierungs-Daten zur Verfügung. Dennoch können zukünftig durch Data-Augmentation noch weitere Daten ohne Probleme für das Training erzeugt werden. So kann zum Beispiel die Methode für die Simulation der Translationsunschärfe oder der Helligkeit verwendet werden, um auf den Trainingsdaten zusätzliche Variationen zu erzeugen. Dadurch würde der Klassifikator auch diese Problemfälle kennenlernen.

Das Zusammenspiel zwischen Segmentierer und Klassifikator konnte bislang nicht wissenschaftlich untersucht werden und sollte in Zukunft nochmals aufgegriffen werden. Erste Versuche auf dem Feld zeigen jedoch, dass der Klassifikator recht robust auf den Schwellwert des Segmentierers reagiert. Bei einem Klassifikator mit den Ausgängen Karotte und Beikraut zeigt sich jedoch, dass viele Fragmente des Hintergrundes als Beikraut klassifiziert werden, was wiederum die Vernichtungseinheit belastet. Daher ist es sinnvoll, den Klassifikator auch die Klasse Hintergrund lernen zu lassen.

5. Zusammenfassung und Ausblick

Die in dieser Dissertation vorgestellten Verfahren ermöglichen die Klassifikation von Pflanzen unter realen Bedingungen. Die besondere Schwierigkeit des Projektes bestand darin, dass dieses Projekt komplett neu war. Es existierten am Anfang weder Messmethoden oder eine Datenbank, in der experimentiert werden konnte. Sämtliche verwendeten Daten mussten selbst im Laufe der Arbeit gesammelt und verarbeitet werden. Dadurch, dass die Daten unter realen Bedingungen gesammelt werden mussten, standen hier nur wenige Wochen im Jahr zur Verfügung. Dieses Zeitfenster erlaubte keinen Spielraum für Fehler, da im gesamten restlichen Jahresverlauf nur mit diesen Daten experimentiert werden konnte.

Die Arbeit teilt sich dabei in zwei Hauptkapitel ein. Im ersten Hauptkapitel wurde die komplette Vorverarbeitung beschrieben und untersucht. Es wurden hier verschiedene Sensor-Technologien die für die Pflanzenklassifikation verwendet werden könnten untersucht und gegenübergestellt. Es wurde anschließend eine Technologie ausgewählt, mit der weiter gearbeitet werden sollte. Auf Basis dieses ausgewählten Sensors wurden sechs Algorithmen für die Segmentierung der Pflanzen vom Hintergrund vorgestellt. Dabei wurden alle Algorithmen auf dieselben Daten angewendet und miteinander verglichen. Der in diesem Kapitel selbst entwickelte HSV-Segmentierungsalgorithmus zeigte ein genaueres und robusteres Verhalten, als das Vegetationsindexverfahren, das bis dato Stand der Technik war. Der Algorithmus ist Teil der vom Autor dieser Dissertation eingereichten nationalen Patente [Patent #1, 2020], [Patent #2, 2020], [Patent #3, 2020] und internationalen Patente [Patent #4, 2020], [Patent #5, 2020], die sich momentan in der Offenlegungsphase befinden. Auch kann es mit einfacheren Sensoren arbeiten, da es keinen zusätzlichen Infrarot-Kanal benötigt. Am Ende des ersten Hauptkapitels wurden die verschiedenen Algorithmen gegenübergestellt und miteinander verglichen. Der HSV-Segmentierungsalgorithmus zeigte die besten Ergebnisse und wurde daher für die weitere Untersuchung ausgewählt. Daher wurde dieser Algorithmus auf verschiedene Störgrößen untersucht, die unter realen Bedingungen auftreten können. Die untersuchten Störgrößen sind Unschärfe, die durch ein fehlerhaft eingestelltes Objektiv oder durch Translationsbewegung hervorgerufen wird, Über- und Unterbelichtung der Bilder sowie Über- und Untersättigung der Farbe im Bild.

Der zweite Hauptteil widmet sich der Entwicklung eines Klassifikators zur Unterscheidung zwischen Nutzpflanzen, hier Karotten und Beikraut. Dafür wurden in dem Kapitel fünf verschiedene Klassifikatoren, die auf der Deep-Learning Technologie basieren, untersucht und verglichen. Alle Klassifikatoren spiegeln dabei den zum Zeitpunkt dieser Arbeit aktuellen Stand der Technik wieder. Es wurden diese lernfähigen Algorithmen ausgewählt, da die Klassifikation der Pflanzen ständiger Veränderung unterliegt. Darüber hinaus führen externe Faktoren wie das Wetter oder das Wachstumsstadium dazu, dass sich die Merkmale für eine Klasse ändern. Auch können über die Jahre neue Pflanzenarten dazu kommen, die vorher nicht auftraten. Die fünf Klassifikatoren wurden unter gleichen Bedingungen getestet und es wurde anhand der Ergebnisse ein Klassifikator für die weitere Arbeit ausgewählt. Dieser Klassifikator wurde tiefer auf Genauigkeit hin untersucht und es wurde herausgefunden, dass in bestimmten Fällen falsch klassifiziert wird. Diese Fälle konnten genau beschrieben werden und im späteren Verlauf des Kapitels konkret korrigiert werden. Der Klassifikator zeigte zwar Genauigkeitsergebnisse im oberen 90%-Bereich, besaß jedoch noch nicht die Geschwindigkeit für die Verarbeitung unter realen Bedingungen. Daher wurden im zweiten Teil des Kapitels neue Methoden entwickelt um Convolutional-Neural-Networks schneller zu verarbeiten. Hier wurden verschiedene Herangehensweisen verwendet. Zuerst wurde analysiert, wie CNNs ihre Daten verarbeiten und eine neue allgemeine Methode definiert, wie dies effizient funktioniert. Anschließend wurde eine eigene Architektur einer CPU entwickelt, die explizit diese neue Methode der Verarbeitung ausnutzt, um die neuronalen Netze zu verarbeiten. Die CPU wurde umgesetzt und im Kapitel untersucht. Als Alternative wurde die Methode auch als GPU Lösung vorgestellt und untersucht. Als letzte Anpassung wurde der Aufbau des CNN auf diese Methode angepasst, um effizienter die neue Methode der Verarbeitung auszunutzen. Es zeigte sich bei einer gründlichen Vergleichsuntersuchung zwischen

der herkömmlichen und optimierten Methode, dass die optimierte Methode nur wenige Prozent in der Genauigkeit verliert, jedoch eine FPS von 20 Bilder pro Sekunde möglich sind, wohingegen die herkömmliche Methode lediglich ein Bild in drei Sekunden schafft. Dies ist ein Speed-Up Faktor von 60. Am Ende des Kapitels wurden beide Methoden auf ihr Verhalten bei den in der Segmentierung vorgestellten Störgrößen verglichen. Außerdem wurde hier das Zusammenspiel des Segmentierers und des Klassifikators aufgezeigt. Auch der Klassifikator mit den Optimierungsmethoden für die GPU und den FPGA sind Teil der vom Autor dieser Dissertation eingereichten nationalen Patente [Patent #1, 2020], [Patent #2, 2020], [Patent #3, 2020] und internationalen Patente [Patent #4, 2020], [Patent #5, 2020], die sich momentan in der Offenlegungsphase befinden.

Der letzte und abschließende Schritt, das Auswerten der Klassifikationsergebnisse und die Bestimmung der Vernichtungs-Koordinaten, ist derzeit nicht wissenschaftlich analysiert und daher in der Arbeit nicht beschrieben. Das Verfahren kann erst in der kommenden Saison entwickelt werden, da eine Abstimmung mit der Vernichtungseinheit nötig ist. Erste Versuche eines Algorithmus, der eine Analyse der einzelnen Klassenwahrscheinlichkeiten in Verbindung mit einem Contourfindungs-Algorithmus verwendet, sind jedoch sehr vielversprechend.

Die hier vorgestellten und entwickelten Technologien haben es geschafft, dass der Klassifikator die Pflanzen mit einer hohen Genauigkeit auch echtzeitfähig klassifizieren kann. Mittlerweile konnte die Technologie nicht nur erfolgreich beim Klassifizieren von Karotten, sondern auch von Spinat und Rote Beete eingesetzt werden. Das neu aufkommende YOLO (You Only Look Once) Netzwerk [Redmon, et al., 2016] zeigt ebenfalls sehr gute Resultate für die Unterscheidung von Pflanzen. Das YOLO Netzwerk besitzt dabei auch einen sehr effizienten Datendurchsatz. Leider stand dieses Netzwerk erst am Ende der Arbeit zur Verfügung, so dass dies nicht mehr gründlich untersucht werden und in die Arbeit einfließen konnte. Dies soll aber in einer nachfolgenden Arbeit näher betrachtet werden. Die ersten Ergebnisse wurden auch bereits veröffentlicht [Czymmek, et al., 2019]. Ebenfalls soll das U-Net mit dem Problem der Pflanzenerkennung in einer zukünftigen Arbeit genauer untersucht werden.

Die Auswahl der Ground-Truth Bilder geschah sowohl beim Test des Segmentierers und beim Test des Klassifizierers durch eine zufällige Auswahl per Hand. Bei der Auswahl wurde jedoch auch darauf geachtet, in den Bildern möglichst viele Variationen und alle möglichen bekannten Fälle des Datensatzes abzubilden. Schöner wäre es, wenn es ein Maß für die Ähnlichkeit der Bilder oder Bildfragmente geben würde. So kann mit dieser Maßzahl eine möglichst große Varianz erzeugt werden. Es wurde zwar versucht, eine solche Zahl mit Hilfe von bildverarbeitungstechnischen Größen wie der Kompaktheit, Fläche, Kontrast usw. zu erzeugen, jedoch half dies bei der Auswahl nicht weiter. Auch der Versuch, die Features des CNNs für die Auswahl zu verwenden, brachte keinen ausreichenden Erfolg. Daher soll hier vorgeschlagen werden, dass in Zukunft untersucht werden soll, wie zwei Bilder sich unterscheiden, um einen Testdatensatz zu erstellen, der möglichst komplex in der Vielfalt ist. Weiter kann mit dieser Möglichkeit auch untersucht werden, wie viele Lernbilder die Netzwerke wirklich benötigen, um ihr Anwendungsgebiet erfolgreich approximieren zu können. Die Frage, wie aussagekräftig ein solcher Datensatz ist, der eine größtmögliche Varianz der Vielfalt aufweist, muss dann auch noch beantwortet werden. Es könnte durch die Auswahl passieren, dass Spezialfälle in der Auswertung, die in diesem Beispiel eventuell einmal pro Saison und einmal pro Feld die gleiche Gewichtung bekommen, wie zum Beispiel 99% der Standardfälle. Ob diese Auswahl dann bei der Untersuchung eines Problems wirklich hilft, muss beantwortet werden.

III. Abbildungsverzeichnis

Abbildung 1: Darstellung eines Karottenfeldes kurz vor dem Auflaufen der Pflanzen.....	10
Abbildung 2: Arbeitsweise von Gewichten und Neuronen in einem künstlichen neuronalen Netz inspiriert aus: [Zell, 1994].....	12
Abbildung 3: Aktivierungsfunktionen eines Neurons inspiriert aus: [Zell, 1994].....	13
Abbildung 4: Beispiele der verwendeten Netztopografie inspiriert aus: [Zell, 1994]	14
Abbildung 5: Schematische Darstellung eines Schwellwertelementes inspiriert aus: [Zell, 1994]	14
Abbildung 6: Anwendungsbeispiel eines Schwellwertelementes auf die Separierung zweier Mengen inspiriert aus: [Kruse, et al., 2011]	15
Abbildung 7: Darstellung des berechneten Schwellwertelementes zum Separieren zweier Mengen inspiriert aus: [Kruse, et al., 2011]	16
Abbildung 8: Nicht linear separierbare Mengen inspiriert aus: [Kruse, et al., 2011]	17
Abbildung 9: Schwellwertelement für die Separierung des Kreises unten rechts inspiriert aus: [Kruse, et al., 2011]	17
Abbildung 10: Umsetzung der Booleschen Operation oder als Schwellwertelement (links). Darstellung der zu separierenden Mengen (rechts), Darstellung des Schwellwertelementes inspiriert aus: [Kruse, et al., 2011].....	18
Abbildung 11: Zusammengesetztes Schwellwertelement zur Separierung nicht linear separabler Mengen inspiriert aus: [Kruse, et al., 2011]	18
Abbildung 12: Schwellwertelement zur Trennung komplexerer Mengen inspiriert aus: [Zell, 1994]	18
Abbildung 13: Darstellung des Fehlers eines Netzes bei der Variation von zwei Gewichten inspiriert aus: [Zell, 1994]	20
Abbildung 14: Fehlerfälle, die beim Gradientenabstiegsverfahren auftreten können inspiriert aus: [Zell, 1994].....	25
Abbildung 15: Aufbau eines MLP zur OCR Erkennung inspiriert aus: [Lämmel & Cleve, 2008]	26
Abbildung 16: Darstellung der Arbeitsweise einer Convolution Schicht	27
Abbildung 17: Charakteristik der Rectifier Linear Unit (ReLu) Funktion inspiriert aus: [Wartala, 2018].....	28
Abbildung 18: Darstellung der Arbeitsweise des Max Pooling.....	29
Abbildung 19: Aufbau des LeNet5 zur Klassifikation von Schriftzeichen [LeCun, et al., 1998].....	30
Abbildung 20: Aufbau der Konfusions-Matrix inspiriert aus: [Japkowicz & Shah, 2012]	31
Abbildung 21: Skizze der Aufteilung der Bestandteile des Jäteroboters	34
Abbildung 22: BoniRob zwischen vier Karottenreihen	36
Abbildung 23: Sensoraufhängung [Knoll, et al., 2016 b]	37
Abbildung 24: Links RGB Bild und Rechts IR-Bild der JAI Kamera.....	38
Abbildung 25: Aufnahme der Nikon 5500 D.....	39
Abbildung 26: TOF 3D Aufnahme der Kinect II.....	40
Abbildung 27: Abgetastetes Bild des Laserscanners	41
Abbildung 28: RGB Aufnahme unter unterschiedlichen Wetterbedingungen und Wachstumsstadien (links junges Wachstumsstadium mit Regen, rechts spätes Wachstumsstadium mit Trockenheit)	42
Abbildung 29: Vergrößerte Darstellung des Randbereichs zwischen Pflanze und Erde	44
Abbildung 30: Genauigkeit der einzelnen Entfernungen der manuellen Segmentierungsmasken	45
Abbildung 31: Genauigkeitsverlauf der einfachen Schwellwerte, ohne Betrachtung der einzelnen Bildelemente und rechts die ROC-Kurve.....	49
Abbildung 32: Genauigkeitsverlauf des einfachen Schwellwertes, Analyse über einzelne Bildelemente, sowie rechts die ROC-Kurve	50
Abbildung 33: NDVI Verfahren	52

Abbildung 34: Maskiertes RGB Bild einer JAI-Kamera	52
Abbildung 35: Genauigkeit über alle Bilder von dem NDVI-Verfahren sowie rechts die ROC-Kurve	53
Abbildung 36: Genauigkeitsverlauf des NDVI-Verfahrens, Analyse über einzelne Bildelemente sowie rechts die ROC-Kurve	54
Abbildung 37: Precisionswerte ohne Skalierung für das NDVI-Verfahren. Links Analyse über alle Bilder; rechts Analyse über Einzelbildmessung	55
Abbildung 38: Flussdiagramm des Segmentierungsalgorithmus für den RGB Farbraum.....	56
Abbildung 39: Das originale RGB Bild (oben) und unten von links, der rote Kanal, der grüne Kanal und der blaue Kanal als Grauwertbild	57
Abbildung 40: Histogramm der drei Farbkanäle	57
Abbildung 41: Darstellung des Grauwertbildes mit dem Namen PlantImg und sein Histogramm	58
Abbildung 42: Maske des RGB Algorithmus und das maskierte Eingangsbild als Ergebnis.....	59
Abbildung 43: Genauigkeitsverlauf des RGB-Segmentierungsverfahrens; Links: Analyse ohne Betrachtung der einzelnen Bilder; Rechts: Analyse über einzelne Bildelemente.	60
Abbildung 44: Genauigkeitsverlauf des RGB-Segmentierungsverfahrens; Links: Analyse ohne Betrachtung der einzelnen Bilder; Rechts: Analyse über einzelne Bildelemente	61
Abbildung 45: Flussdiagramm des Segmentierungsalgorithmus im HSV-Raum.....	62
Abbildung 46: Aufteilung und Darstellung eines RGB Bildes in die drei Kanäle des HSV-Raumes .	64
Abbildung 47: Darstellung des Histogramms des Farbwert-Kanals.....	65
Abbildung 48: Das binärisierte H-Bin-Bild mit dem Bandpassfilter	65
Abbildung 49: Histogramm von dem S-Bild	66
Abbildung 50: S-BIN Bild nach der Verwendung des Schwellwertes	67
Abbildung 51: Vom Hintergrund segmentierte Pflanzen aus dem RGB Bild	67
Abbildung 52: Genauigkeitsverlauf des HSV-Segmentierungsverfahrens; links: Analyse ohne Betrachtung der einzelnen Bilder; rechts: Analyse über einzelne Bildelemente.....	68
Abbildung 53: Verlauf des Precision-Wertes des HSV-Segmentierungsverfahrens; links: Analyse ohne Betrachtung der einzelnen Bilder; rechts: Analyse über einzelne Bildelemente.	70
Abbildung 54: Histogramm der Schwellwertverteilung für den optimalen Schwellwert für jedes Bild im HSV-Segmentierungsverfahren	71
Abbildung 55: Darstellung eines RGB Bildes (links) in einer 3 Dimensionale Punktwolke (rechts) .	72
Abbildung 56: Projektion der Punktwolke auf die rot-grüne Ebene.....	72
Abbildung 57: Projektion der Punktwolke auf die blau-grüne Ebene	73
Abbildung 58: neuronales Netz zum Erstellen der Transformation in den u-v-Raum.....	74
Abbildung 59: Darstellung der Projektion der kompletten Punktwolke auf den neuen zweidimensionalen Raum der u-v-Ebene	75
Abbildung 60: Darstellung der Projektion der reduzierten Punktwolke auf den neuen zweidimensionalen Raum der u-v-Ebene	75
Abbildung 61: Künstliches neuronales Netz zur pixelweisen Segmentierung.....	77
Abbildung 62: Darstellung der verschiedenen Pixel nach der Koordinatentransformation. Die Geraden zeigen die Trenngeraden. Links: vereinigte Darstellung, grün sind Pflanzenpixel, rot sind Hintergrundpixel; Mitte nur Pflanzenpixel; Rechts nur Hintergrundpixel.....	78
Abbildung 63: MLP Netzaufbau zur Segmentierung mit RGB Eingangsbild, zwei verdeckten Schichten und zwei Neuronen als Ausgang	80
Abbildung 64: Darstellung der Genauigkeit der untersuchten Netze auf den Datensatz, der alle Bilder beinhaltet und keine Analyse der Einzelbilder vornimmt.	81
Abbildung 65: Darstellung der Genauigkeit der untersuchten Netze auf den Datensatz, der alle Bilder beinhaltet und die Analyse über die Einzelbilder vornimmt.	82

Abbildung 66: Darstellung der Precision aller untersuchten Netze; Links: Analyse ohne Einzelbildbetrachtung; Rechts: Analyse mit Einzelbildbetrachtung.	83
Abbildung 67: Darstellung der False-Positiv Werte aller untersuchten Netze anhand der Größe der Pixelfläche und Anzahl der Strukturen.....	84
Abbildung 68: Darstellung der Strukturflächenanalyse der neu untersuchten Netze	85
Abbildung 69: Beispielbilder (links: Datensatz trocken, rechts: Datensatz nass) mit weißen Flecken in der Hintergrundstruktur.....	87
Abbildung 70: Darstellung der Verteilung des Pixelabstandes vom Pflanzenpixel zum Hintergrund	87
Abbildung 71: Signalpfad der möglichen hintereinander geschalteten Netze.	88
Abbildung 72: Darstellung des Grauwertproblems der Jai-Kamera; Links: RGB-Bild; Rechts: resultierendes Grauwertbild	89
Abbildung 73: Zusammenfassende Darstellung des Struktur-Flächen-Diagramms aller Segmentierungsverfahren.....	90
Abbildung 74: Darstellung der verschiedenen simulierten Translationsunschärfen am Beispiel eines Bildes; von links: Original, Integration über 10 Pixel, 20 Pixel, 30 Pixel	91
Abbildung 75: Darstellung des Zusammenhangs zwischen Genauigkeit und Translationsunschärfe; Links: Verlauf über Schwellwerte; Rechts: Veränderung der Genauigkeit über die maximalen Werte.....	91
Abbildung 76: Darstellung der verschiedenen simulierten allgemeinen Unschärfen am Beispiel eines Bildes; Von links: Original, Kernel 9, 17 und 25.....	92
Abbildung 77: <i>Darstellung des Zusammenhangs zwischen Genauigkeit und Unschärfe; Links: Verlauf über Schwellwerte; Rechts: beste Genauigkeit der einzelnen Kernelgrößen.....</i>	92
Abbildung 78: Darstellung der verschiedenen simulierten Belichtungsfälle; Von links: 30%, Original, 200% und 300% Belichtung.....	93
Abbildung 79: Darstellung des Zusammenhangs zwischen Genauigkeit und Belichtungsstärke; Links: Verlauf über Schwellwerte; Rechts: beste Genauigkeit über einzelne Belichtungsstufen	94
Abbildung 80: Darstellung der verschiedenen simulierten Farbintensitäten; Von links: 0%, Original, 200% und 300% Farbintensität	94
Abbildung 81: Darstellung des Zusammenhangs zwischen Genauigkeit und Farbintensität; Links: Verlauf über Schwellwerte; Rechts: beste Genauigkeit über einzelne Farbintensitätsstufen.....	95
Abbildung 82: Signalflussgraph der pixelweisen Klassifikation	101
Abbildung 83: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des ersten Klassifikators	105
Abbildung 84: Darstellung der Genauigkeitshistogramme des ersten Klassifikators, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“	105
Abbildung 85: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des ersten Klassifikators mit segmentierten Bildern.	107
Abbildung 86: Darstellung der Genauigkeitshistogramme des ersten Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“	108
Abbildung 87: Struktur des Convolutional-Neural-Networks für die Pflanzenklassifikation [Knoll, et al., 2018].....	109
Abbildung 88: Kamerabild und Klassifikationsergebnis des Pixelweise-Klassifikators	110
Abbildung 89: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des ersten CNN-Klassifikators	111
Abbildung 90: Darstellung der Genauigkeitshistogramme des CNN-Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“	112
Abbildung 91: Ergebnis eines CNN-Klassifikationsergebnisses; Grün: überwiegend Karottenpixel; Rot: überwiegend Beikrautpixel; Blauer Kreis: stellt den Fall der extremen Klassifikationsergebnisse dar.....	113

Abbildung 92: Struktur des optimierten Convolutional-Neural-Networks für die Pflanzenklassifikation	113
Abbildung 93: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des verkleinerten CNN-Klassifikators.....	115
Abbildung 94: Darstellung der Genauigkeitshistogramme des verkleinerten CNN-Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“	115
Abbildung 95: Inceptionmodul des GoogLeNet Version 1 [Szegedy, et al., 2015].....	116
Abbildung 96: Weiterentwicklung des Inceptionmoduls mit Dimensionsreduktion [Szegedy, et al., 2015].....	117
Abbildung 97: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des GoogLeNet-Klassifikators	118
Abbildung 98: Darstellung der Genauigkeitshistogramme des GoogLeNet-Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“	119
Abbildung 99: Aufbau der AlexNet Struktur [Krizhevsky, et al., 2012]	120
Abbildung 100: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des AlexNet-Klassifikators.....	122
Abbildung 101: Darstellung der Genauigkeitshistogramme des AlexNet-Klassifikators mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“	122
Abbildung 102: Ablaufplan für den Area-Klassifikator	123
Abbildung 103: Aufteilung des Kamerabildes in die 18 Bildsegmente für den Area-Klassifikator..	124
Abbildung 104: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des ersten Area-Klassifikators mit 4 Ausgängen.....	126
Abbildung 105: Darstellung der Genauigkeitshistogramme des ersten Area-Klassifikators mit 4 Ausgängen mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“	127
Abbildung 106: Darstellung eines Klassifikationsergebnisses des ersten Area-Klassifikators mit 4 Ausgängen mittels Überlagerungsdarstellung; grüne Flächen sind Karotten, rote Flächen beinhalten Beikraut	127
Abbildung 107: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des modifizierten Area-Klassifikators mit 4 Ausgängen.....	129
Abbildung 108: Darstellung der Genauigkeitshistogramme des modifizierten Area-Klassifikators mit 4 Ausgängen mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“	129
Abbildung 109: Darstellung der vorkommenden verschiedenen Pflanzengenauigkeiten in einem Histogramm des Area-Klassifikators mit 2 Ausgängen.	131
Abbildung 110: Darstellung der Genauigkeitshistogramme des Area-Klassifikators mit 2 Ausgängen mit segmentierten Bildern, aufgeteilt in die Klassen; Links: Histogramm der Klasse „Karotte“; Rechts: Histogramm der Klasse „Beikraut“	132
Abbildung 111: Beispielkonstellation von Pflanzen, die zum Problem der Falschklassifikation führen könnten	135
Abbildung 112: Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen; Links: Klassifikation ohne Ausblendung von Nachbarpflanzen; Rechts: Klassifikation mit Ausblenden von Nachbarpflanzen.....	135
Abbildung 113: Fallunterscheidung bei der Anwendung eines Features.....	140
Abbildung 114: Visualisierung der Ausschnittsbetrachtung zweier benachbarter Berechnungen	141
Abbildung 115: Visualisierung einer identischen Featureberechnung zweier benachbarter Berechnungen.....	141

Abbildung 116: Darstellung des Pooling Problems	142
Abbildung 117: Aufbau der entwickelten CPU für die schnelle Berechnung der CNN Strukturen ..	144
Abbildung 118: Detaillierter Aufbau der Control Unit.....	145
Abbildung 119: Detaillierte Darstellung der Datenpfadeinheit	146
Abbildung 120: Darstellung der Adressaufteilung des Speichers.....	147
Abbildung 121: Aufbau des LeNet 1 zur Klassifikation von Schriftzeichen [LeCun, et al., 1995]...	147
Abbildung 122: Ergebnisse der ursprünglichen Berechnung (links) und der optimierten Berechnung (rechts) desselben Bildes	150
Abbildung 123: Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen; Links: ursprüngliche Methode; Rechts: optimierte Methode	151
Abbildung 124: Ergebnisse der optimierten Berechnung mit den herkömmlichen Gewichten (links) und der optimierten Berechnung mit angepassten Gewichten (rechts)	152
Abbildung 125: Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen; Links: optimierte Methode mit ursprünglichen Gewichten; Rechts: optimierte Methode mit angepassten Gewichten.....	153
Abbildung 126: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) im Zeit-Klassifikations-Diagramm	154
Abbildung 127: Darstellung des Zeitverhaltens der einzelnen Schichten im Netzwerk (erste Convolution-Schicht blau, zweite Convolution-Schicht grün, erste Dense-Schicht rot, zweite und dritte Dense-Schicht violett)	155
Abbildung 128: Klassifikationsergebnis mit der optimierten Netzwerkstruktur und der herkömmlichen (links) und der optimierten (rechts) Methode mit denselben Gewichten.	157
Abbildung 129: Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen mit dem optimierten Netzwerk; Links: ursprüngliche Methode; Rechts: optimierte Methode mit gleichen Gewichten.....	158
Abbildung 130: Klassifikationsergebnis mit der herkömmlichen Netzwerkstruktur und der herkömmlichen (links) und der optimierten Methode mit angepassten Gewichten (rechts).....	158
Abbildung 131: Histogrammdarstellung der Genauigkeitsverteilung über alle Pflanzen; Links: ursprüngliche Methode; Rechts: optimierte Methode mit angepasstem Training.....	159
Abbildung 132: Zeit-Klassifikations-Diagramm der optimierten Netzwerkstruktur mit der herkömmlichen Methode (rot) und der optimierten Methode (blau) über die 1075 Validierungsbilder	160
Abbildung 133: Darstellung der verschiedenen simulierten Translationsunschärfen am Beispiel eines Bildes; Von Links: Original, Integration über 10 Pixel, 20 Pixel und 30 Pixel	162
Abbildung 134: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) mit der Auswertung verschiedener Translationsgeschwindigkeiten (linke Skala); Schwarz ist die Anzahl der Klassifikationen (rechte Skala).....	162
Abbildung 135: Darstellung der Precision (links) und des Recalls (rechts) für die Translationsunschärfe.....	163
Abbildung 136: Darstellung der verschiedenen simulierten allgemeinen Unschärfen am Beispiel eines Bildes; Von links: Original, Integration über 9, 17 und 25 Pixel für die Kernelgröße	165
Abbildung 137: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) mit der Auswertung verschiedener allgemeiner Unschärfen (linke Skala); Schwarz ist die Anzahl der Klassifikationen (rechte Skala).....	165
Abbildung 138: Darstellung der Precision (links) und des Recalls (rechts) für die allgemeine Unschärfe	166
Abbildung 139: Darstellung der verschiedenen simulierten Helligkeitsstufen am Beispiel eines Bildes; Von links: 30%, Original (100%), 200% und 300% Belichtung	168

Abbildung 140: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) mit Auswertung der verschiedenen Helligkeitsstufen (linke Skala); Schwarz ist die Anzahl der Klassifikationen (rechte Skala).....	168
Abbildung 141: Darstellung der Precision (links) und des Recalls (rechts) für die verschiedenen Helligkeitsstufen.....	169
Abbildung 142: Darstellung der verschiedenen simulierten Farbsättigungen am Beispiel eines Bildes; Von links: 0%, Original (100%), 200% und 300% Farbsättigung	171
Abbildung 143: Gegenüberstellung der ursprünglichen Methode (rot) und der optimierten Methode (blau) mit der Auswertung verschiedener Farbsättigungen (linke Skala); Schwarz ist die Anzahl der Klassifikationen (rechte Skala).....	171
Abbildung 144: Darstellung der Precision (links) und des Recalls (rechts) für die verschiedenen Farbsättigungen	172

IV. Tabellenverzeichnis

Tabelle 1: Statistische Auswertung des einfachen Schwellwertes an der Position des besten Genauigkeitswertes der Analyse ohne einzelne Bildelemente	49
Tabelle 2: Statistische Auswertung des einfachen Schwellwertes an der Position des besten Genauigkeitswertes der Analyse über einzelne Bildelemente	51
Tabelle 3: Statistische Auswertung des einfachen Schwellwertes unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild	51
Tabelle 4: Statistische Auswertung des NDVI-Verfahrens an der Position des besten Genauigkeitswertes der Analyse ohne einzelne Bildelemente	53
Tabelle 5: Statistische Auswertung des NDVI-Verfahrens an der Position des besten Genauigkeitswertes der Analyse über einzelne Bildelemente	54
Tabelle 6: Statistische Auswertung des einfachen Schwellwertes unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild	56
Tabelle 7: Statistische Auswertung des RGB-Segmentierungsverfahrens an der Position des besten Genauigkeitswertes der Analyse ohne einzelne Bildelemente	60
Tabelle 8: Statistische Auswertung des RGB-Segmentierungsverfahrens an der Position des besten Genauigkeitswertes der Analyse über einzelne Bildelemente	61
Tabelle 9: Statistische Auswertung des RGB-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild	62
Tabelle 10: Konfusions-Matrix des HSV-Segmentierungsverfahrens über den Datensatz, der alle Bilder beinhaltet und ohne Einzelbildbetrachtung (ohne Skalierung)	69
Tabelle 11: Statistische Auswertung des HSV-Segmentierungsverfahrens an der Position des besten Genauigkeitswertes der Analyse ohne einzelne Bildelemente	69
Tabelle 12: Statistische Auswertung des HSV-Segmentierungsverfahrens an der Position des besten Genauigkeitswertes der Analyse über einzelne Bildelemente	69
Tabelle 13: Statistische Auswertung des HSV-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild	71
Tabelle 14: Statistische Auswertung des KNN-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild mit Skalierung	78
Tabelle 15: Statistische Auswertung des KNN-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild und der Auswertung über alle Bilder mit Skalierung	79
Tabelle 16: Statistische Auswertung des KNN-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild ohne Skalierung	79
Tabelle 17: Statistische Auswertung des KNN-Segmentierungsverfahrens unter Berücksichtigung der optimalen Parameter für jedes einzelne Bild und der Auswertung über alle Bilder ohne Skalierung..	79
Tabelle 18: Statistische Auswertung des MLP zur Segmentierung mit der Analyse ohne einzelne Bildelemente	85
Tabelle 19: Statistische Auswertung des MLP zur Segmentierung mit der Analyse über einzelne Bildelemente	85
Tabelle 20: Statistische Auswertung des MLP zur Segmentierung mit der Analyse über einzelne Bildelemente ohne Skalierung	86
Tabelle 21: Auflistung aller Segmentierungsverfahren mit den jeweils besten Parametern für jedes Bild mit Skalierung	89
Tabelle 22: Auflistung der zusammengefassten Pflanzengruppen für die Auswertungen	98
Tabelle 23: Auflistung der zusammengefassten Pflanzengruppen für die Auswertungen	100

Tabelle 24: Darstellung der Auswertungsergebnisse über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung	104
Tabelle 25: Darstellung der Auswertungsergebnisse über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung und schwarzem Hintergrund	106
Tabelle 26: Darstellung der Auswertungsergebnisse des CNNs über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung	110
Tabelle 27: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des CNN-Klassifikators	111
Tabelle 28: Darstellung der Auswertungsergebnisse des optimierten CNNs über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung	114
Tabelle 29: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des CNN-Klassifikators	114
Tabelle 30: Darstellung der Auswertungsergebnisse des GoogLeNet über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung	117
Tabelle 31: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien der GoogLeNet Klassifikators	118
Tabelle 32: Darstellung der Auswertungsergebnisse des AlexNet über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung	121
Tabelle 33: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des AlexNet Klassifikators	121
Tabelle 34: Darstellung der Elementgrößen des Area Klassifikators	124
Tabelle 35: Darstellung der Auswertungsergebnisse des ersten Area-Klassifikators mit 4 Ausgängen über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung	125
Tabelle 36: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des ersten Area-Klassifikators mit 4 Ausgängen	126
Tabelle 37: Darstellung der Auswertungsergebnisse des modifizierten Area-Klassifikators mit 4 Ausgängen über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung	128
Tabelle 38: Ausgangsinterpretation des Area-Klassifikators mit zwei Ausgangsneuronen	130
Tabelle 39: Darstellung der Auswertungsergebnisse des Area-Klassifikators mit 2 Ausgängen über alle Pixel der 1075 Bilder mit Verwendung einer Skalierung	130
Tabelle 40: Darstellung der Genauigkeit über die einzelnen Pflanzenkategorien des Area-Klassifikators mit 2 Ausgängen	131
Tabelle 41: Gegenüberstellung der Klassifikationsergebnisse aus den globalen Genauigkeitsuntersuchungen	133
Tabelle 42: Benötigte Zeit für die Klassifikation der 1075 Validierungsbilder	134
Tabelle 43: Darstellung der Auswertungsergebnisse der globalen Genauigkeit für die ursprüngliche und optimierte Methode	150
Tabelle 44: Darstellung der Auswertungsergebnisse der globalen Genauigkeit für die ursprüngliche und optimierte Methode und die optimierte Methode mit angepassten Gewichten	152
Tabelle 45: Darstellung der Auswertungsergebnisse der globalen Genauigkeit für die ursprüngliche und optimierte Methode mit denselben Gewichten	157
Tabelle 46: Darstellung der Auswertungsergebnisse der globalen Genauigkeit für die ursprüngliche und optimierte Methode mit und ohne angepasste Gewichte	159
Tabelle 47: Darstellung des Verhaltens der Genauigkeitsverteilung über die Translationsunschärfe	164
Tabelle 48: Darstellung des Verhaltens der Genauigkeitsverteilung über die Zunahme der allgemeinen Unschärfe	167
Tabelle 49: Darstellung des Verhaltens der Genauigkeitsverteilung über die Zunahme der Belichtungsstufen	170

<i>Tabelle 50:</i> <i>Darstellung des Verhaltens der Genauigkeitsverteilung über die Zunahme der Farbsättigung</i>	173
---	-----

V. Literaturverzeichnis

- Albert, A., 2015. *Towards sustainable agriculture by robotics*, s.l.: Bosch.
- Altera, 2015. *DE1-SoC_User_manual_v.1.2.2*. s.l.:Altera.
- Baldominos, A., Saez, Y. & Isasi, P., 2019. A Survey of Handwritten Character Recognition with MNIST and EMNIST. *applied sciences MDPI*.
- Bartsch, H. J., 2004. *Taschenbuch Mathematischer Formeln*. 20. Auflage Hrsg. s.l.:Fachbuchverlag Leipzig.
- Basler, F., 2017. *Datenblatt der daA1920-30uc*. [Online]
Available at: <https://docs.baslerweb.com/#t=en%2Fdaa1920-30uc.htm>
- Bluetechnix, 2015. *Datenblatt der Argos3D_P100_OV.pdf*. s.l.:Bluetechnix.
- Cicco, M. D., Potena, C. & Giorgio, G., 2017. *Automatic Model Based Dataset Generation for Fast and Accurate Crop and Weeds Detection*. s.l., Cornell University.
- Ciresan, D. C., Meier, U., Gambardella, L. M. & Schmidhuber, J., 2010. Deep, big, simple neural nets for handwritten digit recognition.. *Neural Comput*, pp. 3207-3220.
- Czymmek, V., 2017. *Development of a software for the calculation of a Convolutional Neural Network using a CUDA-capable GPU for the classification of plants in organic farming*. s.l.:FWW.
- Czymmek, V., Harders, L. O., Knoll, F. J. & Hussmann, S., 2019. *Vision-Based Deep Learning Approach for Real-Time Detection of Weeds in Organic Farming*. s.l., IEEE Instrumentation and Measurement Society.
- Defant, P. D. M., 2016. Aquisitions- und Risikoanalyse. In: s.l.:s.n.
- Demant, C., Streicher-Abel, B. & Springhoff, A., 2011. *Industrielle Bildverarbeitung*. 3. Hrsg. s.l.:Springer.
- Ganie, M. & Nusrath, A., 2016. *Determining the vegetation indices (NDVI) from Landsat 8 satellite data*. s.l., International Journal of Advanced Research.
- github, 2019. *Stanford-Kursen convolution networks*. [Online]
Available at: <http://cs231n.github.io/convolutional-networks/>
[Zugriff am Juni 2019].
- Grelcke, M., 2017. *Implementation of a real-time Convolutional Neural Network CPU design on a FPGA*. Heide, FWW.
- Haug, S., Michaels, A., Biber, P. & Ostermann, J., 2014. Plant classification system for crop /weed discrimination without segmentation. *IEEE Winter conference on Applications of Computer Vision (WACV)*, p. pp. 1142 – 1149.
- Holtorf, T., Knoll, F. J. & Hussmann, S., 2016. *Multimodal Image Stitching Algorithm for Weed Control Applications in Organic Farming*. London, IEEE Technically Sponsored SAI Computing Conference.
- Hussmann, S. & Hermanski, A., 2013. *One-Phase algorithm for continuous wave TOF machine vision applications*. s.l., IEEE Transactions on Instrumentation and Measurement.

-
- Hussmann, S., Hermanski, A. & Edeler, T., 2011. *Real-Time Motion Artifact Suppression in TOF Camera Systems*. s.l., IEEE Trans. on Instrumentation and Measurement.
- Hussmann, S. & Knoll, F. J., 2013. *Modulation Method for minimizing the Depth Distortion Offset of Lock-in ToF Cameras*. USA, I2MTC 2013, Proc. of the 30th Int. Conf. on Inst. and Meas. Tec.
- Hussmann, S., Knoll, F. J., Czymmek, V., Meissner, A., Holtorf, T., 2019. *Development and evaluation of a low-cost delta robot system for weed control applications in organic farming*. I2MTC 2019, Proc. of the 36th Int. Conf. on Inst. and Meas. Tec, IEEE Instrumentation and Measurement Society.
- Hussmann, S., Knoll, F. J. & Edeler, T., 2014. *Modulation Method Including Noise Model for Minimizing the Wiggling Error of TOF Cameras*. s.l., IEEE Trans. on Instrumentation and Measurement.
- JAI, F., 2015. *Datenblatt von AD-130 GE*. s.l.:JAI.
- Japkowicz, N. & Shah, M., 2012. *Evaluating Learning Algorithms*. 2 Hrsg. s.l.:Cambridge.
- Kaierle, S., Marx, C., Rath, T. & Hustedt, M., 2013. *Find and Irradiate — Lasers Used for Weed Control*. s.l., LaserTechnikJournal.
- Kargar B., A. H. & Shirzadifar, A. M., 2013. *Automatic weed detection system and smart herbicide sprayer robot for corn fields*. IEEE (ICRoM), 2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM).
- Klamt, S., 1998. *Mustererkennung unter Echtzeit-Bedingungen mit deren Anwendung bei der Unterscheidung von Mais und Beikräutern*. Universität Osnabrück: Diplomarbeiten Agentur.
- Knoll, F. J., 2014. *Bestimmung des Wurzelansatzes und des Austrittsvektors von Beikräutern mittels verschiedener 3D-Messverfahren*. s.l.:Masterthesis.
- Knoll, F. J. & Czymmek, V., 2018. *Automation in Agriculture - Securing Food Supplies for Future Generations: The German Vision of Industry 4.0 Applied in Organic Farming*. s.l.:InTech.
- Knoll, F. J., Czymmek, V., Harders, L. O. & Hussmann, S., 2019. *Real-time classification of weeds in organic carrot production using deep learning algorithms*. s.l., Journal Computers and Electronics in Agriculture (Elsevier), Volume 167, 2019..
- Knoll, F. J., Czymmek, V., Poczihoski, S., Holtorf, T., Hussmann, S., 2018. *Improving efficiency of organic farming by using a deep learning classification approach*. s.l., Journal Computers and Electronics in Agriculture (Elsevier), Volume 153, 2018, pp. 347-356..
- Knoll, F. J., Grelcke, M., Czymmek, V., Holtorf, T., Hussmann, S., 2017. *CPU architecture for a fast and energy-saving calculation of convolution neural networks*. s.l., SPIE 10334, Digital Optical Technologies 2017/Automated Visual Inspection and Machine Vision II.
- Knoll, F. J., Holtorf, T. & Hußmann, S., 2015 a. Fast Correspondence Search Algorithm for Stereo Vision Range Sensors,. *I2MTC IEEE Instrumentation and Measurement Society Proc. of the 32th Int. Conf. on Inst. and Meas. Tec*, pp. pp. 23-28..
- Knoll, F. J., Holtorf, T. & Hußmann, S., 2016 a. Investigation of different plant root exit point vector search algorithms in organic farming. *IEEE Trans. on Instrumentation and Measurement*, 65[5].
- Knoll, F. J., Holtorf, T. & Hußmann, S., 2016 b. Investigation of Different Sensor Systems to Classify Plant and Weed in Organic Farming Applications. *IEEE Technically Sponsored SAI Computing Conference*, pp. pp. 343 - 348..

Knoll, F. J., Holtorf, T. & Hussmann, S., 2016 c. *Vegetation index determination method based on color room processing for weed control applications in organic farming*. Proc. of the 33th Int. Conf. on Inst. and Meas. Tec, IEEE Instrumentation and Measurement.

Knoll, F. J., Holtorf, T. & Hußmann, S., 2017. Modified 3D time-of-flight camera for object separation in organic farming. *Proc. SPIE Vol.10335, Digital Optical Technologies*.

Knoll, F. J., Holtorf, T., Schmidt, M. & Hußmann, S., 2015 b. Plant root exit point search algorithm for weed control applications in organic farming. *I2MTC, Proc. of the 32th Int. Conf. on Inst. and Meas. Tec., IEEE Instrumentation and Measurement Society*, pp. pp. 18-22..

Krizhevsky, A., Sutskever, I. & Hinton, G. E., 2012. *ImageNet Classification with Deep Convolutional Neural Networks*. s.l., Neural Information Processing Systems (NIPS).

Kruse, R. et al., 2011. *Computational Intelligence*. s.l.:Vieweg + Teubner.

Lämmel, U. & Cleve, J., 2008. *Künstliche Intelligenz*. s.l.:Hanser.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P., 1998. *Gradient-based learning applied to document recognition*. s.l., IEEE.

LeCun, Y., Cortes, C. & Burges, C. J., 2019. *THE MNIST DATABASE of handwritten digits*. [Online] Available at: <http://yann.lecun.com/exdb/mnist/>

LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., Drucker, H., Guyon, I., Müller, U., Säckinger, E., Simard, P., Vapnik, V., 1995. *Comparison of learning algorithms for handwritten digit recognition*. Paris, EC2 & Cie.

LMI, 2015. *Datenblatt des LMI Gocator 2350*. s.l.:LMI.

Matsushita, B., Yang, W., Chen, J., Onda, Y., Qiu, G., 2007. *Sensitivity of the Enhanced Vegetation Index (EVI) and Normalized Difference Vegetation Index (NDVI) to Topographic Effects: A Case Study in High-Density Cypress Forest*. s.l., Sensors 2007.

Meißner, A., 2018. *Evaluation and integration of a real-time weed destruction unit in an autonomous weeding robot by means of a Delta-robot*. s.l.:FWH.

Michaels, A. Kohlbrecher, M., Trautz, D., Kielhorn, A., Langsenkamp, F., Sellmann, F., 2015. Deutschland, Patentnr. DE 10 2013 222 776 A1 2015.05.13.

Niang, S., 2015. *Development of a real-time weed elimination unit based on current induction by means of a focused Tesla inductor discharge ionization nozzle*. s.l.:FWH.

Nikon, F., 2015. *Produktbeschreibung und Datenblatt der Nikon 5500D*, Dokumentenname: 3258331160.pdf. s.l.:Nikon.

Oebel, H., 2006. *Teilschlagspezifische Unkrautbekämpfung durch raumbezogene Bildverarbeitung im Offline- (und Online-) Verfahren (TURBO)*. Universität Hohenheim: s.n.

Ohser, J., 2018. *Angewandte Bildverarbeitung und Bildanalyse*. s.l.:Hanser.

Patent #1, Hussmann, S., Knoll, F. J. & Czymmek, V., 2020. *Trägersystem mit einem Träger und einer mobilen Vorrichtung zur Bearbeitung des Bodens und/oder zur Manipulation der Flora und Fauna und Verfahren hierzu*. s.l. Patentnr. DE102018120755A1 (27.02.2020).

-
- Patent #2, Hussmann, S., Knoll, F. J. & Czymbek, V., 2020. *Mobile Analyse- und Bearbeitungsvorrichtung, Verfahren und Trägersystem*. s.l. Patentnr. DE102018120756A1 (27.02.2020).
- Patent #3, Hussmann, S., Czymbek, V. & Knoll, F. J., 2020. *Mobile Analyse- und Bearbeitungsvorrichtung*. s.l. Patentnr. DE102018120753A1 (27.02.2020).
- Patent #4, Hussmann, S., Knoll, F. J. & Czymbek, V., 2020. *Carrier system comprising a carrier and a mobile device for tilling the soil and/or for manipulating flora and fauna, and method therefor*. International, Patentnr. WO 2020/039045 A1 (27.02.2020).
- Patent #5, Hussmann, S., Knoll, F. J. & Czymbek, V., 2020. *Mobile analysis and processing device*. International, Patentnr. WO 2020/039046 A1 (27.02.2020).
- Pfau, D., 2016. *Entwurf und Programmierung einer Multirechner-Lernsoftware für Künstliche Neuronale Netze*. FHW: s.n.
- PMD, F., 2015. *Datenblatt der CamCube II*. s.l.:PMD.
- Poczhoski, S., 2016. *Evaluation of Deep-Learning-Networks for pattern recognition for a weed control application*. s.l.:FHW.
- Potena, C., Nardi, D. & Pretto, A., 2017. *Fast and Accurate Crop and Weed Identification with Summarized Train Sets for Precision Agriculture*. s.l., IAS 2016 Intelligent Autonomous Systems 14.
- Raytrix, F., 2016. *Datenblatt der R42*. s.l.:Raytrix.
- Redmon, J., Divvala, S., Girshick, R. & Farhadi, A., 2016. *You Only Look Once: Unified, Real-Time Object Detection*. s.l., The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Ronneberger, O., Fischer, P. & Brox, T., 2015. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Munich: Medical Image Computing and Computer-Assisted Intervention MICCAI 2015.
- Rouse, J. W., Haas, J., Shell, R. H. & Deering, D. W., 1974. *Monitoring Vegetation Systems in the Great Plains with ERTS*. s.l., published by NASA.
- Schüttler, T., Girwidz, R. & Zepp, S., 2017. *Validierung von NDVI-Messungen mit einer modifizierten Digitalkamera–Fernerkundung von Vegetation als Thema des Physikunterrichts*. s.l.:FU-Berlin.
- Shengyuan, P. & Jiaming, L., 2019. *Accuracy Improvement of UNet Based on Dilated Convolution*. s.l.:Journal of Physics.
- statista.com, 2019. *statista.com*. [Online]
[Zugriff am Juni 2019].
- Szegedy, C. Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. *Going deeper with convolutions*. s.l., IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Wartala, R., 2018. *Deep Learning Praxiseinstieg*. s.l.:O'Reilly.
- Wong, W. K., Chekima, A., Mariappan, M., Khoo, B., Nadarajan, M., 2014. *Probabilistic multi svm weed species classification for weed scouting and selective spot weeding*. ROMA 2014, IEEE International Symposium on Robotics and Manufacturing Automation.
- Zell, A., 1994. *Simulation neuronaler Netze*. s.l.:Oldenbourg.

Florian Johannes Knoll
Osterhusumer Str. 31b
25813 Husum

TELEFON 0176 / 219 136 26
E-MAIL florianknoll@t-online.de



CURRICULUM VITAE

PERSÖNLICHE DATEN

Geboren:	19. Oktober 1983 in Husum
Staatsangehörigkeit:	deutsch
Familienstand:	Verheiratet, eine Tochter

Berufliche Erfahrungen

10/2019 – dato	Firma: Naiture Forschungs- und Entwicklungsleiter (unter anderem: Weiterentwicklung des Jäteroboters) Betreuung und Organisation von Abschlussarbeiten
04/2017 – 10/2019	Firma: Dirkshof / EED GmbH & Co. KG Entwicklungsingenieur für ein Passivradarsystem Betreuung und Organisation von Abschlussarbeiten
08/2011 – 03/2020	Dozent an der Fachhochschule Westküste Dozent (unter anderem in den Bereichen: Mikroprozessor- und Digitaltechnik Informatik Digitale Systeme Bildverarbeitung Physik Mathematik Real-time Image Processing) Betreuung und Organisation von Abschlussarbeiten
10/2009 – 10/2019	Fachhochschule Westküste Letztes Projekt: Hochgenaue Unkrautererkennung Forschung an schnellen Klassifikation und Sensorsystemen zur Pflanzenerkennung

2016 – 2017

Dozent an der Zhejiang University of Science and Technology (ZUST) in der Volksrepublik China

Mehrwöchige Blockveranstaltung in dem Fach
Bildverarbeitung

06/2016

Dozent an der Hochschule für Angewandte Wissenschaften Hamburg

Dozent (unter anderem in den Bereich:
Informatik)

09/2009 – 06/2015

VHS-Husum

Dozent (unter anderem in den Bereichen:
Programmierung und Elektrotechnik)

10/2012 – 06/2015

Dozent eines Enrichment Programmes für hochbegabte Kinder der DGhK

In den Bereichen:
Physik und Elektrotechnik.

09/2007 – 04/2008

Fraunhofer Institut für Siliziumtechnologie

Praxissemester und Werkstudent

Programmierung in der Mikrosystemtechnik

Programmierung einer Software zur Anzeige und Auswertung von statistischen Messergebnissen vom Wafer (Siliziumscheibe zur Chipherstellung).

Automatische Kalibrierung eines Wafer anhand von visuellen Echtzeitaufnahmen.

Studienarbeit und Diplomarbeit / Masterthesis

09/2014

Masterthesis

Bestimmung des Wurzelansatzes und des Austrittsvektors von Beikräutern mittels verschiedener 3D-Messverfahren.

08/2009

Diplomarbeit

Inbetriebnahme eines Echtzeitbetriebssystems zur schnelleren Sensordatenübertragung via Ethernet eines bestehenden 3D-TOF-Kamerasystems.

03/2009

Studienarbeit

Aufbau einer Meteosat-Wettersatelliten-Empfangsanlage.

Schulische Ausbildung

08/2002 – 06/2005	Berufliches Gymnasium in Husum (Fachgymnasium) (Abschluss: Allgemeine Hochschulreife) Leistungskurse: Elektrotechnik und Mathematik
08/2000 – 06/2002	Berufsfachschule Technik in Husum (Abschluss: Mittlere Reife) Schwerpunkt: Elektrotechnik
08/1995 – 06/2000	Hauptschule Friedrichstadt
08/1990 – 06/1995	Grundschule Friedrichstadt

Studium

02/2016 – dato	Universität zu Lübeck Promotion zur Klassifikation der Pflanzenerkennung mittels Deep-Learning
03/2013 – 09/2014	Hochschule für Angewandte Wissenschaften Hamburg Studienrichtung Elektrotechnik und Informationstechnik Schwerpunkt: Master Mikroelektronische Systeme Abschluss: Master of Science
10/2009 – 10/2012	Universität Hamburg (UHH) Studienrichtung Physik
09/2005 – 09/2009	Fachhochschule Westküste (FWW) Heide Studienrichtung Elektrotechnik und Informationstechnik Schwerpunkt: Elektrotechnik und Kommunikationstechnik Abschluss: Dipl. Ing.(FH)

Husum, den 20. Juli 2020

Florian J. Knoll